Measurement Science and Technology

PAPER

Hierarchical graph neural network with adaptive cross-graph fusion for remaining useful life prediction

To cite this article: Gang Wang *et al* 2023 *Meas. Sci. Technol.* **34** 055112

View the article online for updates and enhancements.

Meas. Sci. Technol. **34** (2023) 055112 (19pp) https://doi.org/10.1088/1361-6501/acb83e

Hierarchical graph neural network with adaptive cross-graph fusion for remaining useful life prediction

Gang Wang[1,2,3], **Yanan Zhang**[1], **Mingfeng Lu**[1] **and Zhangjun Wu**[1,2,3,*]

[1] School of Management, Hefei University of Technology, Hefei, Anhui, People's Republic of China
[2] Key Laboratory of Process Optimization and Intelligent Decision-Making (Hefei University of Technology), Ministry of Education, Hefei, Anhui, People's Republic of China
[3] Ministry of Education Engineering Research Center for Intelligent Decision-Making and Information System Technologies, Hefei 230009, People's Republic of China

E-mail: wuzhangjun@hfut.edu.cn

Received 10 November 2022, revised 1 February 2023
Accepted for publication 2 February 2023
Published 22 February 2023

Abstract

Multi-sensor monitoring data provide abundant information resources for complex machine systems, which facilitates monitoring the degradation process of machinery and ensuring the reliability of the industrial process. However, previous prognostic methods focus more on the sequential data obtained from multi-sensors, while ignoring the underlying prior structural information of the equipment. To fully leverage the structural information into the modeling process, and thus improve the remaining useful life (RUL) prediction performance, a hierarchical graph neural network with adaptive cross-graph fusion (HGNN-ACGF) method for RUL prediction is proposed in this study. In the HGNN-ACGF method, a hierarchical graph consisting of a sensor graph and a module graph is constructed by introducing the structural information to fully model the degradation trend information of the complex machine system. Besides, the graph neural network (GNN) is adopted to learn the representation at both the module graph and sensor graph, and an adaptive cross-graph fusion (ACGF) block is proposed. Owing to the cross-graph fusion block, the representation from different graphs can be fused adaptively by considering the relative importance between different modules and sensors. To verify the proposed method, the experiments were conducted on a set of degradation data sets of aircraft engines provided by the Commercial Modular Aero-Propulsion System Simulation. The experimental results show that the proposed method has superior performance in RUL prediction over the state-of-the-art methods.

Keywords: hierarchical graph, graph neural network, deep learning, remaining useful life prediction

(Some figures may appear in colour only in the online journal)

1. Introduction

Prognostics and health management (PHM), which can be used to evaluate the health status and thus ensure the reliability of complex machine systems, has been successfully applied in various industrial scenarios [1, 2]. As a core task of PHM, remaining useful life (RUL) prediction can provide residual service lifespan information based on sensor monitoring data and guide predictive maintenance in advance before the equipment completely breaks down [3, 4]. To monitor the health status and provide residual service lifespan information, a

[*] Author to whom any correspondence should be addressed.

large number of sensors can be mounted on the industrial equipment, which will generate massive, multivariate sensor monitoring data, e.g. temperature, pressure, and vibration signal. Such numerous sensor data bring new challenges to RUL prediction. Therefore, it is important to model these data and extract effective degradation trend information to improve the RUL prediction performance for complex machine systems.

In general, the RUL prediction methods can be grouped into model-based methods and data-driven methods [5]. The model-based methods require prior expertise oin mechanical systems for constructing precise physical failure models. Commonly used model-based techniques are the Wiener process model, the Weibull model, etc [6, 7]. However, with the development of complexity and multifunctionality, it is difficult to build a precise and applicable physical failure model to estimate the degradation process of machine systems in real applications [8]. Besides, the generalization ability of model-based methods is weak because of the different degradation mechanisms among machine systems. Owing to the rapid development of sensor techniques, data-driven methods based on massive monitoring data have become a new possibility when facing complex machine systems [9, 10]. Since the data-driven methods are easy to apply without the need for prior expertise, some data-driven methods, such as support vector regression (SVR) and hidden Markov models have been proposed and widely applied in RUL prediction [11, 12]. Although the aboveentioned methods can model the degradation trend information hidden in the sensor data, great efforts are still needed on hand-crafted feature construction, which is time-consuming and laborious [13–15].

Fortunately, deep learning methods can extract high-quality representations automatically from multi-sensor data without introducing domain knowledge [16–18]. Besides, they can synthesize the representation learning and target RUL prediction into one process and have better generalization ability in RUL prediction [19, 20]. Therefore, several deep learning-based RUL prediction methods, such as the convolution neural network (CNN) and the ong short-term memory (LSTM) network have been introduced and successfully applied in the RUL prediction field [21, 22]. For example, Yang *et al* proposed an RUL prediction method based on the CNN with a double-CNN structure, which has powerful feature extraction ability and can also preserve useful degradation information during prognosis [23]. Xiang *et al* utilized a multicellular LSTM method to improve the RUL prediction accuracy, in which multiple cellular units are used to update the input data with different degress of importance [24]. In industrial scenarios, multiple sensors are frequently adopted and mounted on different components, monitoring the health status of machine systems in different aspects consistently [25]. These sensors are interconnected and naturally formatted as a sensor graph network, which can also be used to model the degradation trend of machine systems. However, these deep learning methods take the data collected from multiple sensors as a unit, and it is difficult for them to model such graph data in the non-Euclidean domain [26].

To model such graph data, the graph neural network (GNN), which has been used in various domains such as activity recognition and traffic forecasting [27, 28], becomes a better choice. Therefore, the GNN-based method is employed to model the degradation trend information contained in the relationship between sensors for RUL prediction. For example, Yang *et al* proposed a deep GNN method combined with gated recurrent unit (GRU) and constructed the graph using the regression shapelet, which can better represent the degradation state of a bearing and improve the accuracy of the RUL prediction [29]. Li *et al* proposed a hierarchical attention graph convolutional network (GCN)-based RUL prediction method, which can model multi-sensor data with a self-regularized attention mechanism and enhance RUL prediction performance [30]. In industrial applications, the complex machine system usually consists of several interconnected modules. These modules will influence each other in the degradation process of the machine system, according to the equipment structure, i.e. structural information, predefined in the documents. For example, in the aircraft engine, the abnormal degradation behaviors of the fan module may have a stronger influence on the low-pressure chamber (LPC) which is directly connected to the fan module. Their connections are prior knowledge about the equipment and may contain rich information about the degradation process. Besides, the information contained among these modules can also be formatted as a graph according to the predefined structure. However, previous GNN-based methods focused more on the graph data constructed at sensor level, which failed to leverage the degradation trend information hidden in the underlying structural information at module level. To fully depict the degradation process of a complex machine system, the information contained in the module level also matters and should be considered as well. The sensor graph focuses more on the detailed information contained in the sensors data, while the module graph is concentrated on utilizing the equipment's structural information. The structural information at the module level can guide representation learning in the sensor graph, while the information contained at the sensor level can enrich the representation of each node in the module graph. Combining the monitoring sensor data from sensors and structural information among modules can better reflect the degradation process of machine systems. Therefore, it is necessary to make full use of the sensor-level and module-level information simultaneously, and thus improve RUL prediction performance.

Inspired by the abovementioned facts, a hierarchical GNN with adaptive cross-graph fusion (HGNN-ACGF) method for RUL prediction is proposed in this study. Firstly, a hierarchical graph consisting of a sensor-level graph and a module-level graph is constructed based on the collected sensor data and structural information to fully describe the degradation trend information of the complex machine system. Secondly, the GNN is adopted to learn the representation of an individual graph, and a modified Top-k pooling is utilized to enhance the information flow and find meaningful nodes. Meanwhile, the adaptive cross-graph fusion block (ACG-FB) is proposed to achieve representation fusion across different graphs, and thus make full use of the structural information. Finally, the readout function is used to transform the representation of the nodes into the graph representation and various fully connected layers are adopted to achieve the prediction of RUL. To validate the performance of the proposed method, the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) turbofan engine data sets are adopted [31]. The experimental results verified the effectiveness of the proposed method for RUL prediction, which was superior to the traditional machine learning method, typical deep learning methods, and other state-of-the-art methods. The main contributions of this paper are summarized as follows.

(a) A new end-to-end deep learning framework based on the GNN, namely HGNN-ACGF, is proposed for RUL prediction of complex machine systems. To the best of our knowledge, this is the first study to utilize the sensor graph and module graph simultaneously for RUL prediction of complex machine systems.

(b) A GCN-based RUL prediction method is built to model the sensor-level graph and module-level graph simultaneously, in which the information from different graphs can be fully explored. In addition, the ACG-FB is proposed to fuse the information across the sensor graph and module graph adaptively by considering the relationship between different sensors and modules. This fusion block is helpful to make full use of the structural information and learn the degradation trend information contained in the data.

(c) To validate the proposed HGNN-ACGF, the experiments were conducted on the turbofan engine data sets provided by the C-MAPSS. The experimental results demonstrated the effectiveness and superiority of HGNN-ACGF compared with the traditional machine learning method, typical deep learning methods, and other state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents the proposed HGNN-ACGF method in detail. Section 3 describes the experimental design including the data set, evaluation metrics, compared methods, and experimental procedures. Section 4 reports and discusses the experimental results, in which the superiority of the proposed HGNN-ACGF is demonstrated through comparison with other methods. Finally, section 5 closes the paper with conclusions and future work.

2. Proposed method

2.1. Problem definition for GNN in RUL prediction

In industrial scenarios, the sensors mounted on the corresponding modules of a complex machine system can measure the health status of the machine system continuously. To represent the relationship between different sensors or modules, a graph G = (V, E) is adopted, where V means the sensor or module set, and E is the edge set with the weight between the connected sensors or modules. In the degradation process of the machine system, the features matrix $X_n \in R^{|V| \times p}$ is used to represent the signal data from multi-sensors and will keep changing over time, where $|V|$ is the node number and p is the length of the slide time window. The adjacency matrix A is constructed by measuring the relationship between different nodes, in which a larger value means the connected nodes have stronger connections. With the nodes feature matrix X_n and adjacency matrix A, the sensor graph and module graph can be constructed.

Thus, the RUL prediction problem can be formatted as follows. For the multi-sensor monitoring data $[x_1, x_2, \ldots, x_t], x_t \in R^{|V|}$, the slide window is adopted first to generate samples X, where t is the time point in the sensor monitoring data and $X = \{X_1, X_2, \ldots, X_n\}$, $X_n \in R^{|V| \times p}$. Then, the sensor graphs $[g_1^s, g_2^s, \ldots, g_n^s] \in G^s$ and module graphs $[g_1^m, g_2^m, \ldots, g_n^m] \in G^m$ can be constructed, and the corresponding RUL label y can be assigned for each graph. Finally, with the constructed graph data set D = $\{(g_1^s, g_1^m, y_1),\ (g_2^s, g_2^m, y_2),\ \ldots,\ (g_n^s, g_n^m, y_n)\}$, the task of the RUL prediction with the GNN is to model the relationship between the graphs g_n^s, g_n^m, and y_n.

2.2. Framework of the proposed method

RUL prediction can effectively improve the reliability and security of the machine system by predicting the residual useful life of machine components in advance. Previous deep learning-based prognostic methods fail to leverage the underlying structural information and capture the interdependencies of different modules. To fully leverage the structural information into the modeling process and improve the RUL prediction performance, an HGNN-ACGF method for the RUL prediction is proposed, in which a hierarchical graph, consisting of a module graph and a sensor graph, is constructed and different graphs are fused adaptively. The proposed method mainly consists of two key components, which are multi-sensor data collection and model construction. In the multi-sensor data collection phase, the raw monitoring data of each sensor is collected first, and the samples are generated using the slide time window. In the model construction phase, to fully represent the degradation process, the sensor graph and module graph are constructed first, in which the structural information is taken into consideration. In detail, the sensor is considered as a node of the sensor graph, the corresponding node feature is denoted as the sensor monitoring data, and the nodes are connected with each other. As for the module graph, the module is denoted as the node of the module graph. The node feature is represented by the average value of the sensors installed on the module. And the relationship between different modules is determined by the structural information predefined in the documents. Then, in the proposed HGNN-ACGF method, GRU is adopted to learn the temporal embedding of sensor data and the GCN is adopted to update the representation of each node. Meanwhile, the representations of different level graphs are adaptively fused using the proposed ACG-FB by considering the relationship between different sensors and modules. Finally, the representation of different graphs is transformed into the representation vector by the readout function and the RUL is predicted by adopting several fully connected layers. The diagram of the proposed HGNN-ACGF is shown in figure 1.

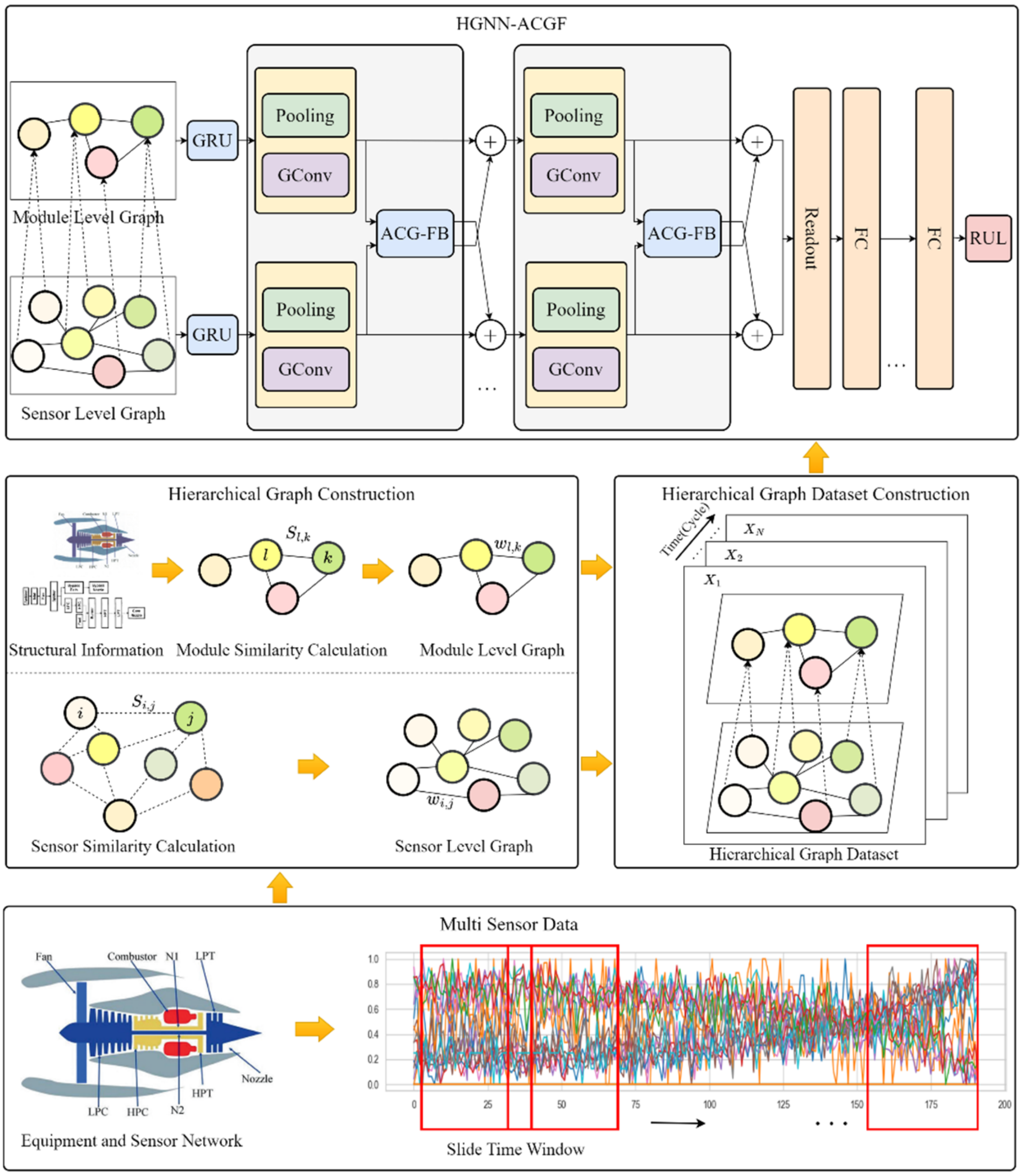


Figure 1. The framework of the proposed method.

2.3. Hierarchical graph construction

Multiple sensors are usually installed on a complex machine system for health status monitoring. These sensors can be naturally formatted as a sensor network according to their spatial relationship. Moreover, a complex machine system usually consists of various modules or components that are interconnected with a well-defined structure. These connection relationships can be a rich source of prior knowledge about the machine degradation process. To generate the hierarchical graph data set, the hierarchical graph, including the sensor graph and module graph, is constructed first. Thus, in this phase, different level graphs are constructed first to better describe the degradation trend information of machines.

To construct the sensor graph, a correlation coefficient-based graph construction approach is adopted [26]. Each sensor of the machine is considered as a node of the graph, and different nodes are connected to each other. As for the weight calculation of each edge of the graph, the Euclidean distance is a commonly used approach in traffic flow forecasting or the action recognition field [32, 33]. However, the relationship between different sensors is implicit: the closest distance between two sensors in the Euclidean domain may not mean they are closely related. That is to say, the Euclidean distance may not be the better choice to measure the relationship between sensors. Thus, the Pearson correlation coefficient (PCC), a frequently used approach to calculate the degree of relevance of two variables, is adopted to measure the relationship between different sensors in this paper, and given as:

$$p^s_{x^s_i, x^s_j} = \frac{E\left(\left(x^s_i - \mu x^s_i\right)\left(x^s_j - \mu x^s_j\right)\right)}{\sigma_{x^s_i}\sigma_{x^s_j}} \quad (1)$$

where $p^s_{x^s_i,x^s_j}$ is the correlation coefficient between nodes x^s_i and x^s_j, which are also denoted as the *i*th and *j*th sensor data, μx^s_i, μx^s_j, $\sigma_{x^s_i}$, and $\sigma_{x^s_j}$ are the mean and standard deviations of nodes x_i and x_j. Finally, the weighted adjacency matrix is given as:

$$A^s_{i,j} = e^{p^s_{x^s_i,x^s_j}}, i \neq j. \tag{2}$$

After obtaining the sensor graph, i.e. g_s, the module graph can be constructed similarly. Usually, various modules of the complex machine system are well organized according to the preexisting documents, which means the relationship between different modules is also defined in advance. Different from the edge weight calculation among sensors, whose data have the same dimension, various sensors may be mounted on one module which will cause a dimension mismatch problem during the edge weight calculation. Therefore, to construct the module graph, the edge of the module graph is defined according to the structural information, and the mean of different sensor data of one module after normalization is considered as the node feature in the module graph. The calculation of edge weight between different modules is given as:

$$x^m_l = \frac{1}{E}\sum_{e=1}^{E} x^s_e \tag{3}$$

$$p^m_{x^m_l,x^m_k} = \frac{E\left(\left(x^m_l - \mu x^m_k\right)\left(x^m_l - \mu x^m_k\right)\right)}{\sigma_{x^m_l}\sigma_{x^m_k}} \tag{4}$$

$$A^m_{l,k} = e^{p^m_{x^m_l,x^m_k}}, l \neq k \tag{5}$$

where x^m_l is the node feature in the module graph, and E is the sensor number of module *l*. $p^m_{x^m_l,x^m_k}$ denotes the PCC between module *l* and module *k*, while $A^m_{l,k}$ is the weighted adjacency matrix of the module graph. Therefore, the module graph g_m can be obtained. Finally, with the constructed sensor graph and module graph, the hierarchical graph, i.e. $g_h = \{g_s, g_m\}$, can be constructed by taking the two graphs together as a unit. With the constructed hierarchical graph, the slide window technique is adopted to generate the hierarchical graph data set.

2.4. Node representation learning

To learn the representation of each node, a GNN-based representation learning method is proposed in this study. There are two key phases of node representation learning. The first one is to fuse the information from neighbor nodes with the graph structure and update the node representations of the graph. The other one is to perform graph pooling after the representation learning process, which can enhance the information flow and find meaningful nodes.

In the representation learning phase, the GCN can perform the information transfer on the graph like the traditional convolutional filters and has shown its superior performance in various graph-based machine learning tasks. Thus, the GCN is chosen for the representation learning in this study, and is given by:

$$Z = \check{D}^{-\frac{1}{2}}\check{A}\check{D}^{-\frac{1}{2}}XW \tag{6}$$

where Z is the updated information by fusing the information from its corresponding neighbor nodes, $\check{A}$ denotes the adjacency matrix, which is given by $\check{A} = A + 1$, and I represents the identity matrix. $\check{D}$ indicates the degree matrix of $\check{A}$, and W is the trainable weight matrix. X is the feature matrix of the graph, in which each row represents the feature vector of a node. Thus, the hidden representation of a single node can be updated from its neighbors and itself with equation (6). In addition, an activation function will be added after the information is transformed to get the representation of the node.

In the graph-pooling phase, the most important thing is to create a criterion for node selection. Generally, a node can be deleted during the pooling process if it can be generated by its neighbor nodes with less information loss at the same time. Based on this assumption, the node information score (NIS), which can measure the information of each node by calculating the correlation coefficient between the node representation and the one generated by its neighbor nodes, is introduced in this paper to guide the pooling. After obtaining the NIS of each node, the nodes with high NISs will be neglected because they are similar to neighbors and can be well generated by their neighbor nodes. The details of the pooling process are given as:

$$v = \rho\left(\acute{H}, H\right) \tag{7}$$

$$idx = \mathrm{top} - \mathrm{rank}\left(v, [r{*}n]\right) \tag{8}$$

$$H^p = H\left(idx, :\right) \tag{9}$$

$$A^p = A\left(idx, idx\right) \tag{10}$$

where equation (7) is used to calculate the NIS, in which ρ is the correlation coefficient calculation function, $\acute{H}$ is the representation obtained by aggregating the representations from neighbor nodes, H is the node representation, and v means the calculated NIS. Equations (8)–(10) are used to generate the graph after pooling, in which r is the preserved ratio, n denotes the indices of nodes, H^p is the node representation matrix after pooling, idx is a list of the preserved nodes, and A^p is the adjacency matrix after pooling.

To summarize, the graph representation can be updated as follows:

$$f(A,X) = Pooling\left(s\left(\tilde{A}XW\right)\right) \tag{11}$$

where $Pooling(\cdot)$ means the pooling function, which is helpful to find the meaningful nodes of the graph. $s(\cdot)$ is the activation function, such as ReLU, Tanh. To be clear, $\tilde{A}$ is represented by $\tilde{A} = \check{D}^{-\frac{1}{2}}\check{A}\check{D}^{-\frac{1}{2}}$. Thus, the updated nodes representation

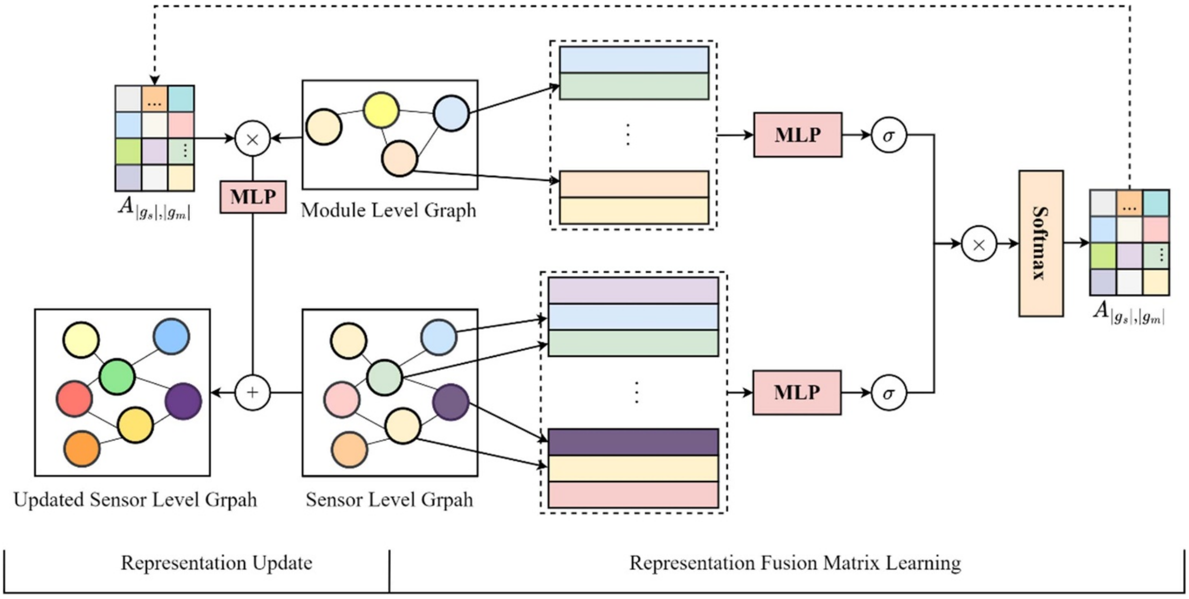


Figure 2. Structure of the adaptive cross-graph fusion block. MLP: multilayer perceptron.

of each graph can be obtained by following the given process above.

2.5. ACG-FB

Different level graphs together can provide comprehensive information about the degradation process, and it is necessary to consider them simultaneously during the prediction of RUL. To achieve information fusion across different graphs adaptively, an ACG-FB that can fuse the representation from one graph to another is proposed. For instance, the features of the module nodes in the module-level graph can potentially guide the feature learning of the sensor nodes in the sensor-level graph. The ACG-FB aims to infer the representation fusion from different level graphs adaptively and the details of the proposed representation fusion block are illustrated in figure 2.

The representation fusion matrix $A_{|g_s|,|g_m|} \in R^{|g_s|,|g_m|}$ is adopted to achieve representation fusion between different graphs, where $|g_s|$ denotes as the node number in the sensor-level graph and $|g_m|$ means the node number in the module-level graph. Each row of $A_{|g_s|,|g_m|}$ means the relative importance of the representations in the module-level graph to the representations in the sensor-level graph. The relationship between the nodes from the sensor-level graph and the nodes from the module-level graph is inferred as follows:

$$X_{g_s} = tanh\left(W_{g_s} X_{g_s} + b_{g1}\right) \tag{12}$$

$$X_{g_m} = tanh\left(W_{g_m} X_{g_m} + b_{g_m}\right) \tag{13}$$

$$A_{|g_s|,|g_m|} = softmax\left(X_{g_s} \cdot X_{g_m}^T\right) \in R^{|g_s|,|g_m|} \tag{14}$$

where $tanh(\cdot)$ is the nonlinear activation function, and $softmax(\cdot)$ means the softmax function operated along the row of the inner product of the matrix X_{g_s} and $X_{g_m}^T$, which is used to normalize the effects of features between two graphs. Equations (12) and (13) are used to update the features X_{g_s} and X_{g_m} from different graphs. Equation (14) is used to calculate the representation fusion matrix, where the element in the matrix is considered as the relative importance of the features in the sensor-level graph to the features in the module-level graph [20, 34]. Furthermore, different from the fixed-size single graph inference, the representation from different graphs can be adaptively fused with the cross-graph fusion block.

After generating the representation fusion matrix $A_{|g_s|,|g_m|}$, the representation from the module level graph can be fused into the sensor-level graph. Given the feature matrixes X_{g_s} and X_{g_m} from different graphs, the representation from the sensor-level graph can be updated as:

$$\check{x}_{g_s} = X_{g_s} + A_{|g_s|,|g_m|} X_{g_m} W \tag{15}$$

where $\check{X}_{g_s}$ is the updated representation of the sensor-level graph and W is the weight matrix that is trainable. Therefore, the information contained in different modules of the module-level graph can be adaptively fused into the corresponding sensors of the sensor-level graph.

As for the module graph, the representation in the sensor-level graph can enrich the information flow in the module-level graph. The representation of the sensor-level graph can also be fused into the module-level graph with the proposed ACG-FB. After obtaining the updated sensor-level graph and module-level graph, these two graphs are fed into the next GNN layer. With the proposed ACG-FB, the information in one graph can be adaptively fused into another graph, which can effectively utilize the information from different graphs and thus improve the performance of the RUL prediction tasks.

Table 1. Pseudo-code of the proposed HGNN-ACGF.

```
Input:
Dataset D = $\{(g_1^s, g_1^m, y_1),\ (g_2^s, g_2^m, y_2),\ \ldots,\ (g_n^s, g_n^m, y_n)\}$;
Number of GCN and ACG-FB: $H$;
Number of samples: N; Top-k pooling ratio: $k_1$, $k_2$;
Processing:
For $n \in \{1, 2, \ldots, N\}$ do:
    For $h \in \{1, 2, \ldots, H\}$ do:
        Learn the embedding from the node feature matrix $X_n^s$ and $X_n^m$ by GRU;
        Learn the new representation of the sensor graph $\acute{X}_n^s$ and module graph $\acute{X}_n^m$ with GCN;
        Calculate the NIS of different nodes and perform Top-k pooling with ratio $k_1$ on the sensor graph;
        Calculate the NIS of different nodes and perform Top-k pooling with ratio $k_2$ on the module graph;
        Calculate the representation fusion matrix matrices $A_{|g_s|,|g_m|}$ and $A_{|g_m|,|g_s|}$;
        Update module graph representation with sensor graph representation $\acute{X}_n^s$ and $A_{|g_s|,|g_m|}$ by ACG-FB;
        Update sensor graph representation with module graph representation $\acute{X}_n^m$ and $A_{|g_m|,|g_s|}$ by ACG-FB;
    End for
    Perform the Readout function on $\acute{X}_n^s$ and $\acute{X}_n^m$;
    Predict the RUL $\tilde{y}_n$ by MLPs;
End for
Output: $\tilde{Y}_i = \{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_N\}$
```

2.6. Prediction of RUL

After obtaining the updated representation of each node in the graph, the readout function is adopted to transform the node representations matrix of the graph into a single feature vector for RUL prediction. Thus, the RUL can be predicted through several stacked, fully connected layers with the concatenated representation from different graphs. The prediction process is formulated as:

$$Readout\,(H) = \sigma\left(\frac{1}{I}\sum_{i=1}^{I} h_i\right) \tag{16}$$

$$Rul = f([Readout\,(g_s)\,, Readout\,(g_m)]) \tag{17}$$

where σ is the activation function, I is the node number of the graph, and h_i means the feature vector of node i in equation (16). In equation (17), $f(\cdot)$ denotes the MLPs, g_s and g_m are the sensor graph and module graph, $Readout\,(\cdot)$ denotes the readout function which can transform the nodes' features matrix into the graph representation vector, and $[\cdot,\cdot]$ means the feature concatenation operation. Therefore, the representation of each sample can be learned by the GNN with pooling and ACG-FB. Meanwhile, the corresponding RUL can be predicted by various fully connected layers. In addition, the mean square error (MSE) is chosen as the loss function to optimize the parameters of the proposed HGNN-ACGF, which is given as:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i)^2 \tag{18}$$

where $\tilde{y}_i$ and y_i are the predicted result and the real label, respectively, θ means the trainable parameters, and N is the number of samples. The total process of the proposed method is illustrated in table 1.

3. Experimental setup

3.1. Experimental data set

The experimental data sets were collected from C-MAPSS which belongs to a turbofan engine simulation program developed by NASA [31]. The simulation system consists of various modules, such as a fan-speed controller, regulators, and limiters. The comprehensive logic structure of various modules assembled in the system is the same as the engine used in practice. Figure 3 shows the logical structure of these modules and corresponding sensors. Each module has one or more sensors.

In detail, (a) means the original logical structure of the module and (b) denotes the logical structure after the invalid sensors are removed. In this data set, four different data subsets, namely, FD001, FD002, FD003, and FD004, were generated under different operating conditions. All four subsets are adopted to evaluate the effectiveness of the proposed method. The FD001 and FD003 data subsets have a single operating condition while FD002 and FD004 have six operating conditions according to altitudes, sea-level temperatures, and Mach numbers.

In addition, each data subset has one (high pressure compressor (HPC) degradation) or two (HPC degradation and fan degradation) fault modes. The detailed information of these four data subsets is described in table 2. For example, the data subset FD001 is divided into the training set and testing set, reordering three operating conditions measurements and 21 sensor channels to characterize the fault evolution. The training set had trajectories that ended at the failure threshold, while the testing set was pruned to stop sometime before the failure threshold. In the training set, train_FD001, the run-to-failure data of multiple turbofan engines are recorded, and each engine trajectory signifies each engine life cycle. In the testing set, test_FD001, the time series from multi-sensors are pruned sometime before system failure, which means the

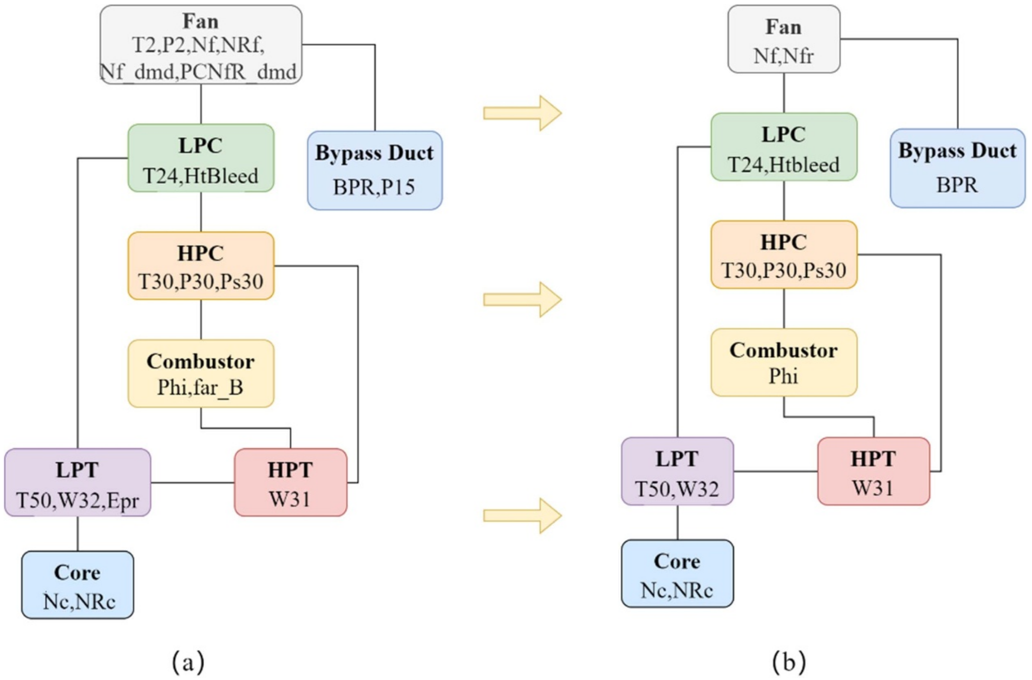


Figure 3. The logical structure of different modules. (a) Before removal of invalid sensors; (b) after removal of invalid sensors.

Table 2. Detailed description of C-MAPSS data subsets.

Data subset	FD001	FD002	FD003	FD004
Operating conditions	1	6	1	6
Fault modes	1	1	2	2
Train trajectories	100	260	100	249
Maximum lifespan (cycles)	362	378	525	543
Minimum lifespan (cycles)	128	128	145	128
Test trajectories	100	259	100	248

engine trajectory in the testing set is not run-to-failure recording. The RUL of each engine in the testing sets is recorded in RUL files, e.g. RUL_FD001. The other three data subsets are similar to FD001 except that FD002 and FD004 have more complex operating conditions.

3.2. Evaluation metrics

To evaluate the performance of the proposed method in this paper, two commonly used evaluation metrics, namely root MSE (RMSE) and the scoring function (SF), which were used in the 2008 international conference on PHM data challenge, are adopted [31]. According to [35], the calculation formulas of RMSE and SF are given as equations (19) and (20):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\Delta R_i{}^2} \tag{19}$$

$$\mathrm{score} = \begin{cases} \sum\limits_{i=1}^{N}\exp\left(-\frac{\Delta R_i}{13}\right) - 1, \Delta R_i < 0 \\ \sum\limits_{i=1}^{N}\exp\left(\frac{\Delta R_i}{10}\right) - 1, \Delta R_i \geqslant 0 \end{cases} \tag{20}$$

where N is the sample number, and $\Delta R_i = \widetilde{RUL}_i - RUL_i$ means the difference between the predicted $\widetilde{RUL}_i$ and true RUL_i of the ith sample. In detail, the RMSE assigns equal weight to both early and late predictions and has been widely adopted in regression tasks. Meanwhile, the score function gives more penalization to late prediction as this may result in a huge disaster in industrial applications.

3.3. Experimental procedure

3.3.1. Data preprocessing. The C-MAPSS data set contains multiple time-series data which are collected from 21 sensors. The detailed definition of different sensors is illustrated in table 3.

For each data subset from C-MAPSS, the z-score normalization method is used to normalize the time-series data generated from the sensors [36], which is given as follows:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i} \tag{21}$$

where i represents ith sensor, μ_i and σ_i are the mean value and corresponding standard deviation of the ith sensor, and x' is the normalized sensor data.

According to previous studies [37, 38], only the sensor data with monotonic trends can provide valuable information about the degradation and have a significant effect on RUL prediction. Therefore, those sensors such as 1, 5, 6, 10, 16, 18, and 19, with irregular/unchanged values are removed, while the other 14 sensors with monotonic increasing/decreasing trends are retained, based on the degradation trends of 21 sensors shown in table 4.

The visualization of the raw sensor data and the normalized sensor data from 14 sensors of FD001 are shown in figure 4.

Table 3. Detailed description of each sensor.

No.	Sensor	Description	No.	Sensor	Description
1	T2	Total temperature at fan inlet	12	Phi	Ratio of fuel flow to Ps30
2	T24	Total temperature at LPC outlet	13	NRf	Corrected fan speed
3	T30	Total temperature at HPC outlet	14	NRc	Corrected core speed
4	T50	Total temperature at LPT outlet	15	BPR	Bypass ratio
5	P2	Pressure at fan inlet	16	farB	Burner fuel:air ratio
6	P15	Total pressure in bypass duct	17	htBleed	Bleed enthalpy
7	P30	Total pressure at HPC outlet	18	Nf_dmd	Demanded fan speed
8	Nf	Physical fan speed	19	PCNfR_dmd	HPT coolant bleed
9	Nc	Physical core speed	20	W31	HPT coolant bleed
10	Epr	Engine pressure ratio	21	W32	LPT coolant bleed
11	Ps30	Static pressure at HPC			

Note: LPC refers to low pressure compressor; LPT refers to low pressure turbine.

Table 4. Degradation trends of 21 sensors.

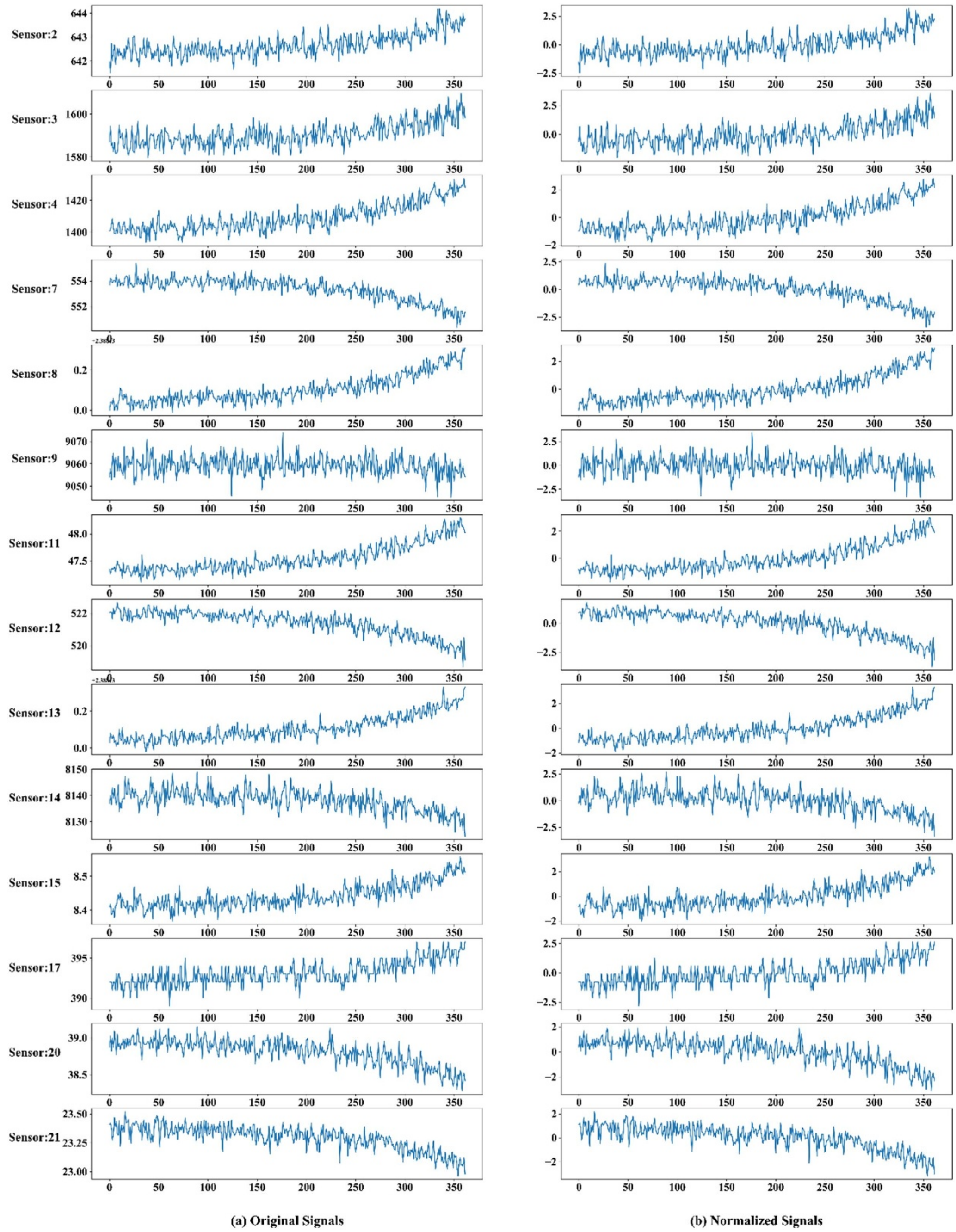
Trend type	Sensor number
Increasing	2, 3, 4, 8, 9, 11, 13, 15, 17
Decreasing	7, 12, 14, 20, 21
Irregular/unchanged	1, 5, 6, 10, 16, 18, 19

Specifically, (a) is the raw sensor data, and (b) means the normalized sensor data. The degradation trend can be seen from the figures. Besides, a piece-wise linear degradation model has been validated to be suitable to obtain the RUL label for this data set [39], which assumes that the engine has a constant RUL label in the initial period, then degrades linearly to 0. There is no doubt that the initial RUL label has a noticeable effect on the performance of the RUL prediction. Therefore, 125 is chosen as the initial constant RUL label for FD001 and FD003, and 150 for FD002 and FD004 according to previous studies [13, 40]. Furthermore, according to [40], the size of the slide time window is set to 30, 20, 30, and 15 because the minimum number of record cycles of four testing data sets is 31, 21, 38, and 19.

Then the datasets for training and testing are prepared by slide time window processing, in which each instance contains the sequential information within the time window of p length. At each time step, the sensor graph is constructed with the historical data from multiple sensors and the module graph is generated with the historical sensor data and the structural information, within the time window. The process of slide time window processing is shown in figure 5.

3.3.2. Hierarchical graph data set construction. To construct the hierarchical graph data set, the sensor graph which contains the features of each sensor and the relationship between different sensors should be constructed first. For the multivariate temporal sensor data set, each sensor is considered as a node in the sensor graph, and the weight of edges between different nodes can be defined as the correlation coefficient between sensors. Since the sensors mounted on the machine will record various types of data, such as temperature, pressure, and vibration, these data have different data intervals and feature spaces. Some traditional distance metrics such as Euclidean distance cannot adjust to such characteristics of data. Thus, the PCC is adopted to calculate the relationship between different sensors. After obtaining the sensor graph, the module graph can be constructed similarly. One thing to notice is that multi-sensors mounted on a single module will cause a feature dimension mismatch problem when calculating the correlation coefficient. Therefore, the features of modules should be converted to the same dimension before the calculation. To simplify the calculation process, the mean value of the sensor data obtained from a single module is denoted as the module feature vector. The constructed sensor graph and module graph are illustrated in figure 6. The sensor graph contains 14 nodes that are connected with different weights. For the module graph, eight modules are represented by eight nodes, and their connection is determined by the predefined structure between different modules, which has been given in figure 3.

3.3.3. Comparison of methods and model training. Based on the hierarchical graph data set, the proposed deep learning model HGNN-ACGF for RUL prediction can be trained. The experiments are implemented by Pytorch 1.9 and conducted on the NVIDIA T4 GPU with 16G memory and an Intel I7-7800K CPU. For better model training, some deep learning training tricks are introduced, such as the early stopping technique and dropout strategy. Adam is chosen as the optimization method and the learning rate decay strategy is adopted. Finally, the RUL of samples in the testing set can be estimated by the trained model. Additionally, to validate the performance of the proposed method more comprehensively, traditional machine learning methods such as SVR and typical deep learning methods, including the DNN with five fully connected layers, the CNN with three convolution layers and two fully connected layers, and LSTM with two LSTM layers and two fully connected layers, are chosen for comparison. In addition, some methods from published studies are also selected to validate the effectiveness of the proposed method, such as deep belief network (DBN) [41], deep convolutional neural network (DCNN) [42], bidirectional LSTM based autoencoder (BiLSTM-ED) [36], multicellular LSTM (M-LSTM) [24], multiobjective deep belief networks ensemble (MODBNE) [41], attention-based deep learning approach (ABDLA) [43], and times series memory auto-encoder with sequentially updated reconstructions (SUR-TSMAE) [44]. Because the



(a) Original Signals

(b) Normalized Signals

Figure 4. The raw data and the normalized data from 14 sensors.

four data subsets have different operating conditions the parameters for FD001 and FD003, with single operating conditions, and FD002 and FD004, with six operating conditions, are different. To determine the appropriate hyperparameters, a grid search is adopted. In the training stage, the data set for model training is divided into two parts, i.e. the training set and the validation set, first. Then, different hyperparameter sets, e.g. [16, 32, 64, 128, 256] for the batch size and [0.0001, 0.001, 0.01, 0.1] for the initial learning rate, are defined in advance. The proposed method is trained with hyperparameters from the hyperparameter sets on the training set and validated on the validation set. Finally, the hyperparameters with the best performance on the validation set are selected as the default hyperparameters of the proposed method after numerous experiments. The detailed hyperparameters used in the proposed method are given in table 5.

4. Results and discussion

4.1. Experimental results

4.1.1. Main results. The results of the proposed method and comparison methods following the experiments conducted on the CMAPSS data set are summarized in table 6.

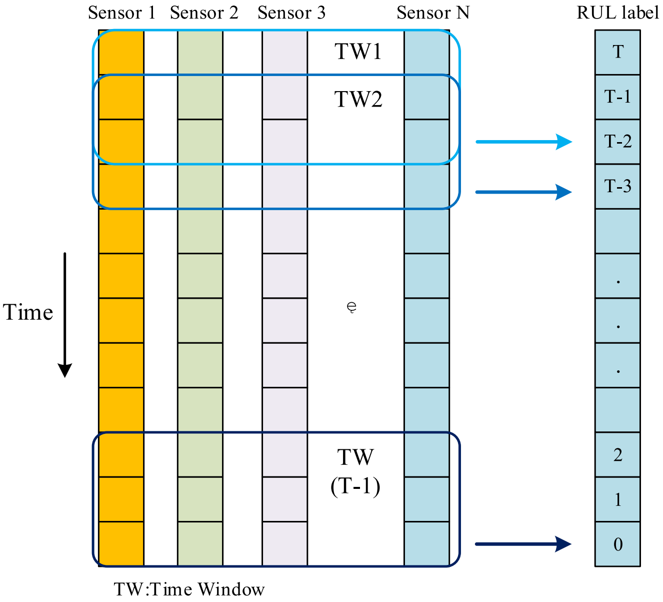


Figure 5. The process of time window processing.

It is clear in table 6 that the proposed method achieves superior RUL prediction performance, which is 12.58 on FD001, 21.67 on FD002, 12.40 on FD003, and 22.43 on FD004. It can also be seen that all the methods perform well on FD001 and FD003, but worse on FD002 and Fd004. The reason is that FD002 and FD004 have more complex operating conditions, which have a big negative influence on RUL prediction. In detail, it is obvious that the SVR performs worst on the four data subsets compared with other deep learning-based methods, because of its limited ability in degradation trend information modeling. The DNN performs better than the CNN and LSTM on FD001 but worse on the other three data subsets. The reason is that the data of FD001 is easy to analyze while the other three are more complex, and DNN has limited ability to handle these complex data. The LSTM performs better than the CNN on the four data subsets, which indicates the usefulness of the sequence models in RUL prediction. As for the comparison of other deep learning methods, the proposed method also achieves superior performance because it can model the sensor data and the relationship between sensors and modules simultaneously. The comparison of the predicted RUL values and the ground truth RUL of each engine from four data subsets are given in figure 7, which is consistent with the results shown in table 6 and indicates the effectiveness of HGNN-ACGF on RUL prediction.

It can be observed that the proposed HGNN-ACGF has promising RUL prediction performance according to the results in table 6. Consequently, the reliability and safety of the machine can be improved based on the RUL prediction results.

4.1.2. Robustness analysis. The performance of the proposed method may be influenced by different initial parameters and data sets. Specifically, to verify the robustness of the proposed method, more experiments were conducted on the four data subsets and repeated 20 times with different initial parameters. The corresponding variance of RMSE and SF from the 20 repetitions are shown in table 7.

Table 7 shows that the proposed HGNN-ACGF has the lowest variance of RMSE and SF among these methods because it can introduce the structural information as prior knowledge while modeling. What's more, from the results, it is found that the typical deep learning methods have weak robustness compared with traditional machine learning methods. The reason may be that deep learning methods are more prone to overfitting. In addition, the results also reveal that the proposed HGNN-ACGF has better robustness compared with other typical deep learning methods.

4.2. Discussion

4.2.1. Evaluation of different slide window sizes. The length of window size will influence the degradation trend information of monitoring data in the temporal domain, and thus influence the prediction performance of the proposed method. To evaluate the influence of window size on the proposed HGNN-ACGF method, different window sizes are adopted on the four data subsets. Due to the limitation of maximum sequence length in testing sets, the maximum length of FD001 to FD004 is set to 30, 20, 30, and 15. The detailed results of RMSE and SF of the proposed method with different slide window sizes on FD001 to FD004 are shown in figure 8.

As shown in figure 8, on increasing window size, the performance of the proposed HGNN-ACGF also increases. The reason is that more degradation information contained in the sample can be obtained when the window size is larger. What's more, these four data subsets have various operating conditions: FD001 and FD003 have a single operating condition while FD002 and FD004 have six. It can be seen from the results of FD001 to FD004 that the complex operating conditions have much more effect on the performance of the proposed method. Specifically, when the window size increases from 10 to 15, the RMSE of the proposed HGNN-ACGF decreases from 27.42 to 22.43 on FD004, while decreasing from 27.87 to 25.67 on FD002. The reason is that compared with FD002, the degradation information of FD004 is easier to obtain from the same window size.

4.2.2. Evaluation of the number of GCN-ACG-FB. The GCN and ACG-FB together can be considered as a basic unit, i.e. GCN-ACG-FB, of the proposed HGNN-ACGF, and its number will influence the performance of RUL prediction. To validate the influence The experiments were conducted on the four data subsets with different operating conditions, which can make the results more robust. The experimental results of RMSE and SF are shown in figure 9.

It can be observed that the RMSE and SF of the proposed method decrease first and then increase with the increasing number of GCN-ACG-FB, and the best RMSE and SF are obtained when the number is 2, which means more layers

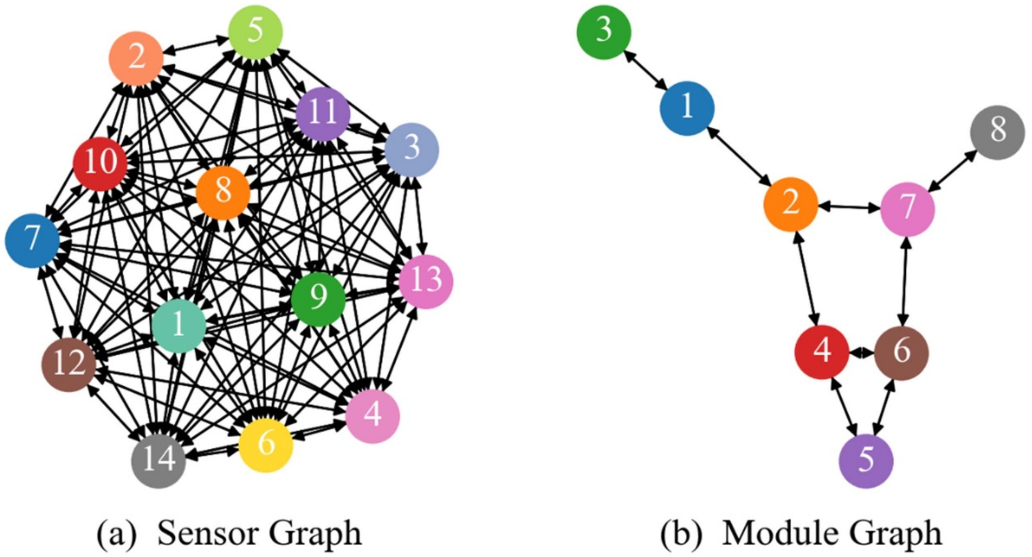


Figure 6. The constructed hierarchical graph. (a) Sensor graph with 14 nodes; (b) module graph with 8 nodes.

Table 5. Hyperparameters used in the proposed method.

Hyperparameter	FD001 & FD003	FD002 & FD004
Batch size	64	64
Initial learning rate	0.001	0.001
Epoch	200	200
Dropout rate	0.2	0.2
Num of GCN layers	2	2
Embedding size of GRU	64	64
Node representation size in GCN	64	64
Settings of MLP layers	128/64/1	128/64/64/1
Preserved ratio of pooling	0.8	0.9

are useful to improve the performance for RUL prediction. However, with the number of GCN-ACG-FB increasing and reaching a certain number, the proposed method may suffer oversmoothing and overfitting problems, which will make the nodes similar to each other and have a negative influence on the final RUL prediction performance. What's more, the RMSE on FD001 and FD003 is less than that on FD002 and FD004 with the same GCN-ACG-FB number, which indicates that compared with the single operating condition, the degradation information for the complex operating conditions is harder to extract. Besides, one thing should be noticed: the influence of GCN-ACG-FB number on FD001 and FD003 is less than that on FD002 and FD004. In detail, the difference between the maximum and minimum of RMSE are 4.92 and 3.31 on FD002 and FD004, respectively, while they are 1.27 and 2.22 on FD001 and FD003, respectively. This indicates complex operating conditions have a larger influence on the performance of RUL prediction.

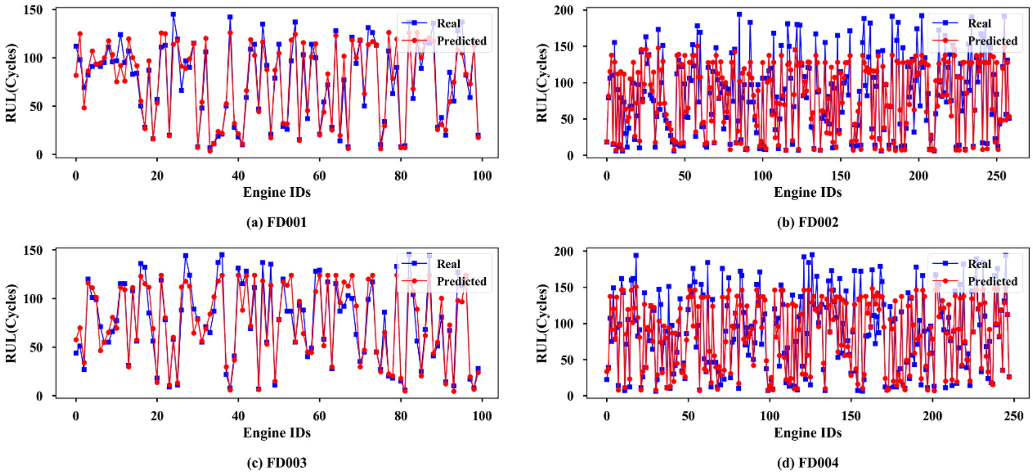
4.2.3. Ablation study of the proposed method. Each part of the proposed HGNN-ACGF has a certain contribution to the final RUL prediction results. To validate which part of the proposed method contributes most to the final RUL prediction results, four variants of the original model are adopted and validated, which are shown in table 8.

In detail, Models A, B, C, and D are acquired by removing GRU, ACG-FB, the module graph part, or the sensor graph part from the HGNN-ACGF, respectively. All the hyperparameters settings adopted are the same as the experiments conducted before and the RMSE and SF of these models on four data subsets are shown in figure 10.

Firstly, it can be seen from the results, Model A has the poor prediction performance, i.e. 13.66 on FD001, 25.87 on FD002, 13.31 on FD003, and 24.84 on FD004, among these variant models, due to its insufficient ability to model the temporal information of the multi-sensor data. Secondly, Model B does not have the ACG-FB for information fusion across different graphs, and the information from multi-graphs is concatenated after the readout function without interacting with each other, which results in the worst performance, i.e. 14.11 on FD001, 26.51 on FD002, 14.43 on FD003, 26.23 on FD004, among the models. The reason is that the information contained in multi-graphs is not well considered during information processing, which proves the usefulness of the proposed ACG-FB in representation fusion across different graphs. Thirdly, compared with Model D, Model C performs better, i.e. −0.14 on FD001, 1.06 on FD002, 0.21 on FD003, and 0.97 on FD004, due to the rich information contained in the sensor-level graph. Besides, the comparison results prove that the module graph with structural information is also useful for RUL prediction. It is interesting that Model D performs better than Model C on FD001 but worse on the other three data subsets. The reason may be that under the single operating condition, the structural information in the module graph can better reflect the degradation trend information. Model E is the proposed method, which can extract degradation trend information and fuse the information across graphs simultaneously, as well as achieve the best performance for RUL prediction. And the results prove the effectiveness of the proposed ACG-FB in handling the

Table 6. Experimental results of four testing data subsets.

	FD001		FD002		FD003		FD004	
Method	RMSE	SF	RMSE	SF	RMSE	SF	RMSE	SF
SVR	18.28	1004.75	30.50	17 132.17	21.37	2084.75	34.11	15 740.27
DNN	14.90	370.14	28.85	14 733.60	13.94	323.09	31.60	16 581.07
CNN	15.84	374.65	28.73	11 394.12	13.53	318.93	30.66	14 334.14
LSTM	14.91	333.80	28.68	12 908.59	13.37	316.55	30.32	11 771.06
DBN [41]	15.21	417.59	27.12	9031.64	14.71	442.43	29.88	7954.51
DCNN [42]	12.61	273.70	22.36	10 412.00	12.64	284.10	23.31	12 466.00
BiLSTM-ED [36]	14.74	273.00	22.07	3099.00	17.48	574.00	23.49	3202.00
M-LSTM [24]	13.71	315.00	N/A	N/A	N/A	N/A	23.81	4826.00
MODBNE [41]	15.04	334.20	25.05	5585.30	12.51	421.90	28.66	6557.60
ABDLA [43]	14.53	322.44	N/A	N/A	N/A	N/A	27.08	5649.14
SUR-TSMAE [44]	14.46	**200.00**	**21.10**	**1383.00**	17.16	370.00	22.61	**2602.00**
HGNN-ACGF	**12.58**	218.04	21.67	4584.97	**12.40**	**248.47**	**22.43**	2737.86



(a) FD001

(b) FD002

(c) FD003

(d) FD004

Figure 7. The predicted RUL and the ground-truth RUL of four data subsets.

Table 7. The variance of the predicted RUL compared with other methods.

	FD001		FD002		FD003		FD004	
Method	RMSE	SF	RMSE	SF	RMSE	SF	RMSE	SF
SVR	0.24	16.2	0.34	684	0.24	19.5	0.23	524
DNN	0.25	18.4	0.48	894	0.39	21.7	0.26	814
CNN	0.23	26.1	0.35	496	0.16	29.4	0.46	865
LSTM	0.59	43.5	0.43	791	0.32	31.3	0.41	976
HGNN-ACGF	**0.09**	**6.2**	**0.13**	**271**	**0.11**	**7.3**	**0.15**	**365**

hierarchical graphs. Finally, the results from figure 10 prove that the information from the hierarchical graphs is useful to enhance the predictive ability of the proposed method if the information from different graphs can be fused adaptively.

4.2.4. Evaluation of the preserved ratio of NIS pooling. The preserved ratio of NIS pooling can control the information after the pooling process; a smaller preserved ratio can improve the efficiency of the proposed HGNN-ACGF, while a bigger preserved ratio can keep more information. Therefore, it is reasonable to evaluate the influence of the preserved ratio on RUL prediction performance, which can be used to balance computing efficiency and RUL prediction performance. Therefore, the performance under different preserved ratios k is discussed in this section. Specifically, the preserved ratio k is set from 0.5 to 1.0 to evaluate the performance of the proposed method on four data subsets, and the corresponding RMSE and SF are shown in figure 11.

It can be seen from the results that the best RUL prediction results are achieved at 0.8, 0.9, 0.8, and 0.9 on FD001, FD002, FD003, and FD004 with 12.58, 21.67, 12.40, and 22.43,

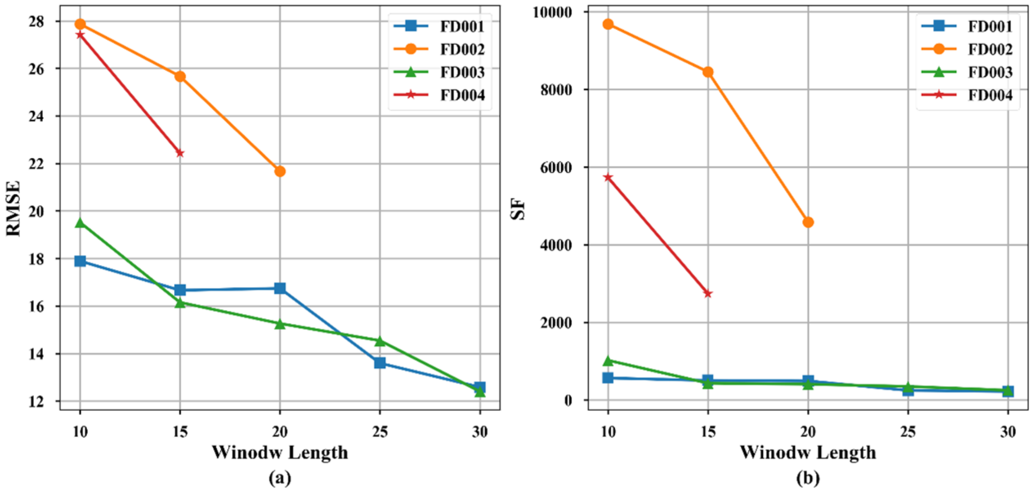


Figure 8. Experimental results for different window sizes. (a) RMSE; (b) SF.

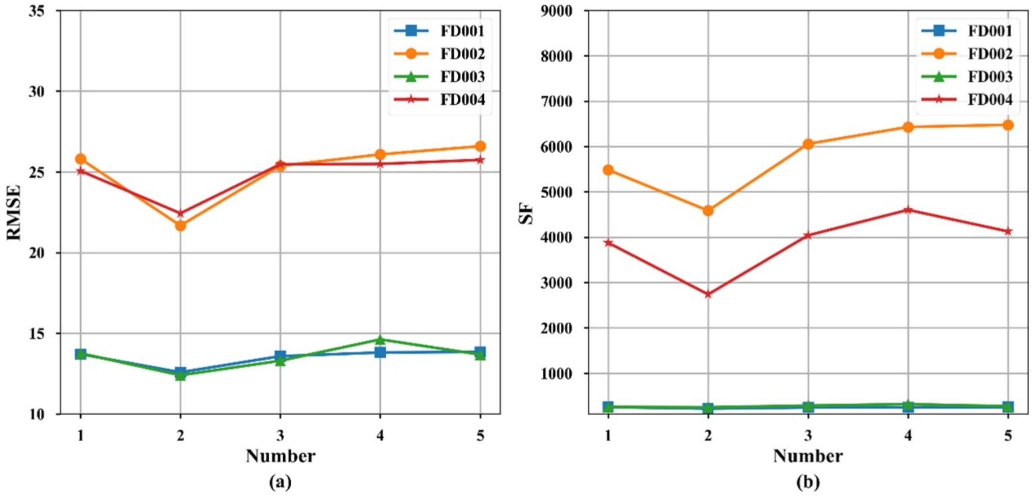


Figure 9. The experimental results from different data subsets with a different number GCN-ACG-FBs. (a) RMSE; (b) SF.

Table 8. Ablation model description.

Model	Description
Model A	HGNN-ACGF without GRU
Model B	HGNN-ACGF without ACG-FB
Model C	HGNN-ACGF without module graph
Model D	HGNN-ACGF without sensor graph
Model E	HGNN-ACGF

respectively. FD001 and FD003 have a smaller preserved ratio than FD002 and FD004. The reason is that the degradation trend of FD002 and FD004 is not clear enough, and larger k can preserve more degradation information from data in the pooling process. FD001 and FD003 have the simple operating condition and their degradation trend is easy to for the proposed method to capture even with a smaller preserved ratio. What's more, based on the results of different preserved ratios, 10, 12, 10, and 12 nodes from FD001 to FD004 are retained. In conclusion, the results in figure 10 prove that the proposed NIS pooling can be a useful approach to guide the node selection and improve the information flow in RUL prediction.

4.2.5. Visualization of the fusion matrix in ACG-FB. In order to visualize the relative importance among different sensors and modules, and evaluate the necessity of the proposed ACG-FB, the relative importance of different sensors and modules are shown in figures 12 and 13. In detail, figure 12 shows the relative importance of module features to sensors in the four data subsets, while figure 13 shows the relative importance of different sensor features to the modules. For better presentation, the representation fusion matrix in figure 12 is transposed. Because the preserved ratios of different data subsets are different, the representation fusion matrix of FD001 and FD003 is 10 × 6, and FD002 and FD004 is 12 × 8. Four sensors (T24, T30, Nf, bypass ratio (BPR)) and two modules (Core Nozzle, Bypass Ratio) are removed in FD001 and FD003, while two sensors (Nf, BPR) are removed from FD002 and FD004 in the pooling.

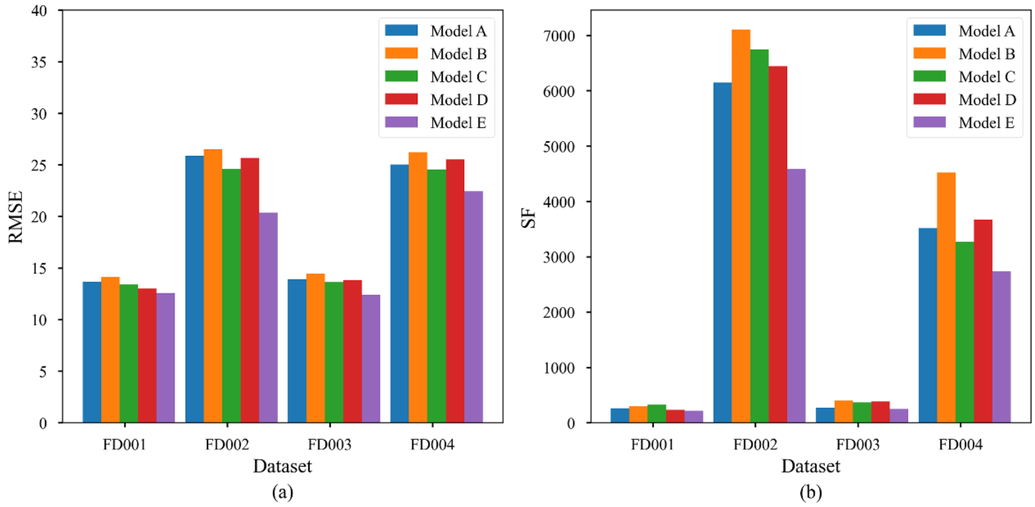


Figure 10. Experimental results for the ablation study. (a) RMSE; (b) SF.

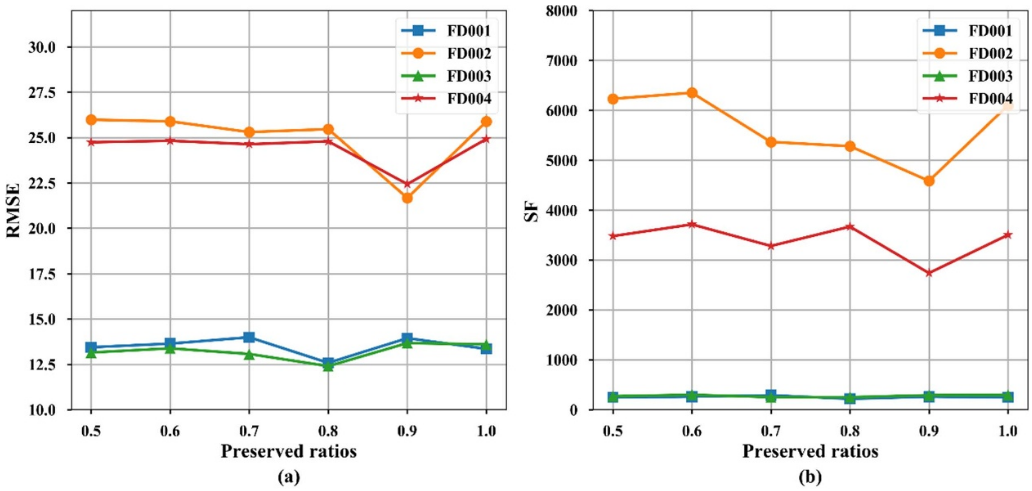
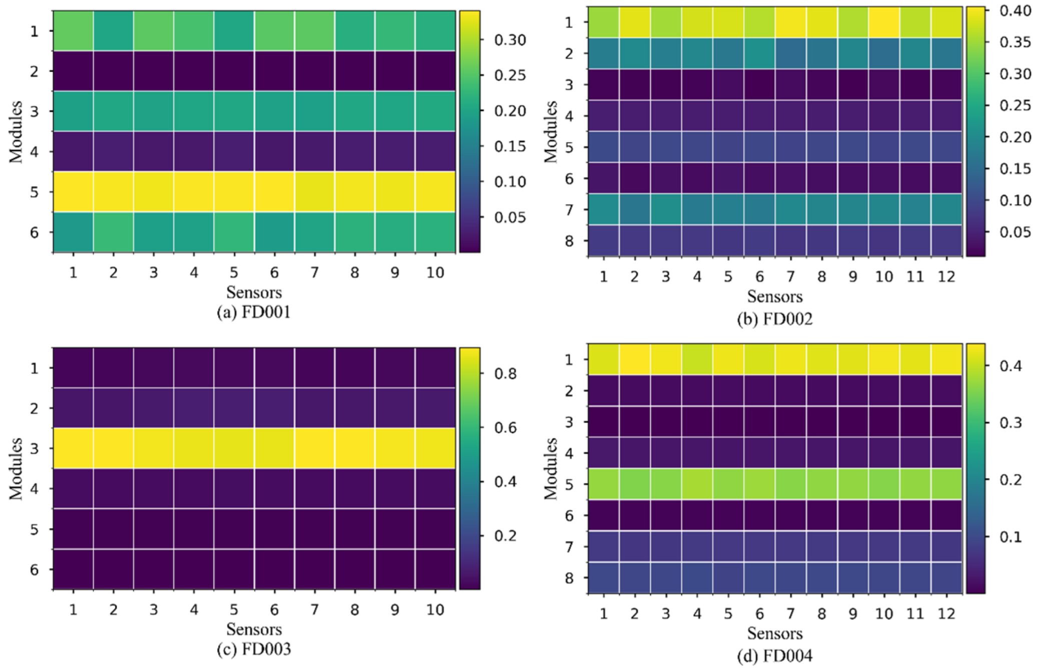


Figure 11. Evaluation of different preserved ratios on four data subsets. (a) RMSE; (b) SF.

In FD001 and FD003, it is clear that the modules, i.e. Fan, HPT, HPC, and LPT, have a larger weight when fused to the sensor graph, while the modules LPC and Combustor have less influence during the fusion. In addition, when fusing the sensor graph representation to the module graph, sensors Nc, W31, and W32 are more important compared with other sensors. The reason may be that these modules and sensors contain more degradation trend information and have easily been assigned larger weights during the training for producing improved RUL prediction performance. In FD002 and FD004, modules Fan, Combustor, and LPT contain more useful information for the RUL prediction, while modules Bypass Duct and HPT have the least contribution to the final prediction results, when fusing the module graph representation to the sensor graph. What's more, sensors T24, T50, P30, and NRf have relatively larger weights when fusing to the module graph, and sensor T30 has less influence in the fusion process. The reason is that the degradation trend of FD002 and FD004 is not clear enough under the complex operating conditions, and more sensor data is needed during the model training compared with FD001 and FD003. In figure 12, the weights of different modules in FD001 and FD003 have large differences, while the weights of different modules in FD002 and FD004 are more balanced. The reason may be that the degradation trend of FD001 and FD003 is clear and only the modules with a lot of degradation trend information are needed, and the other modules may be neglected with small weights. Finally, from figures 12 and 13, it can be seen that different sensors and modules are assigned with different weights, showing their distinctive ability in the RUL prediction task. This also illustrates the necessity of the proposed ACG-FB, in which different sensors and modules are fused with specific weights. In summary, through the proposed ACG-FB, the interpretability of the proposed method can be improved, which is shown in the two figures.

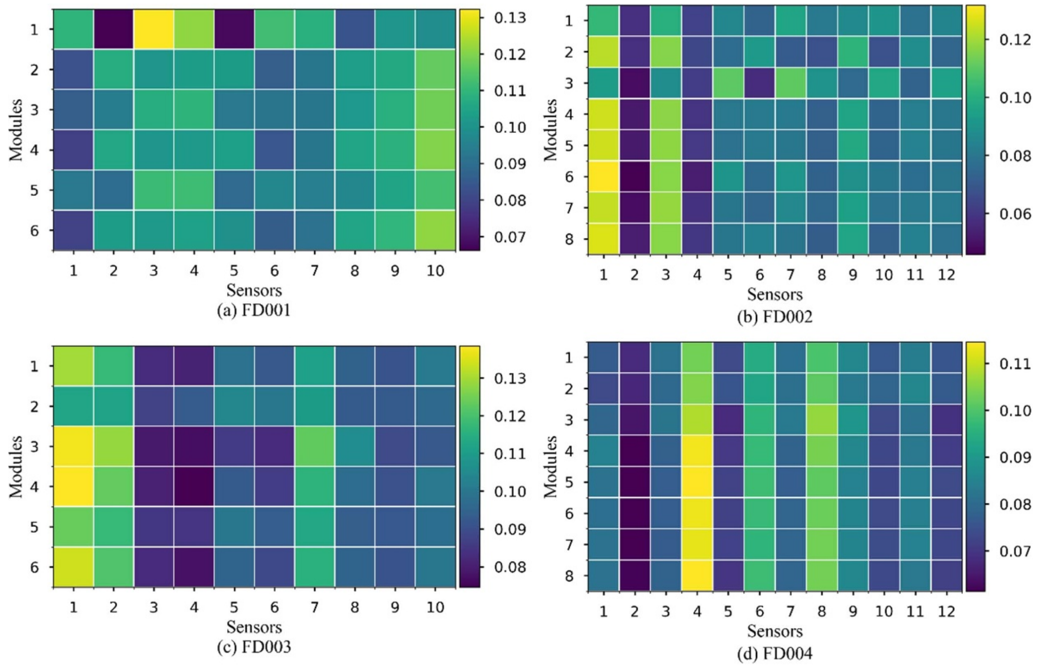
4.2.6. Evaluation of the node representation size in GCN.

The node representation size in GCN has a significant influence on the RUL prediction performance. A smaller node representation size may reduce the complexity of the model and make the network convergence faster, while a bigger node



(a) FD001
(b) FD002
(c) FD003
(d) FD004

Figure 12. Illustration of the representation fusion matrix for module graph to sensor graph. (a) FD001; (b) FD002; (c) FD003; (d) FD004.



(a) FD001
(b) FD002
(c) FD003
(d) FD004

Figure 13. Illustration of the representation fusion matrix for sensor graph to module graph. (a) FD001; (b) FD002; (c) FD003; (d) FD004.

representation size can improve the model's ability for learning more information from the signal data. On this account, the RUL prediction performance with different node representation sizes l on four data subsets is discussed in this section. Specifically, the node representation size l is set from 16 to 256 to evaluate the performance of the proposed method, and the corresponding RMSE and SF are shown in figure 14.

It is clearly shown in figure 14 that the performance of the proposed HGNN-ACGF increases as the node representation size increases from 16 to 64 and then decreases. The best performance of the node representation size is achieved at 64. The reason is that the original input sample size is 30, 20, 30, and 15 in FD001, FD002, FD003, and FD004, respectively, and small node representation sizes have limited ability to

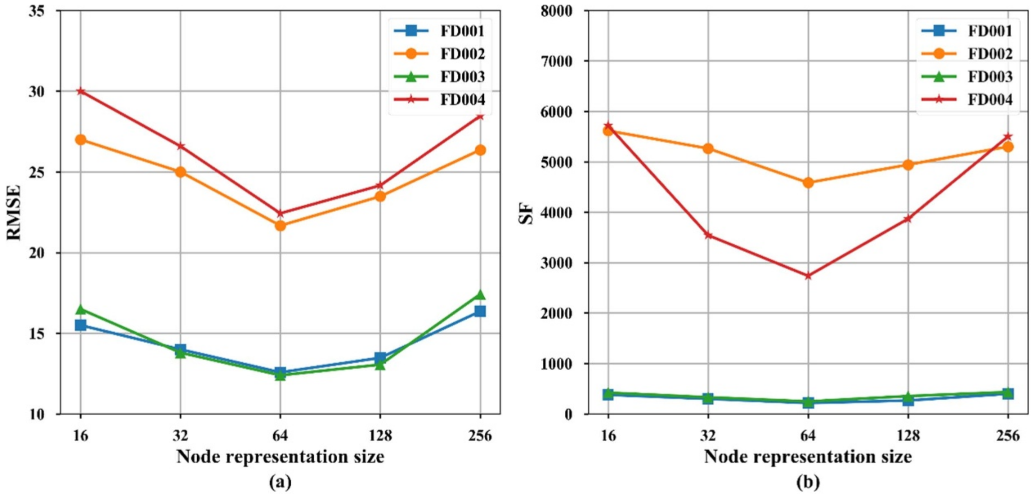


Figure 14. Evaluation of different node representation sizes on four data subsets. (a) RMSE; (b) SF.

learn high-quality representation. The RMSE and SF of different node representation sizes on FD001 and FD003 are better than those on FD002 and FD004 due to the influence of complex operating conditions in FD002 and FD004. Besides, with the node representation size becoming larger and larger, the proposed HGNN-ACGF may suffer from the overfitting problem. What's more, compared with FD001 and FD002, the fluctuation of RMSE and SF with different node representation sizes on FD003 and FD004 is larger: the reason is that FD003 and FD004 have more fault modes than FD001 and FD002. Based on the experimental results above, the node representation size is set to 64 as the default setting on four data subsets.

5. Conclusions and future work

In this paper, an HGNN-based method is proposed to improve the performance of RUL prediction for complex machine systems. The hierarchical graph, consisting of a sensor graph and a module graph, is constructed by introducing the sensor data and structural information from the complex machine system. Then, an HGNN is adopted to model these graphs, and thus predict the RUL. Meanwhile, the degradation trend information contained in different level graphs is considered and fused simultaneously. To verify the proposed method, data sets from C-MAPSS were adopted, and the experimental results show superior performance of the proposed method compared with other methods.

In addition, some recommendations should be noted for further research directions. Firstly, there is no paradigm for the construction of the graph for the RUL prediction and it is vital to find a proper method for the construction of the graph. Secondly, the information from different level graphs is fused by the proposed ACG-FB and it would be interesting to extend the ACG-FB and integrate more graphs. Thirdly, the SF is reasonable in the real application for RUL prediction, so integrating the SF with the loss function of the deep learning methods may also be a promising research direction.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (72071062), the Science Fund for Distinguished Young Scholars of Anhui (2208085J12), the Anhui Provincial Key Research and Development Program (202104a05020038), and the Fundamental Research Funds for the Central Universities (PA2021KCPY0032).

ORCID iDs

Gang Wang https://orcid.org/0000-0002-6395-9409
Zhangjun Wu https://orcid.org/0000-0003-2414-5768

References

[1] Lin Y, Li X and Hu Y 2018 Deep diagnostics and prognostics: an integrated hierarchical learning framework in PHM applications *Appl. Soft Comput.* **72** 555–64

[2] Wang G, Zhang F, Cheng B and Fang F 2021 DAMER: a novel diagnosis aggregation method with evidential reasoning rule for bearing fault diagnosis *J. Intell. Manuf.* **32** 1–20

[3] Duan Y, Li H, He M and Zhao D 2021 A BiGRU autoencoder remaining useful life prediction scheme with attention mechanism and skip connection *IEEE Sens. J.* **21** 10905–14

[4] Deutsch J and He D 2018 Using deep learning-based approach to predict remaining useful life of rotating components *IEEE Trans. Syst. Man Cybern. Syst.* **48** 11–20

[5] Liao L and Köttig F 2016 A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction *Appl. Soft Comput.* **44** 191–9

[6] Żuławiński W, Maraj-Zygmąt K, Shiri H, Wyłomańska A and Zimroz R 2023 Framework for stochastic modelling of long-term non-homogeneous data with non-Gaussian characteristics for machine condition prognosis *Mech. Syst. Signal Process.* **184** 109677

[7] Qiu G, Gu Y and Chen J 2020 Selective health indicator for bearings ensemble remaining useful life prediction with genetic algorithm and Weibull proportional hazards model *Measurement* **150** 107097

[8] Cai B, Shao X, Liu Y, Kong X, Wang H, Xu H and Ge W 2019 Remaining useful life estimation of structure systems under the influence of multiple causes: subsea pipelines as a case study *IEEE Trans. Ind. Electron.* **67** 5737–47

[9] Ren L, Cui J, Sun Y and Cheng X 2017 Multi-bearing remaining useful life collaborative prediction: a deep learning approach *J. Manuf. Syst.* **43** 248–56

[10] Wang G, Huang J and Zhang F 2021 Ensemble clustering-based fault diagnosis method incorporating traditional and deep representation features *Meas. Sci. Technol.* **32** 095110

[11] Ferreira C and Gonçalves G 2022 Remaining useful life prediction and challenges: a literature review on the use of machine learning methods *J. Manuf. Syst.* **63** 550–62

[12] Wang G, Li H, Zhang F and Wu Z 2022 Feature fusion based ensemble method for remaining useful life prediction of machinery *Appl. Soft Comput.* **129** 109604

[13] Pillai S and Vadakkepat P 2022 Deep learning for machine health prognostics using Kernel-based feature transformation *J. Intell. Manuf.* **33** 1665–80

[14] Wang G, Zheng D, Yang S and Ma J 2018 FCE-SVM: a new cluster based ensemble method for opinion mining from social media *Inf. Syst. e-Bus. Manage.* **16** 721–42

[15] Li X, Zhang F, Wang G and Fang F 2020 Joint optimization of statistical and deep representation features for bearing fault diagnosis based on random subspace with coupled LASSO *Meas. Sci. Technol.* **32** 025115

[16] Khan S and Yairi T 2018 A review on the application of deep learning in system health management *Mech. Syst. Signal Process.* **107** 241–65

[17] Wang G and Zhang F 2021 A sequence-to-sequence model with attention and monotonicity loss for tool wear monitoring and prediction *IEEE Trans. Instrum. Meas.* **70** 1–11

[18] Mao X, Zhang F, Wang G, Chu Y and Yuan K 2021 Semi-random subspace with Bi-GRU: fusing statistical and deep representation features for bearing fault diagnosis *Measurement* **173** 108603

[19] Xu X, Tao Z, Ming W, An Q and Chen M 2020 Intelligent monitoring and diagnostics using a novel integrated model based on deep learning and multi-sensor feature fusion *Measurement* **165** 108086

[20] Liu H, Liu Z, Jia W and Lin X 2021 Remaining useful life prediction using a novel feature-attention-based end-to-end approach *IEEE Trans. Ind. Inform.* **17** 1197–207

[21] Li X, Zhang W and Ding Q 2019 Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction *Reliab. Eng. Syst. Saf.* **182** 208–18

[22] Ni Q, Ji J C and Feng K 2023 Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network *IEEE Trans. Ind. Inform.* **19** 1301–11

[23] Yang B, Liu R and Zio E 2019 Remaining useful life prediction based on a double-convolutional neural network architecture *IEEE Trans. Ind. Electron.* **66** 9521–30

[24] Xiang S, Qin Y, Luo J, Pu H and Tang B 2021 Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction *Reliab. Eng. Syst. Saf.* **216** 107927

[25] Kong Z, Jin X, Xu Z and Zhang B 2022 Spatio-temporal fusion attention: a novel approach for remaining useful life prediction based on graph neural network *IEEE Trans. Instrum. Meas.* **71** 1–12

[26] Wang M, Li Y, Zhang Y and Jia L 2021 Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines *Aerosp. Syst.* **4** 29–36

[27] Wu Z, Pan S, Chen F, Long G, Zhang C and Philip S Y 2021 A comprehensive survey on graph neural networks *IEEE Trans. Neural Netw. Learn. Syst.* **32** 4–24

[28] Yan R, Xie L, Tang J, Shu X and Tian Q 2020 HiGCIN: hierarchical graph-based cross inference network for group activity recognition *IEEE Trans. Pattern Anal. Mach. Intell.* (https://doi.org/10.1109/TPAMI.2020.3034233)

[29] Yang X, Zheng Y, Zhang Y, Wong D S-H and Yang W 2022 Bearing remaining useful life prediction based on regression shapalet and graph neural network *IEEE Trans. Instrum. Meas.* **71** 1–12

[30] Li T, Zhao Z, Sun C, Yan R and Chen X 2021 Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction *Reliab. Eng. Syst. Saf.* **215** 107878

[31] Saxena A, Goebel K, Simon D and Eklund N 2008 Damage propagation modeling for aircraft engine run-to-failure simulation *2008 Int. Conf. Prognostics and Health Management PHM 2008* (https://doi.org/10.1109/PHM.2008.4711414)

[32] Zhang S, Guo Y, Zhao P, Zheng C and Chen X 2022 A graph-based temporal attention framework for multi-sensor traffic flow forecasting *IEEE Trans. Intell. Transp. Syst.* **23** 7743–58

[33] Yi Y and Lin M 2016 Human action recognition with graph-based multiple-instance learning *Pattern Recognit.* **53** 148–62

[34] Li M, Chen S, Zhao Y, Zhang Y, Wang Y and Tian Q 2020 Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (*Seattle, WA, USA*) pp 211–20

[35] Lin H, Lei Z, Wen G, Tian X, Huang X, Liu J, Zhou H and Chen X 2021 A novel approach of label construction for predicting remaining useful life of machinery *Shock Vib.* **2021** 6806319

[36] Yu W, Kim I Y and Mechefske C 2019 Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme *Mech. Syst. Signal Process.* **129** 764–80

[37] Wang S, Zhang X, Gao D, Chen B, Cheng Y, Yang Y, Yu W, Huang Z and Peng J 2018 A remaining useful life prediction model based on hybrid long-short sequences for engines *2018 21st Int. Conf. on Intelligent Transportation Systems (ITSC)* pp 1757–62

[38] Shi Z and Chehade A 2021 A dual-LSTM framework combining change point detection and remaining useful life prediction *Reliab. Eng. Syst. Saf.* **205** 107257

[39] Listou Ellefsen A, Bjørlykhaug E, Æsøy V, Ushakov S and Zhang H 2019 Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture *Reliab. Eng. Syst. Saf.* **183** 240–51

[40] Li H, Zhao W, Zhang Y and Zio E 2020 Remaining useful life prediction using multi-scale deep convolutional neural network *Appl. Soft Comput. J.* **89** 87–96

[41] Zhang C, Lim P, Qin A K and Tan K C 2017 Multiobjective deep belief networks ensemble for remaining useful life

estimation in prognostics *IEEE Trans. Neural Netw. Learn. Syst.* **28** 2306–18

[42] Li X, Ding Q and Sun J-Q 2018 Remaining useful life estimation in prognostics using deep convolution neural networks *Reliab. Eng. Syst. Saf.* **172** 1–11

[43] Chen Z, Wu M, Zhao R, Guretno F, Yan R and Li X 2021 Machine remaining useful life prediction via an attention-based deep learning approach *IEEE Trans. Ind. Electron.* **68** 2521–31

[44] Fu S, Zhong S, Lin L and Zhao M 2022 A novel time-series memory auto-encoder with sequentially updated reconstructions for remaining useful life prediction *IEEE Trans. Neural Netw. Learn. Syst.* **33** 7114–25