2-1-2024

Remaining useful life prediction using graph convolutional attention networks with temporal convolution-aware nested residual connections

Yupeng Wei
San Jose State University, yupeng.wei@sjsu.edu

Dazhong Wu
University of Central Florida

Janis Terpenny
College of Engineering and Computing

Recommended Citation

Yupeng Wei, Dazhong Wu, and Janis Terpenny. "Remaining useful life prediction using graph convolutional attention networks with temporal convolution-aware nested residual connections" *Reliability Engineering and System Safety* (2024). https://doi.org/10.1016/j.ress.2023.109776

Reliability Engineering and System Safety 242 (2024) 109776

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Remaining useful life prediction using graph convolutional attention networks with temporal convolution-aware nested residual connections

Yupeng Wei [a,*], Dazhong Wu [b], Janis Terpenny [c]

[a] Department of Industrial and Systems Engineering, San Jose State University, San Jose, 95192, CA, USA
[b] Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, 32816, FL, USA
[c] Department of Systems Engineering and Operations Research and Department of Mechanical Engineering, George Mason University, Fairfax, 22030, VA, USA

ARTICLE INFO

Keywords:
Degradation modeling
Residual network
Graph convolutional network
Temporal convolutional network

ABSTRACT

Degradation of engineered systems can result in poor performance and failure. Graph Convolutional Networks (GCNs) have been used to predict the remaining useful life (RUL) of engineered systems by analyzing condition monitoring data. Conventional GCNs typically stack multiple spectral graph convolutional layers, where each layer aggregates condition monitoring data and then projects the aggregated data into another feature space. However, conventional GCNs suffer from two issues. Firstly, repeated aggregation operations affect the temporal correlation of condition monitoring data. Secondly, repeated aggregation and projection operations may generate less significant features, resulting in poor prediction performance. To address these issues, we introduce a temporal convolutional operation to extract and preserve temporal features prior to repeated aggregation and projection operations. Additionally, we create an internal residual connection to skip some aggregation and projection operations to reduce the negative impact of the less significant features. Finally, we use an attention mechanism to extract the most significant features obtained from previous GCN layers and feed them to next GCN layers. We demonstrate the effectiveness of our method through three case studies. Our numerical results show that the proposed approach outperforms existing data-driven methods.

1. Introduction

The degradation of engineered mechanical and electronic systems such as bearings, aircraft engines, and lithium-ion batteries can take place due to wear, corrosion, excessive load, and fatigue [1]. Degradation can result in low reliability, reduced life, and even unexpected failures [2]. For example, the degradation process reduces the total capacity and state of health of batteries of electronic vehicles, leading to a decreased driving range and increased charging frequency [3]. The bearing degradation in aircraft engines may lead to engines' unexpected failures, resulting in delayed or canceled flights [4]. Therefore, developing effective degradation modeling methods is crucial to remaining useful life (RUL) predictions of complex systems.

Over the past few years, deep learning methods have been increasingly used to predict the RUL due to their exceptional prediction accuracy and fidelity [5]. These deep learning methods include neural networks-related approaches, such as autoencoder (AE) [6], convolutional neural network (CNN) [7], and variational autoencoder (VAE) [8]. For example, Yao et al. [9] adopted one-dimensional CNN to extract features from vibration signals to predict the RUL of rotatory bearings, where fully connected layers were replaced by maximum pooling layers for regularization. Wei et al. [10] proposed a dynamic conditional VAE to learn the health indices of aircraft engines, and the learned health indices have been demonstrated as effective to improve the RUL prediction accuracy and fidelity. In addition to the methods mentioned above, neural networks with recurrent characteristics are also widely applied for RUL predictions as these networks with recurrent characteristics can take into account the temporal dynamic dependency of condition monitoring data. The deep learning methods with recurrent attributes involve the recurrent neural network (RNN) [11], long short-term memory (LSTM) [12], gated recurrent unit (GRU) [13], and so on. For example, Shi and Chehade [14] introduced a dual LSTM method to integrate the change point detection process with the RUL prediction, the proposed method used the condition monitoring data after the change point to boost the RUL prediction performance. Cao et al. [15] presented a transferable bidirectional GRU network to predict the RUL of bearings operated under multiple

* Corresponding author.
E-mail address: yupeng.wei@sjsu.edu (Y. Wei).

https://doi.org/10.1016/j.ress.2023.109776
Received 10 March 2023; Received in revised form 19 October 2023; Accepted 31 October 2023
Available online 3 November 2023

working conditions, where both the historical and future condition monitoring data was examined via the bidirectional operation.

Although the effectiveness of existing deep learning methods has been demonstrated in degradation modeling and RUL predictions, they are not effective in revealing the correlation in condition monitoring data at different timestamps [16]. The aforementioned correlation can be used to recognize condition monitoring data with high affinity at different timestamps and aggregate them for improving model robustness and fidelity [17]. In the current literature, such a correlation is typically built by an undirected or directed graph [18], where each graph node represents a vector of condition monitoring data at a timestamp and each edge between two nodes indicates the high affinity among condition monitoring data at different timestamps [19]. To take advantage of the topological architecture of a graph, graph convolutional networks (GCNs) have been increasingly used in recent years. For example, GCNs were used to interpret and predict the mechanical and chemical behaviors of polycrystalline materials by modeling the topological structure of polycrystalline units [20]. GCNs were also used for degradation diagnostics of bearings via affinity graph construction and the aggregation of high similarity frequency-domain features extracted from vibration sensor signals at different timestamps [18].

GCNs usually stack several identical spectral graph convolutional layers for degradation modeling and RUL predictions, where each graph convolutional layer performs two operations. The first operation is to aggregate condition monitoring data with high affinity at different timestamps based on pre-constructed affinity graphs; the second operation is to project the aggregated data into a higher or lower dimensional feature space to obtain features with different dimensionality [21]. Alternatively, GCNs can be interpreted as repeated aggregation and projection operations. While such repeated aggregation and projection operations have been demonstrated to be effective in degradation modeling and RUL predictions, there still exist two issues. First, repeated aggregations of condition monitoring data at different timestamps can severely destroy the temporal correlation of condition monitoring data. Second, similar to many other deep learning methods, the repeated aggregation and projection operations in GCNs can generate less significant features and reduce feature correlation [22]. Feeding these less significant features to next spectral convolutional layers can result in poor prediction performance. To address the first issue, we introduce a temporal convolutional operation to extract temporal-correlated features prior to the repeated aggregation and projection operations in GCNs. To preserve these extracted temporal features, a residual connection is created to provide another shortcut for features to reach the latter portion of a neural network by skipping a few portions. We employ an external residual connection to allow these extracted temporal features to skip all aggregation and projection operations and reach to the end of the neural network so that the temporal correlation in condition monitoring data can be preserved. To address the aforementioned second issue, two methods are introduced to reduce the negative impact of less significant features: (1) an internal residual connection is created among multiple GCN layers to skip a few aggregation and projection operations so that less significant features can be removed; (2) attention mechanisms are used after some GCN layers to extract the most significant features obtained from previous GCN layers and feed them to next GCN layers, where the attention mechanism introduces an attention matrix that identifies the most significant part of features. The contributions of this work are as follows:

- A temporal convolutional operation is introduced to extract temporal features prior to the repeated aggregation and projection operations in GCN layers, and an external residual connection is created to preserve these temporal features.
- Internal residual connections are created among multiple GCN layers to skip a few aggregation and projection operations to reduce the impact of less significant features on prediction accuracy.
- Attention mechanisms are used after some GCN layers to extract the most significant features obtained from previous GCN layers and then feed them to next GCN layers.

The rest of this work is arranged as follows: Section 2 introduces the proposed graph convolutional attention network with temporal convolution-aware nested residual connections. Section 3 illustrates the efficacy and generalizability of the presented graph convolutional attention network with temporal convolution-aware nested residual connections using a publicly available bearing dataset. Section 4 discusses the model performance on an aircraft engine dataset. Section 5 uses a publicly available lithium-ion battery dataset to demonstrate the proposed approach. Section 6 concludes the proposed method and directs future work.

2. Graph convolutional attention network with nested residual connections

In this section, we introduce the graph convolutional attention network with temporal convolution-aware nested residual connections, where the graph convolutional attention network is first introduced in Section 2.1 and the temporal convolution-aware nested residual connections are introduced in Section 2.2.

2.1. Graph convolutional attention network

Fig. 1 shows the flowchart of the graph convolutional attention network (GCAN) including one graph convolutional attention layer. A sliding window with the window size of n is first used to obtain $\mathbf{X}_{i,l}$ which refers to the lth sampled condition monitoring data for unit i. The cosine similarity matrix $\mathbf{S}_{i,l}$ is derived for the sampled condition monitoring data. Next, a masking function $M(\cdot)$ with the connection threshold of ζ is used to construct the affinity graph $\mathbb{G}_{i,l}$, where the edge in the graph refers the high affinity of the sampled condition monitoring data at distinct time-points. Using the constructed affinity graph $\mathbb{G}_{i,l}$, the condition monitoring data at distinct time-points are gathered and projected to derive features $\mathbf{H}_{i,l}$ in another feature space, where the aggregation operation is conducted if there is an edge between two nodes in the affinity graph $\mathbb{G}_{i,l}$. The repeated aggregations and projections generate less significant features, and the self-attention mechanism is used to reduce the negative impact of less significant features. The attention mechanism first generates a query $\mathbf{Q}_{i,l}$ and a key $\mathbf{K}_{i,l}$, and the query and key are utilized to obtain an attention matrix $\mathscr{A}_{i,l}$, and the attention matrix is capable of identifying the most significant and relevant parts of features obtained from previous GCN layers. Last, the attention matrix is used to reduce the negative impact of less significant features in $\mathbf{H}_{i,l}$. More details about the GCAN can be found in the rest paragraphs of Section 2.1.

The first step of the presented graph convolutional attention network is to construct affinity graphs, where the nodes of the affinity graphs represent vectors of condition monitoring data at distinct time-points and edges represent the affinity correlation of these vectors. To construct affinity graphs, a sliding window of using the size of n and the sliding length of 1 is adopted for sampling condition monitoring data. The condition monitoring data captured by the lth sliding window for degradation unit i is represented as $\mathbf{X}_{i,l} \in \mathbb{R}^{n\times p}$, where p represents the amount of channels of condition monitoring data. Here, $x_{i,j,k,l}$ denotes the element of the sampled condition monitoring data. The cosine similarity matrix $\mathbf{S}_{i,l} \in \mathbb{R}^{n\times n}$ for unit i and the lth sampled condition monitoring data is obtained, and the element of $\mathbf{S}_{i,l}$ is expressed as Eq. (1),

$$s_{i,l,j,j'} = \left(\sum_{k=1}^{p} x_{i,j,k,l} \cdot \sum_{k=1}^{p} x_{i,j',k,l}\right) \Big/ \left(\left\|\sum_{k=1}^{p} x_{i,j,k,l}\right\| \cdot \left\|\sum_{k=1}^{p} x_{i,j',k,l}\right\|\right), \quad i = 1,\ldots,m; \quad j = 1,\ldots,n; \quad k = 1,\ldots,p \tag{1}$$

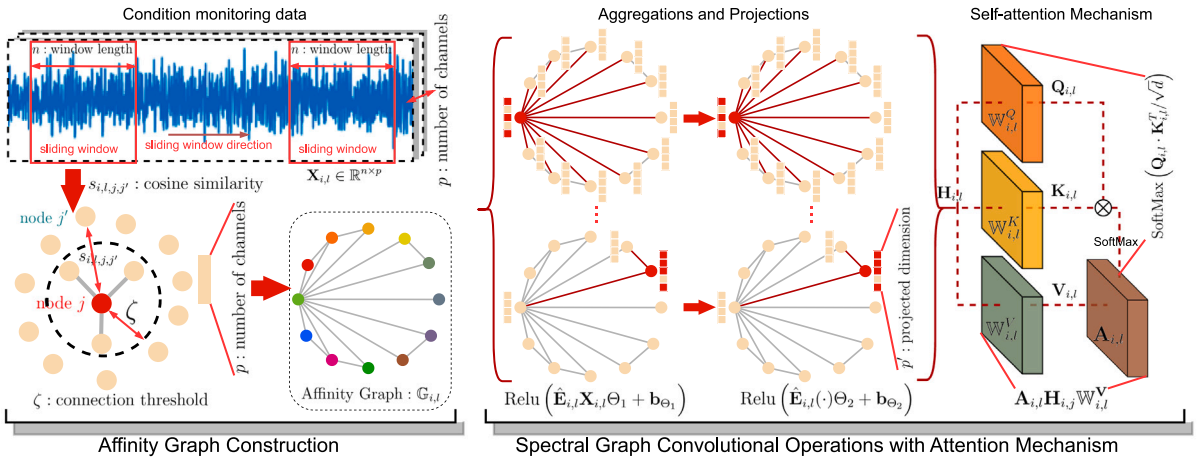


Fig. 1. The flowchart of the presented graph convolutional attention network including one layer.

where $x_{i,j,k,l}$ refers to the condition monitoring data obtained by the l sliding window for degradation unit i and data channel k at time point j; and m is the number of degradation units. Next, a masking function $M(\cdot)$ is implemented on the matrix $\mathbf{S}_{i,l}$ to obtain Dijkstra matrices $\mathbf{E}_{i,l} \in \mathbb{R}^{n\times n}$ for all i and l, which can be written as $\mathbf{E}_{i,l} = M(\mathbf{S}_{i,l})$. The element of the Dijkstra matrices can be obtained by using Eq. (2), where ζ represents the threshold level to establish edges between different vertices.

$$e_{i,l,j,j'} = M(s_{i,l,j,j'}) = \begin{cases} 1 & s_{i,l,j,j'} \geq \zeta, \\ 0 & s_{i,l,j,j'} < \zeta. \end{cases} \tag{2}$$

The affinity graph $\mathbb{G}_{i,l}$ for condition monitoring data sampled by the lth sliding window and unit i can be constructed based on the obtained Dijkstra matrices $\mathbf{E}_{i,l}$, where $e_{i,l,j,j'} = 1$ refers that there exists an edge between node j and node j' and where $e_{i,l,j,j'} = 0$ refers that there does not exist an edge between node j and node j'.

It has been justified that stockpiling over two or three GCN layers may cause a severe decrement in the model's accuracy and fidelity [23]. Therefore, the proposed graph convolutional attention network uses an attention mechanism to reduce the negative impact of less significant features for every two spectral graph convolutional operations, and the first spectral graph convolutional operation can be written as Eq. (3). In this equation, $\mathscr{F}$ is the graph Fourier transform, $\mathscr{F}^{-1}$ is the inverse graph Fourier transform; and $\mathbb{F}_1$ is the graph filter matrix brought by $\mathbb{G}_{i,l}$ during the first spectral graph convolutional operation.

$$\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l} = \mathscr{F}^{-1}\left(\mathscr{F}(\mathbb{F}_1)\odot\mathscr{F}(\mathbf{X}_{i,l})\right) \tag{3}$$

Solving Eq. (3) can be unmanageable. Therefore, we rewrite Eq. (3) as Eq. (4) which is the eigendecomposition of $\mathbb{L}_{i,l}$, here $\mathbf{U}$ refers to the eigenvectors of the Laplacian matrix $\mathbb{L}_{i,l}$; Δ is a diagonal matrix with diagonal entries refer to the eigenvalues of $\mathbb{L}_{i,l}$; and θ_1 refers to the set of parameters provided by the graph filter $\mathbb{F}_1$ during the first spectral graph convolutional operation.

$$\mathbf{U}\left(\mathbf{U}^T\mathbb{F}_1\odot\mathbf{U}^T\mathbf{X}_{i,l}\right) = \mathbf{U}\mathbb{F}_{\theta_1}(\Delta)\mathbf{U}^T\mathbf{X}_{i,l} \tag{4}$$

The Laplacian matrix $\mathbb{L}_{i,l}$ can be written as Eq. (5), and this matrix can be decomposed into eigenvectors and a matrix of eigenvalues. Here, $\mathbf{I} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with all elements of one. $\mathbf{D}_{i,l}$ refers to a matrix with the diagonal element equals to the degree of the Dijkstra matrix $\mathbf{E}_{i,l}$, where the diagonal element $d_{i,l,j,j}$ equals to $\sum_{j'=1}^{n} e_{i,l,j,j'}$.

$$\mathbb{L}_{i,l} = \mathbf{I} - \mathbf{D}_{i,l}^{-1/2}\mathbf{E}_{i,l}\mathbf{D}_{i,l}^{-1/2} = \mathbf{U}\Delta\mathbf{U}^{-1} \tag{5}$$

Because of the Laplacian matrix is symmetric, Eq. (5) is rewritten as Eq. (6).

$$\mathbb{L}_{i,l} = \mathbf{U}\Delta\mathbf{U}^{-1} = \mathbf{U}\Delta\mathbf{U}^T \tag{6}$$

Next, we obtain Eq. (7) by substituting Eq. (6) into Eq. (4).

$$\mathbf{U}\mathbb{F}_{\theta_1}(\Delta)\mathbf{U}^T\mathbf{X}_{i,l} = \mathbb{F}_{\theta_1}(\mathbb{L}_{i,l})\mathbf{X}_{i,l} \tag{7}$$

To simplify the computational procedure, we use the 1st-order Chebyshev polynomials to simplify the spectral graph convolutional operation [24,25]. Thus, Eq. (7) is expressed as Eq. (8) in which $\tilde{\mathbb{L}}_{i,l}$ represents the scaled $\mathbb{L}_{i,l}$; $\mathscr{C}_{1,r}(\tilde{\mathbb{L}}_{i,l})$ represents the Chebyshev polynomial with r orders for the first spectral graph convolution; and $\Theta_1 \in \mathbb{R}^{p\times p'}$ refers to the graph filter matrices for the first spectral graph convolutional operation.

$$\sum_{r=1}^{1}\theta_{1,r}\mathscr{C}_{1,r}(\tilde{\mathbb{L}}_{i,l})\mathbf{X}_{i,l} = \Theta_1\left(\mathbf{D}_{i,l}^{-1/2}\mathbf{E}_{i,l}\mathbf{D}_{i,l}^{-1/2}+\mathbf{I}\right)\mathbf{X}_{i,l} \tag{8}$$

Next, we can obtain Eq. (9) by setting $\tilde{\mathbf{E}}_{i,l} = \mathbf{E}_{i,l} + \mathbf{I}$ and substituting $\tilde{\mathbf{E}}_{i,l}$ into Eq. (8), where $\tilde{\mathbf{D}}_{i,l}$ is the degree matrix of $\tilde{\mathbf{E}}_{i,l}$ and the diagonal entry of this degree matrix is $\sum_{j'=1}^{n}\tilde{e}_{i,l,j,j'}$. Here, $\tilde{e}_{i,l,j,j'}$ is the element of the matrix $\tilde{\mathbf{E}}_{i,l}$.

$$\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l} = \Theta_1\left(\tilde{\mathbf{D}}_{i,l}^{-1/2}\tilde{\mathbf{E}}_{i,l}\tilde{\mathbf{D}}_{i,l}^{-1/2}\right)\mathbf{X}_{i,l} \tag{9}$$

Then, we set $\hat{\mathbf{E}}_{i,l} = \tilde{\mathbf{D}}_{i,l}^{-1/2}\tilde{\mathbf{E}}_{i,l}\tilde{\mathbf{D}}_{i,l}^{-1/2}$, and substitute $\hat{\mathbf{E}}_{i,l}$ into Eq. (9) to derive Eq. (10).

$$\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l} = \hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1 \tag{10}$$

As we use an attention mechanism to reduce the negative impact of less significant features for every two spectral graph convolutional operations, the resulting tensor of the first operation is fed into the second spectral graph convolutional operation, followed by the attention mechanism. Such a process can be mathematically written as Eq. (11), where $\mathbb{F}_2$ is the graph filter matrix brought by $\mathbb{G}_{i,l}$ during the second spectral graph convolutional operation, $\Theta_2 \in \mathbb{R}^{p'\times p}$ refers to the graph filter matrix for the second spectral graph convolution, and $\mathscr{A}_{i,l}[\cdot]$ represents the attention mechanism function used to reduce the negative impact of less significant features for graph $\mathbb{G}_{i,l}$.

$$\mathscr{A}_{i,l}\left[\mathbb{F}_2(\circ_{\mathbb{G}_{i,l}})\left(\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l}\right)\right] = \mathscr{A}_{i,l}\left[\hat{\mathbf{E}}_{i,l}\left(\hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1\right)\Theta_2\right] \tag{11}$$

In order to improve the performance of the proposed graph convolutional attention network, activation functions and bias vectors are incorporated into these spectral graph convolutional operations. Thus, Eq. (11) can be written as Eq. (12), where Relu represents the rectified linear unit activation function; $\mathbf{b}_{\Theta_1}$ refers to the bias vector incorporated in the first spectral graph convolutional operation; and $\mathbf{b}_{\Theta_2}$ is the bias vector included in the second spectral graph convolutional operation.

$$\mathscr{A}_{i,l}\left[\mathbb{F}_2(\circ_{\mathbb{G}_{i,l}})\left(\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l}\right)\right]$$

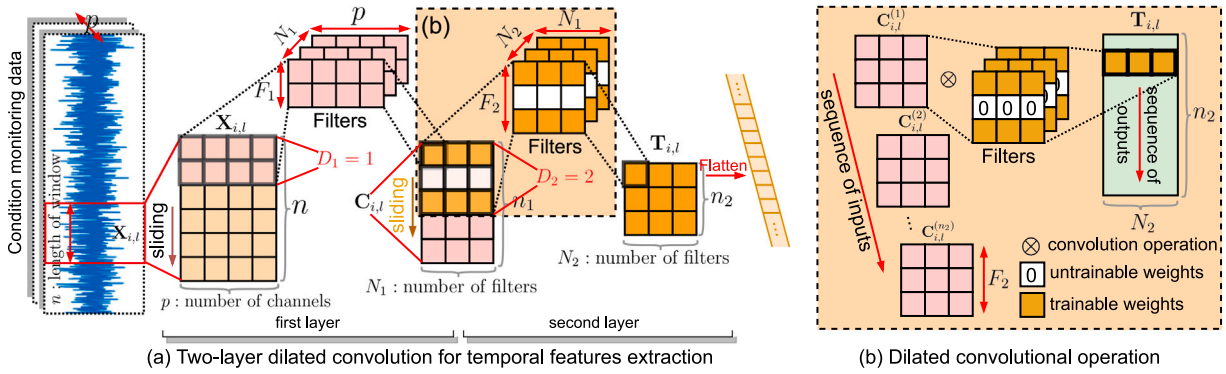


Fig. 2. Temporal convolution network with two dilated convolution layers for extracting temporal features from condition monitoring data.

$$= \mathscr{A}_{i,l}\left[\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\cdot\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1+\mathbf{b}_{\Theta_1}\right)\Theta_2+\mathbf{b}_{\Theta_2}\right)\right] \tag{12}$$

The self-attention mechanism is implemented on the proposed attention function $\mathscr{A}_{i,l}$ for degradation unit i at the lth sliding window. The main intention of the self-attention mechanism is to construct an attention matrix that is capable of identifying the most relevant part of features extracted from previous GCN layers. To derive such an attention matrix, the self-attention mechanism first constructs a query $\mathbf{Q}_{i,l}$ and a pair of key and value $\{\mathbf{K}_{i,l},\mathbf{V}_{i,l}\}$ by using the output $\mathbf{H}_{i,l}\in\mathbb{R}^{n\times p'}$ of the spectral graph convolutional operations for degradation unit i at window l, this output can be written as Eq. (13).

$$\mathbf{H}_{i,l}=\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\cdot\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1+\mathbf{b}_{\Theta_1}\right)\Theta_2+\mathbf{b}_{\Theta_2}\right) \tag{13}$$

Eq. (14) shows the construction process for a query $\mathbf{Q}_{i,l}$ and a pair of key and value $\{\mathbf{K}_{i,l},\mathbf{V}_{i,l}\}$, where $\mathbb{W}^{\mathbf{Q}}_{i,l}\in\mathbb{R}^{p'\times d}$, $\mathbb{W}^{\mathbf{K}}_{i,l}\in\mathbb{R}^{p'\times d}$, and $\mathbb{W}^{\mathbf{V}}_{i,l}\in\mathbb{R}^{p'\times d}$ respectively refers to the weight matrices to obtain the query, key, and value.

$$\left(\mathbf{Q}_{i,l},\mathbf{K}_{i,l},\mathbf{V}_{i,l}\right)=\mathbf{H}_{i,l}\cdot\left(\mathbb{W}^{\mathbf{Q}}_{i,l},\mathbb{W}^{\mathbf{K}}_{i,l},\mathbb{W}^{\mathbf{V}}_{i,l}\right) \tag{14}$$

The attention matrix can be derived by using Eq. (15), where d is the dimension of the matrices to obtain the query, key, and value; and SoftMax represents the softmax activation function.

$$\mathscr{A}_{i,l}=\mathrm{SoftMax}\left(\mathbf{Q}_{i,l}\cdot\mathbf{K}^T_{i,l}/\sqrt{d}\right) \tag{15}$$

Next, the value $\mathbf{Q}_{i,l}$ is multiplied by the attention matrix to reduce the negative impact of less significant features, which can be written as $\mathscr{A}_{i,l}\cdot\mathbf{V}_{i,l}$. In summary, the graph convolutional attention network with one layer is expressed as Eq. (16).

$$\begin{aligned}&\mathscr{A}_{i,l}\left[\mathbb{F}_2(\circ_{\mathbb{G}_{i,l}})\left(\mathbb{F}_1(\circ_{\mathbb{G}_{i,l}})\mathbf{X}_{i,l}\right)\right]\\&=\mathrm{SoftMax}\left(\mathbf{Q}_{i,l}\cdot\mathbf{K}^T_{i,l}/\sqrt{d}\right)\\&\quad\cdot\left[\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\cdot\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1+\mathbf{b}_{\Theta_1}\right)\Theta_2+\mathbf{b}_{\Theta_2}\right)\right]\cdot\mathbb{W}^{\mathbf{V}}_{i,l}\end{aligned} \tag{16}$$

2.2. Temporal convolution-aware nested residual connections

In this section, the temporal convolution-aware nested residual connections are introduced, where the internal residual connection is proposed to skip a few aggregation and projection operations in GCAN layers so that some transmission of less significant features can be skipped; and the external residual connection is introduced to extract and preserve the temporal features of condition monitoring data so that the temporal correlation of condition monitoring data cannot be destroyed. More details about the nested residual connections are introduced in the rest paragraphs of this subsection.

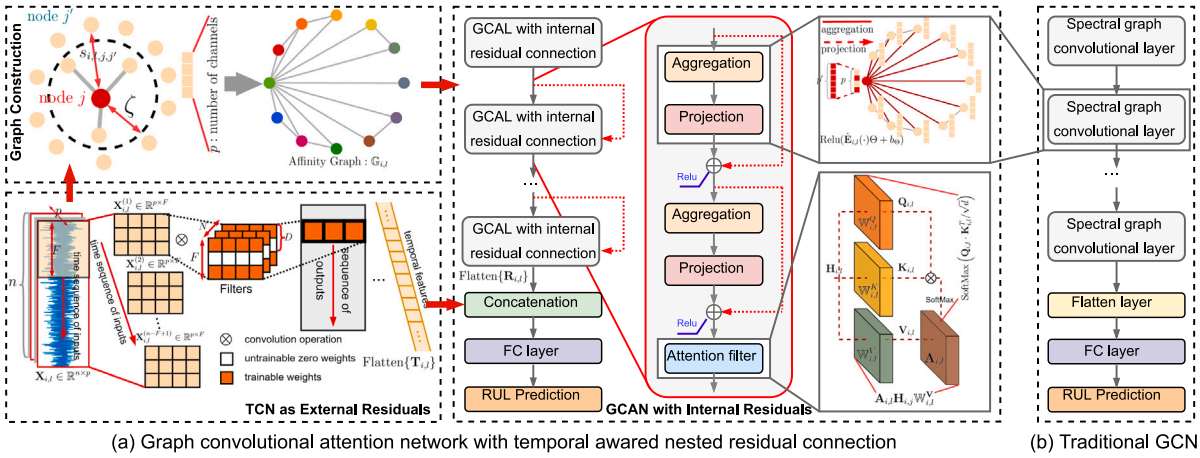
The proposed internal residual connection is inspired by the residual neural network [26], which adds the activated output of the previous spectral graph convolutional layer to the higher-dimensional features projected in the current spectral graph convolutional layer. The output $\mathbf{R}_{i,l}$ of the proposed graph convolutional attention layer (GCAL) with internal residual connections is mathematically expressed as Eq. (17), in which $\mathbf{O}_{i,l}$ is the output of the first spectral graph convolutional layer; $\hat{\mathbf{H}}_{i,l}$ represents the output of the second spectral graph convolution layer with internal residual connection; $\hat{\mathbf{E}}_{i,l}$ represents the scaled $\mathbf{E}_{i,l}$ of the constructed $\mathbb{G}_{i,l}$; Θ_1 and Θ_2 refer to the projection matrix in the first and second spectral graph convolutional operations, respectively; b_{Θ_1} and b_{Θ_2} are bias vectors in the first and second spectral graph convolutional operations; $\mathbf{Q}_{i,l}$ is the generated query; $\mathbf{K}_{i,l}$ is the generated key; and d is the dimension of weighted matrices used to generate the key and query. Based on this equation, we can observe that the output $\mathbf{O}_{i,l}$ of the first spectral graph convolutional layer is not only fed into the second spectral graph convolutional layer, but also added to the projected features in the second spectral graph convolutional layer. Such an internal residual connection can simultaneously guarantee the continuous connection between different layers and skip some of the aggregation and projection operations in some GCN layers so that some transmissions of less significant features generated by the repeated aggregation and projection process can also be skipped.

$$\begin{aligned}\mathbf{R}_{i,l}&=\mathrm{SoftMax}\left(\mathbf{Q}_{i,l}\cdot\mathbf{K}^T_{i,l}/\sqrt{d}\right)\cdot\hat{\mathbf{H}}_{i,l}\cdot\mathbb{W}^{\mathbf{V}}_{i,l}\\\hat{\mathbf{H}}_{i,l}&=\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\mathbf{O}_{i,l}\Theta_2+b_{\Theta_2}+\mathbf{O}_{i,l}\right)\\\mathbf{O}_{i,l}&=\mathrm{Relu}\left(\hat{\mathbf{E}}_{i,l}\mathbf{X}_{i,l}\Theta_1+b_{\Theta_1}\right)\end{aligned} \tag{17}$$

The repeated aggregation and projection operations of the graph convolutional network can destroy the temporal correlation of condition monitoring data. The proposed external residual connection is used to extract and preserve temporal features, representing the temporal correlation of condition monitoring data prior to the repeated aggregation and projection operations in GCN layers. The extracted temporal features $\mathbf{T}_{i,l}$ for degradation unit i at window l are combined with the output $\mathbf{R}_{i,l}$ of the proposed graph convolutional attention layers with internal residual connections for degradation prognostics and RUL predictions. There exists a variety of recurrent neural network architectures that have been introduced to extract the temporal correlation of condition monitoring data, such as LSTM, Convolutional LSTM, GRU, and RNN. However, it has been suggested that the temporal convolutional network (TCN) can achieve better prediction performance in modeling sequential data as TCN can refrain from general drawbacks of recurrent architectures, such as gradient vanishing, gradient explosion, and shortage of memory [27]. TCN primarily uses the causal dilated convolutional operation to extract temporal features from the condition monitoring data. The output of the causal dilated convolution operation $\mathscr{D}$ on the data matrix $\mathbf{X}\in\mathbf{R}^{F\times N}$ can be written as Eq. (18),

$$\left(\mathbf{X}\circledast_D\mathscr{D}\right)=\sum_{f=0}\sum_{b=1}x_{F-f\cdot D,b}\cdot\omega_{F-f\cdot D,b} \tag{18}$$

where $\circledast_D\mathscr{D}$ refers to the dilated convolutional operation with dilation factor D; x is the element of the data matrix $\mathbf{X}$; ω represents the weight



(a) Graph convolutional attention network with temporal awared nested residual connection (b) Traditional GCN

Fig. 3. Flow diagram of the proposed graph convolutional attention network with temporal convolution-aware nested residual connections for degradation prognostics and RUL predictions.

of the filter in $\mathscr{D}$; F refers to the filter size; and N represents to the amount of filters in the previous dilated convolution layer or the amount of channels.

Fig. 2 shows an example of the temporal feature extraction process using the temporal convolution network with two dilated convolution layers. In this figure, the first dilated convolutional layer has a dilation factor $D_1 = 1$, the second dilated convolutional layer has a dilation factor $D_2 = 2$, and the stride length of these two layers is set as one for simplification. In the first layer, the number of filters is N_1 and the dimension of each filter is F_1 by p, and these filters are multiplied by the sampled condition monitoring data $\mathbf{X}_{i,l}$ to obtain $\mathbf{C}_{i,l} \in \mathbb{R}^{N_1 \times n_1}$, where each element of $\mathbf{C}_{i,l}$ can be obtained by using Eq. (17) and n_1 equals $n - F_1 - D_1 + 2$. In the second layer, the number of filters is N_2 and the dimension of each filter is F_2 by N_1, and these filters are multiplied by $\mathbf{C}_{i,l}$ to finalize the extracted temporal features $\mathbf{T}_{i,l} \in \mathbb{R}^{N_2 \times n_2}$, where each element of $\mathbf{T}_{i,l}$ can be obtained by using Eq. (17) and n_2 equals $n_1 - F_2 - D_2 + 2$. Next, the extracted temporal features $\mathbf{T}_{i,l}$ and the output $\mathbf{R}_{i,l}$ from the proposed graph convolutional attention network with internal residual connections are flattened and concatenated as one single vector $\mathbf{v}_{i,l}$. Such a vector is given to a fully connected layer for degradation prognostics.

Fig. 3 shows the flow diagram of the graph convolutional attention network with temporal convolution-aware nested residual connections for RUL prediction, and compares the presented method with the traditional GCN. It can be observed from Fig. 3 that the traditional GCN stacks multiple spectral graph convolutional layers, where each layer perform aggregation and projection operations based on the constructed affinity graph. However, with respect to the proposed method, affinity graphs $\mathbb{G}_{i,l}$ are first initialized for sampled condition monitoring data $\mathbf{X}_{i,l}$ for all unit i and window l. Next, each graph convolutional attention layer (GCAL) uses the constructed graphs to aggregate condition monitoring data at different time points, project the aggregated data into a higher dimensional space, and reduce the negative impact of less significant features generated from the repeated aggregation and projection process. For each graph convolutional attention layer, the internal residual connections are adopted to skip some aggregation and projection operations to skip transmissions of less significant features, and the output of the GCAN with internal residual connections is written as $\mathbf{R}_{i,l}, \forall i, l$. Prior to the aggregation and projection operations in the graph convolutional attention layers, the temporal convolutional operation is used to extract temporal features $\mathbf{T}_{i,l}, \forall i, l$ from the condition monitoring data. Next, the output of the GCAN with internal residual connections is flattened and concatenated with the flattened extracted temporal features, and the concatenation procedure can be written as Eq. (19).

$$\mathbf{v}_{i,l} = \text{Concat}\left(\text{Flatten}\{\mathbf{R}_{i,j}\}, \text{Flatten}\{\mathbf{T}_{i,j}\}\right) \tag{19}$$

Next, the concatenated vector $\mathbf{v}_{i,l}$ is fed into a fully connected (FC) layer to obtain the training loss L, which can be written as Eq. (20). In this equation, $\hat{y}_{i,l}$ and $y_{i,l}$ respectively represent the predicted and ground truth of RUL for degradation unit i at the lth sampled sliding window; t_i is the total life cycle for degradation unit i; and $t_i - n + 1$ represents the number of sampled condition monitoring data with utilizing the sliding window with a size of n for degradation unit i. Next, the gradient descent method is adopted for training the proposed method for learning the parameters.

$$L = \sum_{i=1}^{m} \sum_{l=1}^{t_i - n + 1} (y_{i,l} - \hat{y}_{i,l})^2 \Big/ \sum_{i=1}^{m} t_i - n + 1 \tag{20}$$

3. Case study I: RUL prediction of bearings

3.1. Dataset description

The Prognostics and Health Management 2012 Challenge bearing data was used to evaluate the performance of the proposed method in this case study. This bearing dataset was obtained from a platform that was capable of accelerating the development of deterioration so that bearing failures can be detected within a short period [28]. Fig. 4 exhibits the platform that was adopted to obtain this bearing data, normal and defective bearings generated in this experimental platform. For this dataset, run-to-failure tests were performed to collect vibration signals in two directions for degradation modeling and RUL predictions, and these tests were terminated when the measured vibrations were above 20 g-forces. This dataset was obtained under 3 different conditions. Table 1 shows the operating conditions used in PHM 2012 Challenge bearing dataset and bearing indices associated with different operating conditions. In this work, 7-fold cross-validation was implemented on bearings in the first bearing dataset (seven bearings were included). This cross-validation test aimed at demonstrating the presented method can precisely predict the RUL of bearings operated under a single operating condition. Moreover, 5-fold cross-validation was implemented on bearings in both the second and third bearing datasets (ten bearings were included) to demonstrate that the proposed method is capable of accurately predicting the RUL of bearings operated under multiple operating conditions, where the first fold involves Bearing2_1 and Bearing2_6; the second fold involves Bearing2_2 and Bearing2_7; the third fold involves Bearing2_3 and Bearing3_1; the fourth fold involves Bearing2_4 and Bearing3_2; and the last fold involves Bearing2_5 and Bearing3_3.

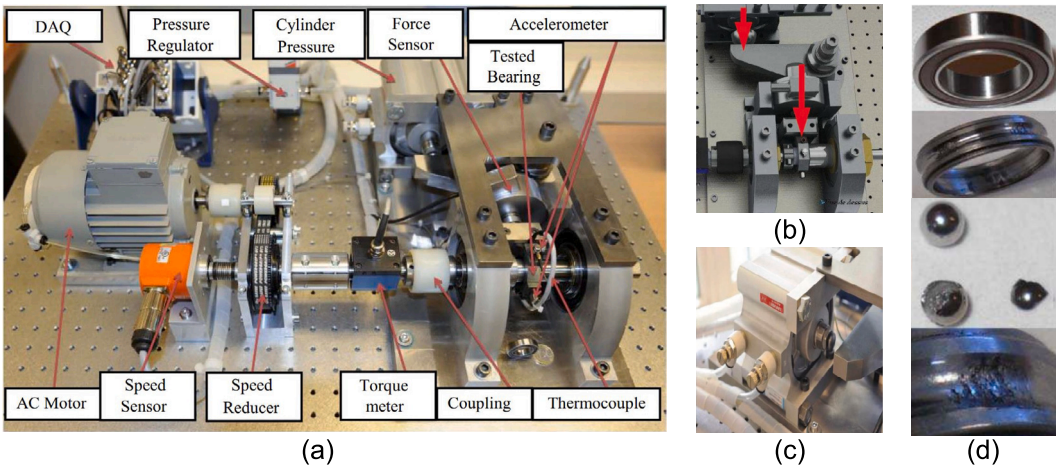


Fig. 4. (a) PRONOSTIA platform was adopted to obtain the PHM 2012 Challenge data; (b) Force transmission; (c) Pneumatic jack; (d) Normal and degraded bearings [28].

Table 1
The operating conditions used in PHM 2012 Challenge bearing dataset and bearing indices associated with different operating conditions.

Condition	Rotation rate (rpm)	Radial pressure (kN)	Applied torque (N m)	Bearing indices
1	1800	4.0	1.326	Bearing1_1 to Bearing1_7
2	1650	4.2	1.447	Bearing2_1 to Bearing2_7
3	1500	5.0	1.591	Bearing3_1 to Bearing3_3

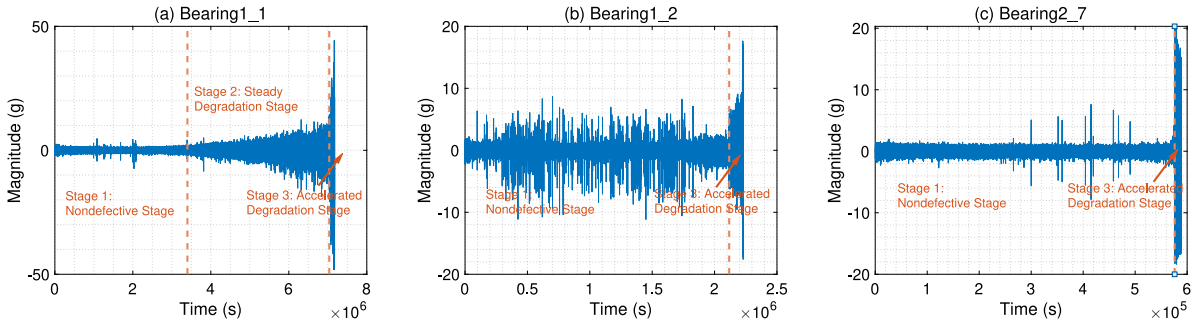


Fig. 5. Collected vibration signals and degradation stage detection results for (a) Bearing1_1, (b) Bearing1_2, and (c) Bearing2_7.

3.2. Degradation stage detection and features extraction

It is widely accepted that the detection of degradation stages could improve the prediction performance of the RUL of rotating bearings [29]. In this case study, we used the most often used abrupt change point detection method to detect different degradation stages, and details of the abrupt change point detection method can be found in [30]. Fig. 5 shows some of the vibration signals and degradation stage detection results. In this figure, we can observe that the number of detected degradation stages varies with respect to different bearing units in this case study. For example, there are three stages are detected with respect to Bearing1_1, where one non-defective stage and two defective stages are incorporated. Bearing1_1 shifts from a non-defective stage (first stage) to a steady degradation stage (second stage) and wraps up in the accelerated degradation stage (third stage). As shown in Fig. 5, only two stages are detected for Bear1_2 and Bearing2_7 in comparison with three stages for Bearing1_1, where these two stages include the non-defective stage and accelerated degradation stage. Due to some bearing units used in this case study do not include the steady degradation stage, we use one predictive model to predict RUL for both the first and second stages and another predictive model to predict RUL for the third degradation stage.

Next, twelve (12) time-domain and eight (8) frequency-domain features are derived from the signals collected in two orientations. These time-domain features consist of maxima, minima, variance, standard deviation, root-mean-square, average, entropy, kurtosis, skewness, peak-to-peak value, standard deviation of inverse sine, and standard deviation of inverse tangent. These frequency-domain features consist of average frequency, median frequency, band power, occupied bandwidth, power bandwidth, maximum power spectral density, maximum amplitude, and frequency of maximum amplitude. Moreover, we use a cumulative sum function to obtain the cumulative version of each extracted feature as it has been demonstrated that the cumulative function can increase the monotonicity of features [31]. The cumulative function is expressed as Eq. (20) in which $x_{i,t,k}$ represents the kth obtained feature for bearing i at time t and $\tilde{x}_{i,t,k}$ is the generated cumulative feature.

$$\tilde{x}_{i,t,k} = \sum_{j=1}^{t} x_{i,j,k} \Big/ \left(\left| \sum_{j=1}^{t} x_{i,j,k} \right| \right)^{1/2} \tag{21}$$

In summary, twenty (20) time-domain and frequency-domain features are extracted from the vibration signals in each direction, and forty (40) features are extracted in total as there is a total of two directions. Moreover, a cumulative function is applied to each extracted feature so that 40 cumulative features are generated from the extracted features. Thus, there are eighty (80) features are extracted in this case study. Next, these extracted features can be used to construct a feature matrix, then a sliding window with the window size of n is used to sample

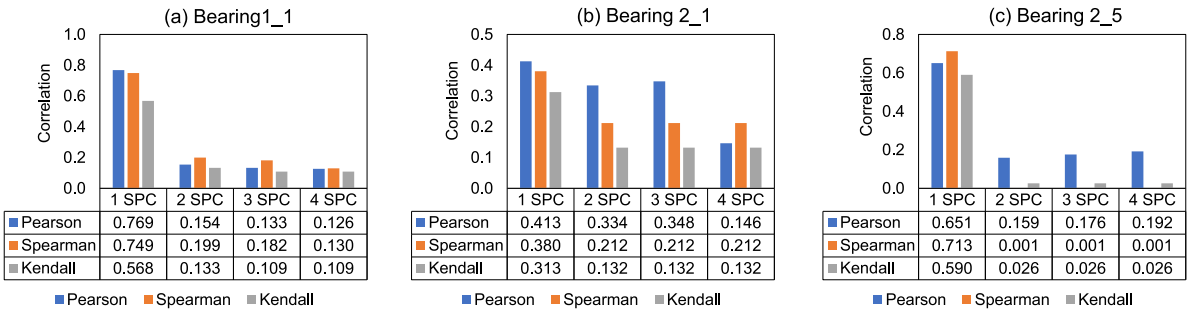


Fig. 6. The Pearson, Spearman, and Kendall correlations between the target RUL and the extracted features with using different number of aggregation and projection operations for the selected three bearing units.

Table 2
The network architecture of the presented graph convolutional attention network with temporal convolution-aware nested residual connections.

Number of stacked layers	Layer description	Connected to	Output dimensionality
1	Input layer	–	$B \times n \times p$
k_1	GCAL with internal residuals	Input layer	$B \times n \times p'$
1	GCN with internal residuals	GCAL with internal residuals	$B \times n \times p'$
1	Flatten layer	GCN with internal residuals	$B \times np'$
k_2	Dilated convolution layer	Input layer	$B \times n_{k_2} \times N_{k_2}$
1	Flatten layer	Dilated convolution layer	$B \times n_{k_2} N_{k_2}$
1	Concatenation layer	Two flatten layers	$B \times (np' + n_{k_2} N_{k_2})$
1	Fully connected layer	Concatenation layer	$B \times 1$

the feature matrix to obtain the feature matrix $\mathbf{X}_{i,l}$ for training and testing. Here, $\mathbf{X}_{i,l}$ refers to the lth sampled condition monitoring data for bearing unit i.

3.3. Features significance after repeated aggregation and projection operations

We conducted an experiment to demonstrate that the repeated aggregation and projection operations in the conventional spectral graph convolutional (SPC) layer can lead to less significant features and reduced feature correlation. Three bearing units (Bearing1_1, Bearing2_1, Bearing2_5) were used in the experiment. We examined the correlations between the target RUL and features extracted using the conventional SPC. Fig. 6 presents the correlations between the target RUL and the extracted features using different numbers of aggregation and projection operations for the selected three bearing units. The correlation metrics used include Pearson correlation, Spearman correlation, and Kendall correlation. In Fig. 6, 1 SPC denotes one aggregation and projection operation in the conventional SPC layer. From the results shown in Fig. 6, it is evident that the feature correlation decreases as the number of aggregation and projection operations increases. For example, in the case of Bearing1_1, the Kendall correlation between the extracted features and RUL is 0.568 after one aggregation and projection operation. However, this correlation decreases to 0.133 when two aggregation and projection operations are performed. Similarly, for Bearing2_5, the Spearman correlation is 0.713 after one aggregation and projection operation, but it drops dramatically to 0.001 after two aggregation and projection operations. The observed decrease in feature correlation emphasizes the need for the proposed method, which aims to mitigate the negative impact of less significant features and enhance prediction performance.

3.4. Hyperparameters

Table 2 exhibits the network architecture of the presented graph convolutional attention network with temporal convolution-aware nested residual connections used in this case study. In Table 2, B represents the batch size, n is the time length of the sampled features, p refers to the amount of features, p' represents the projected dimension using the proposed graph convolutional attention layers, k_1 represents the number of stacked graph convolutional attention layers with internal residuals, k_2 refers to the number of stacked dilated convolution layers, N_{k_2} is the number of filters provided by the k_2-th dilated convolution layer, and n_{k_2} refers to the output dimension after using the dilated convolutional operations.

To optimize the prediction performance, the hyperparameters are set as follows. For all bearing units in the PHM 2012 Challenge bearing dataset under three defective stages, B is determined as 100, n is 20, the number of extracted features p is 80 based upon Section 3.2, p' is set as 100, the number of graph convolutional attention layer with internal residuals k_1 is 3, the number of dilated convolution layer is set as 2, the value of d in the attention mechanism is 100, the number of filters and the filter size for both dilated convolution layers respectively set as 10 and 5, and the dilation factors for two layers is set as $D_1 = 1$ and $D_2 = 2$.

3.5. RUL prediction results

Fig. 7 shows the RUL prediction results for some bearing units, where Fig. 7(a) to Fig. 7(d) shows the predicted RUL versus the true RUL for bearings operated under the first operating condition, and Fig. 7(e) to Fig. 7(i) shows the predicted RUL versus the true RUL for bearings operated under the second and third operating conditions. From Fig. 7, the preliminary conclusion we could draw is that the presented graph convolutional attention network with temporal convolution-aware nested residual connections can forecast the bearings' RUL with high precision as the predicted value is capable of tracing the path of the ground truth of RUL. For instance, with respect to Bearing 1_1, the predicted RUL is 0.994 with the ground truth of RUL is 0.993; and the predicted RUL is 0.620 with the true RUL is 0.668. For Bearing1_6, the RUL prediction is 0.269 with the ground truth is 0.270.

3.6. Ablation and comparative study

To evaluate the efficacy of the temporal convolution-aware nested residual connections and the proposed graph convolutional attention

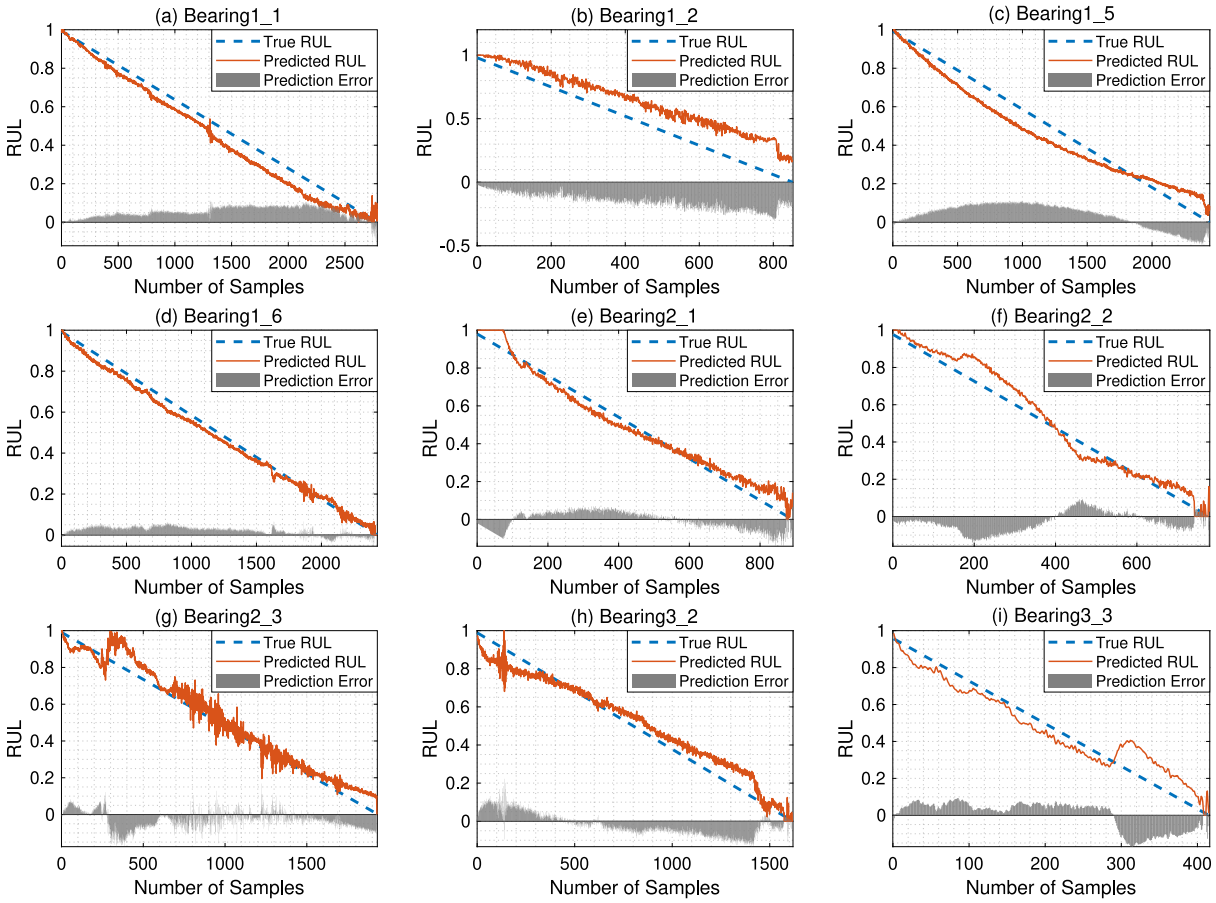


Fig. 7. (a) to (d) shows the predicted RUL versus the true RUL for bearings operated under the first operating condition, and (e) to (i) shows the predicted RUL versus the true RUL for bearings operated under the second and third operating conditions.

Table 3
The abbreviations of the methods used in the ablation study.

Abbreviation	Description
GCAN-TCNR	Graph convolutional attention network with temporal convolution-aware nested residual connections
GCN-TCNR	Graph convolutional network with temporal convolution-aware nested residual connections
GCAN	Graph convolutional attention network without nested residual connections
GCN	Conventional graph convolutional network

network, an ablation study was conducted. Table 3 shows the abbreviations of the methods used in this ablation study. In this table, GCAN-TCNR refers to the presented method, which is the graph convolutional attention network with temporal convolution-aware nested residual connections. GCN-TCNR refers to the graph convolutional network with temporal convolution-aware nested residual connections, which is the ablation study of the self-attention mechanism. GCAN represents the graph convolutional attention network, which is the ablation study of temporal convolution-aware nested residual connections. GCN represents the traditional graph convolutional network, serving as a benchmark method. For GCN-TCNR, the network architecture and hyperparameters setup are identical to GCAN-TCNR (proposed method) except that GCN-TCNR does not include attention mechanism. For GCAN, the network architecture and hyperparameters setup are identical to the proposed method except that GCAN does not have internal and external residual connections. The network architecture and hyperparameters setup of GCN are also the same as the proposed method, but GCN does not include attention mechanism or nested residual connections.

Table 4 shows the RUL prediction RMSE for Bearing1_1 to Bearing3_3 operated under three different conditions in the PHM 2012 Challenge dataset with using the methods listed in this ablation study. From Table 4, we conclude that the proposed temporal convolutional-aware nested residual connections can improve the RUL prediction performance. For instance, regarding Bearing1_1, the prediction RMSE of the proposed GCAN-TCNR is 0.060 in comparison with the RMSE of GCAN is 0.085. Based on this table, we can also observe that the proposed self-attention mechanism can improve the RUL prediction performance. For example, with respect to all bearing units in the PHM 2012 Challenge dataset, the mean RMSE of the GCAN-TCNR is 0.121 compared to the mean RMSE of methods listed in Table 4 ranges from 0.125 to 0.210. In addition, by using the attention mechanism only, the average prediction RMSE reduces by 27.14%. By utilizing the temporal convolution-aware nested residual connections only, the average prediction RMSE reduces by 40.48%. Furthermore, when both the temporal convolution-aware nested residual connections and the attention mechanism are employed, the average prediction RMSE reduces by 42.38%. The proposed nested residual connections enhance prediction performance due to the external residual connection's ability to preserve temporal features prior to repeated feature aggregation in GCN. Additionally, the internal residual connection skips some aggregation operations, reducing the negative impact of less significant features. The self-attention mechanism improves prediction performance by passing the most relevant part of features extracted from the previous GCN layers to the next GCN layers.

Table 4
The RUL prediction RMSE for Bearing1_1 to Bearing3_3 in the PHM 2012 Challenge dataset with using the methods listed in this ablation study.

Operating condition	Bearing index	GCAN-TCNR	GCN-TCNR	GCAN	GCN
Operating Condition 1	Bearing1_1	0.060	0.080	0.085	0.109
	Bearing1_2	0.163	0.179	0.129	0.181
	Bearing1_3	0.074	0.072	0.063	0.082
	Bearing1_4	0.125	0.152	0.139	0.105
	Bearing1_5	0.071	0.075	0.055	0.129
	Bearing1_6	0.029	0.086	0.046	0.124
	Bearing1_7	0.120	0.125	0.112	0.101
Operating Condition 2	Bearing2_1	0.048	0.081	0.033	0.274
	Bearing2_2	0.061	0.144	0.096	0.263
	Bearing2_3	0.060	0.227	0.055	0.284
	Bearing2_4	0.264	0.162	0.235	0.278
	Bearing2_5	0.219	0.338	0.257	0.286
	Bearing2_6	0.103	0.082	0.073	0.274
	Bearing2_7	0.258	0.257	0.268	0.276
Operating Condition 3	Bearing3_1	0.265	0.188	0.262	0.262
	Bearing3_2	0.062	0.127	0.137	0.279
	Bearing3_3	0.077	0.227	0.084	0.265
	Average	**0.121**	0.153	0.125	0.210

Table 5
The average prediction RMSE of the presented GCAN-TCNR, methods used in the ablation study, and other deep learning methods described in the literature.

Condition	GCAN-TCNR	GCN-TCNR	GCAN	GCN	CLSTM [32]	CNN [33]	DAN [34]	GANN [35]	GCNSA [36]
1	**0.092**	0.110	0.090	0.119	0.159	0.189	0.206	0.105	0.087
2	**0.145**	0.145	0.185	0.145	0.277	0.260	0.206	0.187	0.152
3	**0.135**	0.181	0.161	0.269	0.152	0.290	0.366	–	0.206

In addition, the proposed GCAN-TCNR is also compared to other deep learning methods described in the literature, which used the same dataset. Table 5 exhibits the average prediction RMSE of the presented GCAN-TCNR, methods used in the ablation study, and other deep learning methods described in the literature, where these methods incorporate convolutional long short-term memory (CLSTM), convolutional neural network (CNN), deep adversarial network (DAN), generative adversarial neural networks (GANN) and GCN with self-attention mechanism (GCNSA). In this table, it is observed that the presented GCAN-TCNR outperforms many deep learning algorithms described in the literature in most cases. For instance, with respect to bearings in the PHM Challenge dataset operated under the third operating condition, the average prediction RMSE of the proposed GCAN-TCNR is only 0.135. However, the average prediction RMSE of the deep learning algorithms used in the literature fluctuates from 0.152 to 0.366.

4. Case study II: RUL prediction of aircraft engines

4.1. Dataset description

An aircraft engine dataset collected by the commercial modular aero-propulsion system simulation (C-MAPSS) tool was utilized to evaluate the efficacy of the proposed GCAN-TCNR in the RUL prediction of aircraft engines. In this case study, one subset (FD001) of this engine dataset was used, where run-to-failure signals were collected from 100 engine units. More details about this dataset can be found in [37,38]. Fig. 8 shows examples of the collected run-to-failure signals. Moreover, 5-fold cross-validation was adopted to thoroughly illustrate the performance of the proposed GCAN-TCNR on all engine units. These 100 engine units are equally divided into five folds, where the first fold includes engine No. 1 to engine No. 20, the second fold includes engine No. 21 to engine No. 40, the third fold includes engine No. 41 to engine No. 60, the fourth fold includes engine No. 61 to engine No. 80, and the last fold includes engine No. 81 to engine No. 100.

4.2. Features significance after repeated aggregation and projection operations

We conducted an experiment to demonstrate that the repeated aggregation and projection operations in the conventional SPC layer can result in less significant features and reduced feature correlation. Three engine units (Engine No. 1, Engine No. 41, Engine No. 61) were used in the experiment. We examined the correlations between the target RUL and features extracted using the conventional SPC. Fig. 9 presents the correlations between the target RUL and the extracted features, using different numbers of aggregation and projection operations for the selected three engine units. The results shown in this figure clearly indicate that the feature correlation decreases as the number of aggregation and projection operations increases. This decrement in feature correlation can significantly impact prediction performance, emphasizing the necessity for the proposed method, which aims to mitigate the negative impact of less significant features and enhance prediction performance.

4.3. Sensor selections and hyperparameters

In this engine dataset, there are in total 21 sensors have been used to collect run-to-failure data for RUL predictions. However, not all these sensors are relevant to the degradation trend and trajectory of aircraft engines. Therefore, it is critically important to eliminate irrelevant sensors so that the computational cost can be largely reduced and prediction performance can be more robust. One of the most acceptable ways to select relevant sensors is based on their monotonicity and signal-to-noise ratio [10]. In this case study, 14 sensor signals were selected, including T24 (temperature at LPC outlet), T30 (temperature at HPC outlet), T50 (temperature at LPT outlet), P30 (pressure at HPC outlet), Nf (fan speed), Nc (core speed), Ps30 (static pressure at HPC outlet), phi (fuel flow ratio to Ps30), NRf (corrected fan speed), NRc (corrected core speed), BPR (bypass ratio), htBleed (bleed enthalpy), W31 (HPT coolant bleed speed) and W32 (LPT coolant bleed speed). More details of the sensor selection process can be found in [39]. Moreover, all selected sensors are standardized to assure that these sensor signals are in the same scale. Next, these selected sensors are

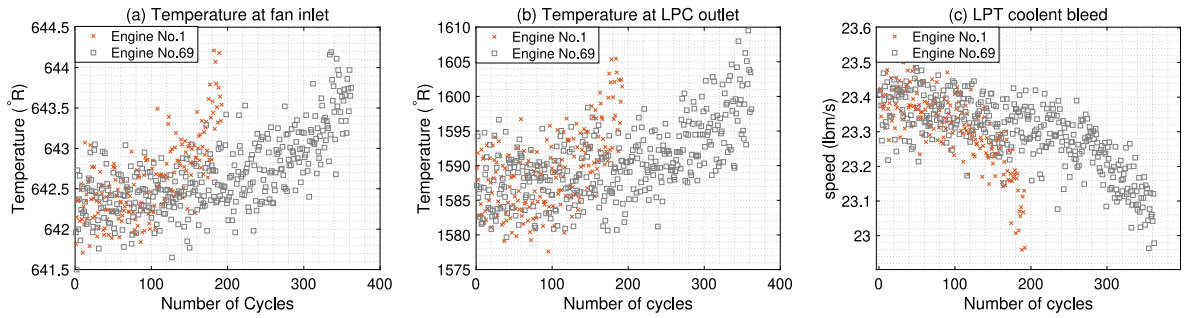


Fig. 8. The collected run-to-failure signals for engine No. 1 and engine No. 69, where (a) shows the trajectory of the total temperature at fan inlet, (b) shows the trajectory of the total temperature at LPC outlet, and (c) shows the trajectory of the LPT coolant bleed speed.

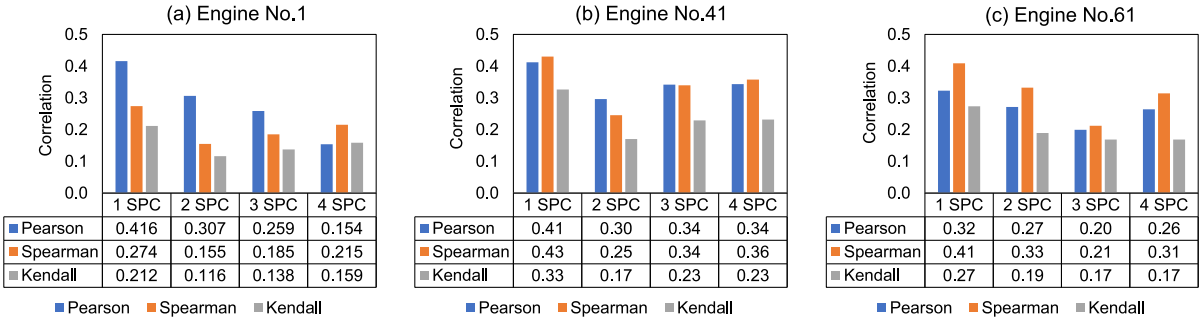


Fig. 9. The Pearson, Spearman, and Kendall correlations between the target RUL and the extracted features with using different number of aggregation and projection operations for the selected three engine units.

used to construct a feature matrix, then a sliding window with the window size of n is used to sample the feature matrix to obtain the feature matrix $\mathbf{X}_{i,l}$ for training and testing. Here, $\mathbf{X}_{i,l}$ refers to the lth sampled condition monitoring data for engine unit i.

The network architecture of the proposed method used in this case study can be found in Table 2. To optimize the prediction performance, the hyperparameters are set as follows. Batch size B is 100, n is 20, the amount of selected sensors p is 14, p' is set as 100, the number of graph convolutional attention layers with internal residuals k_1 is 3, the number of dilated convolution layer is set as 2, the value of d in the attention mechanism is 100, the number of filters and the filter size for both dilated convolution layers is respectively set 10 and 5, and the dilation factors for two layers is set as $D_1 = 1$ and $D_2 = 2$.

4.4. RUL prediction results

Fig. 10 shows the RUL prediction results for some randomly selected engine units. It is evident that the presented GCAN-TCNR is capable of predicting the RUL of aircraft engines with high precision as the predicted value is close to the ground truth of RUL. In addition, we can also observe that the prediction error decreases with the decrement of remaining useful cycles. For example, the true remaining useful life cycle for Engine No. 5 is 9, and the predicted remaining useful life cycle is 10.32. With respect to Engine No. 41, the predicted remaining useful life cycle is 158.47 when the true remaining useful life cycle is 157.

4.5. Ablation and comparative study

To evaluate the efficacy of the presented method, an ablation study was conducted. Fig. 11 depicts the box plot of absolute errors for the methods used in this ablation study across different ranges of true remaining useful cycles, where these methods are listed in Table 3. Based on this figure, it is evident that the proposed GCAN-TCNR has a lower absolute prediction error and less dispersion effect for different ranges of remaining useful cycles. For example, regarding all engine units with true remaining useful cycles ranging from 1 to 10 cycles, the median absolute error of the proposed method is 4 cycles in comparison with other methods' median absolute error ranges from 4.96 cycles to 6.58 cycles, the interquartile range of the proposed method is 4.59 cycles in comparison with other methods' interquartile range fluctuates from 6 cycles to 10.71 cycles. For all engine units with true remaining useful cycles ranging from 21 to 30 cycles, the median absolute error of the proposed method is 6.48 cycles compared to GCN's median absolute error is 16.47, the interquartile range of the proposed method is 8.00 cycles compared to GCAN' interquartile range of 14.50 cycles. Moreover, when the RUL ranges from 1 cycle to 10 cycles, with using the attention mechanism only, the average prediction RMSE reduces by 38.30%; with using the temporal convolution-aware nested residual connections only, the average prediction RMSE reduces by 17.02%; and with using both the temporal convolution-aware nested residual connections and the attention mechanism, the average prediction RMSE reduces by 48.94%.

Table 6 shows the average relative error rate (ARER) of the GCAN-TCNR, GCN-TCNR, GCAN, GCN tabulated in Table 3, and methods described in the literature with respect to different ranges of true RUL. In this table, ANN refers to the artificial neural network (ANN) using a single sensor channel and EDPAE represents an ensemble deep probabilistic autoencoder with using multiple sensor channels. The average relative error rate for engine number i can be expressed as $\text{ARER}_i = \sum_{t=1}^{T_i} \left(y_{i,t} - \hat{y}_{i,t}\right)/T_i$, where $y_{i,t}$ is the ground truth of RUL for engine number i in cycle t, $\hat{y}_{i,t}$ is the predicted value for engine number i in cycle t, and T_i is the total life cycles for engine unit i. From this table, we observe that the presented GCAN-TCNR outperforms other methods in many situations. For instance, when the true RUL ranges from 11 cycles to 20 cycles, the ARER of the presented GCAN-TCNR is 0.031 and the ARER of other methods fluctuates from 0.040 to 0.155. When the true RUL ranges from 1 cycle to 50 cycles, the ARER of GCN fluctuates from 0.047 to 0.154. However, the ARER of the proposed GCAN-TCNR fluctuates from 0.024 to 0.098.

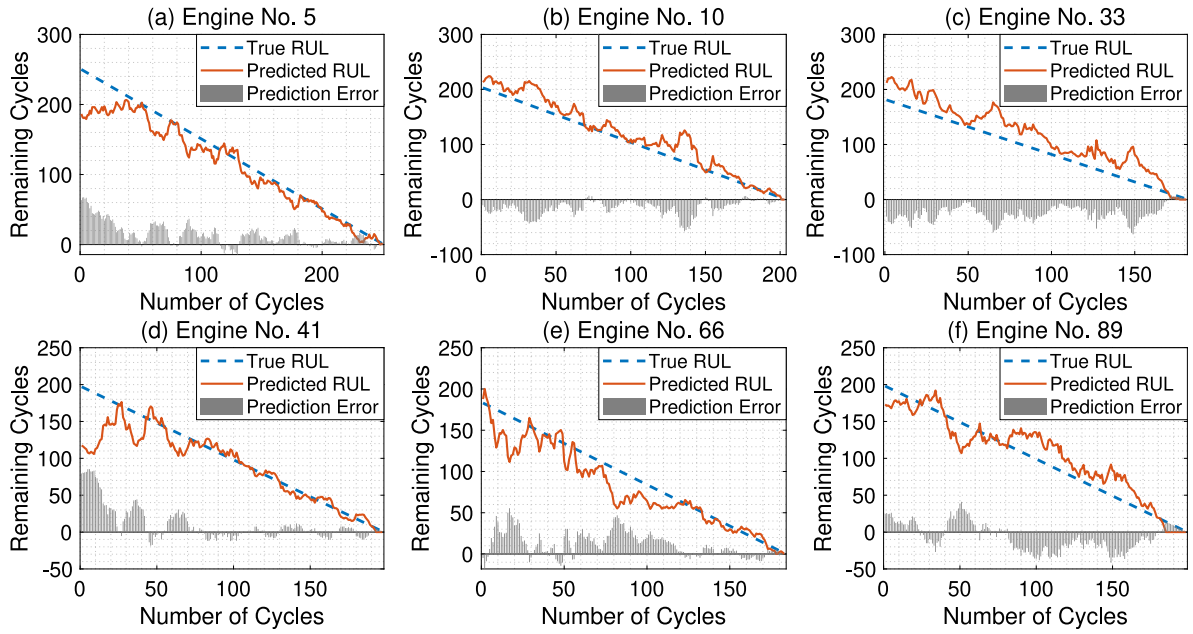


Fig. 10. The RUL prediction results for (a) Engine No. 5, (b) Engine No. 10, (c) Engine No. 33, (d) Engine No. 41 (e) Engine No. 66, and (f) Engine No. 89.

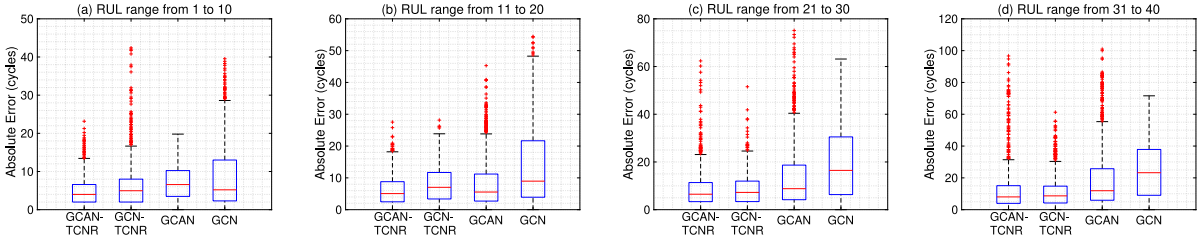


Fig. 11. The box plot of absolute error for methods used in this ablation study for different ranges of true remaining useful cycles.

Table 6
The ARER of the proposed GCAN-TCNR and other deep learning methods used in the literature.

RUL Range (cycles)	GCAN-TCNR	GCN-TCNR	GCAN	GCN	ANN [39]	EDPAE [39]
1–10	0.024	0.029	0.039	0.047	0.069	0.020
11–20	0.031	0.040	0.046	0.077	0.155	0.055
21–30	0.047	0.044	0.075	0.110	0.135	0.066
31–40	0.070	0.061	0.106	0.137	0.197	0.124
41–50	0.098	0.084	0.131	0.154	0.191	0.135

5. Case study III: Battery state of health prediction

5.1. Dataset description

A lithium-ion battery dataset collected by the NASA Ames Prognostics Center of Excellence was adopted to illustrate that the presented GCAN-TCNR can accurately predict the state of health (SOH) of batteries, where four batteries (Battery No. 5, No. 6, No. 7, and No. 18) were utilized in this section. The condition monitoring data was collected during the replicated charging and discharge process, and these condition monitoring data include voltage, current, and temperature. In each charge cycle, a consistent current of 1.5 A was applied unless the voltage measurement rose to 4.2 V, and continued at a consistent voltage unless the current measurement decreased to 20 mA. In each discharge cycle, a consistent current load of 2 A was applied until the measured voltage decreased to certain voltage thresholds. The run-to-failure test was stopped until the capacity dropped to 70%. More details of this experiment can be found in [40]. In practice, the measurements in charge cycles are usually unavailable or hard to obtain [41]. Therefore, we only utilize those measurements obtained in the discharge cycles for SOH predictions in this case study. In this work, 4-fold cross-validation was implemented on these four batteries. Moreover, the SOH prediction starting point is 20 cycles, meaning that the condition monitoring data collected from the previous 20 discharge cycles is used to predict the SOH in the current discharge cycle.

5.2. Features significance after repeated aggregation and projection operations

We conducted an experiment to demonstrate that the repeated aggregation and projection operations in the conventional SPC layer can result in less significant features and reduced feature correlation. Two battery cells (Battery No. 6 and Battery No. 18) were used in the experiment. We examined the correlations between the target SOH and features extracted using the conventional SPC to demonstrate that the feature correlation reduces with the increased numbers of aggregation and projection operations. Fig. 12 presents the correlations between the target SOH and the extracted features, using different numbers of aggregation and projection operations for the selected two battery cells.

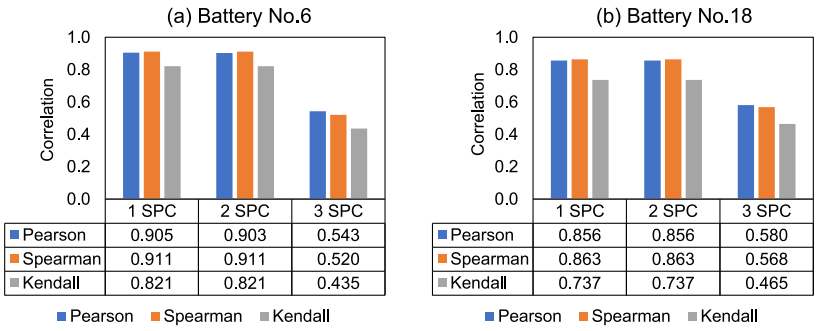


Fig. 12. The Pearson, Spearman, and Kendall correlations between the target SOH and the extracted features with using different number of aggregation and projection operations for the selected two battery cells.

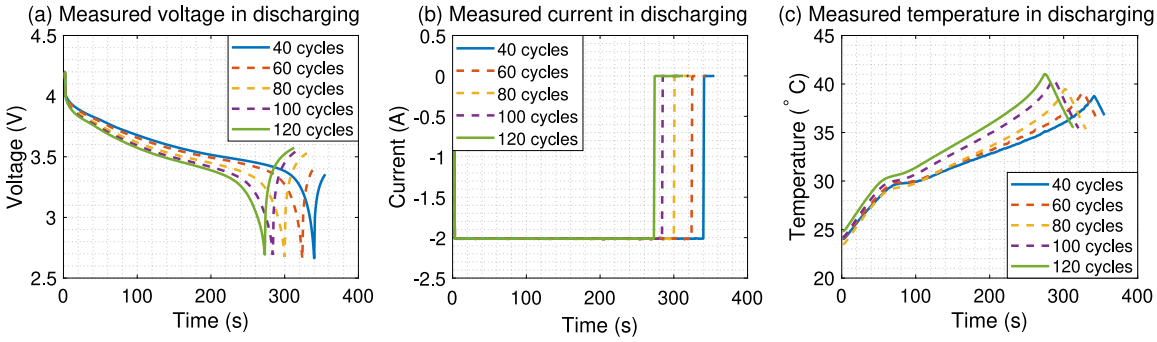


Fig. 13. (a) Measured voltage in discharge cycles, (b) measured current in discharge cycles, and (c) measured temperature in discharge cycles.

The results shown in this figure indicate that the feature correlation decreases with the increment of the number of aggregation and projection operations. This decrement in feature correlation can lead to degraded prediction performance, therefore, the proposed method should be used to reduce the negative impact of less significant features.

5.3. Features extraction and hyperparameters

Fig. 13 shows the trajectory alternation of the measured voltage, current, and temperature with the increment of discharge cycles for Battery No. 6. To better capture the trajectory alternation of these measurements in discharge cycles, temporal features are extracted from the measured voltage, current, and temperature. These extracted temporal features include time to the minimum voltage in discharge cycles, time discharged under a constant current mode, and time to the maximum temperature in discharging [23]. To fully evaluate the capability of the presented method, four-fold cross-validation is implemented on these four batteries.

The network architecture of the presented method used in this case study can also be found in Table 2. To optimize the prediction performance, the hyperparameters are set as follows. Batch size B is determined as 100, n is 20, the amount of extracted features p is 3, p' is set as 100, the number of graph convolutional attention layers with internal residuals k_1 is 3, the number of dilated convolution layer is set as 2, the value of d in the attention mechanism is 100, the number of filters and the filter size for both dilated convolution layers is respectively set 10 and 5, and the dilation factors for two layers is set as $D_1 = 1$ and $D_2 = 2$.

5.4. SOH prediction results

Fig. 14 shows the SOH prediction results for all four batteries. The battery SOH represents its current capacity rated to the maximum capacity. The SOH is expressed as Eq. (21), where $\mathrm{SOH}_{i,t}$ represents the SOH for battery cell i in the discharge cycle t, $y_{i,t}$ is the true capacity for battery cell i in the discharge cycle t, and $\mathbf{C}_i$ represents the maximum capacity for battery cell i.

$$\mathrm{SOH}_{i,t} = y_{i,t}/\mathbf{C}_i \tag{22}$$

In this figure, we can observe that the ground truth of SOH is not monotonically decreasing because of the capacity regeneration phenomenon (CRP) led by the rest time of lithium-ion batteries. Evidently, even though the capacity regeneration phenomenon can significantly impact the degradation trajectories of batteries in this case study, the proposed GCAN-TCNR could still predict the SOH and track the trajectory efficiently. For instance, regarding Battery No. 5, the predicted SOH is 0.883 when the ground truth is 0.902. For Battery No. 18, the predicted SOH is 0.682 when the ground truth is 0.699.

5.5. Ablation and comparative study

Likewise, an ablation study was also conducted to further evaluate the efficacy of the presented GCAN-TCNR in SOH predictions. Table 7 shows the prediction RMSE of the ablation study and compares the presented GCAN-TCNR with other methods reported in the literature in terms of RMSE. In this table, MGP refers to Multiple Gaussian Processes, LRGP represents Logistic Regression with Gaussian Processes, GBT stands for Gradient Boosting Decision Tree, GP refers to Gaussian Process Regression, and OEGCN denotes the optimal entropy graph convolutional network with LSTM. It can be observed from this table that the proposed temporal convolution-aware nested residual connections and the graph convolutional attention network are capable of increasing the prediction performance significantly. For instance, the average RMSE of the presented GCAN-TCNR for these four batteries is 0.0144, however, the average RMSE of GCN-TCNR and GCAN are 0.0160 and 0.0423. With using the attention mechanism only, the average prediction RMSE reduces by 81.78%; with using the temporal convolution-aware nested residual connections only, the average prediction RMSE reduces by 51.82%; and with using both the temporal convolution-aware nested residual connections and the

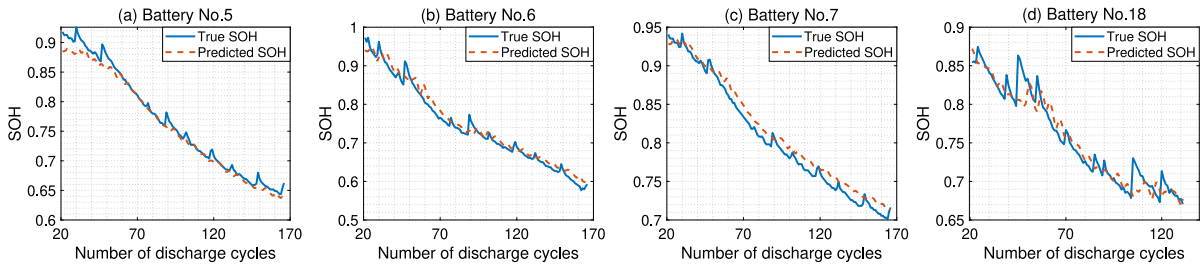


Fig. 14. SOH prediction results for Battery No. 5, No. 6, No. 7, and No. 18.

Table 7
The prediction RMSE of the ablation study and the comparison between the presented GCAN-TCNR and other methods reported in the literature.

Battery	GCAN-TCNR	GCN-TCNR	GCAN	GCN	MGP [42]	LRGP [43]	GBT [44]	GP [43]	OEGCN [23]
No. 5	0.0122	0.0118	0.0202	0.0879	0.0096	0.0168	0.0192	0.0751	0.0167
No. 6	0.0169	0.0202	0.1076	0.1102	0.0167	0.0292	0.0281	0.0406	0.0138
No. 7	0.0115	0.0152	0.0239	0.0895	0.0129	–	0.0157	–	0.0123
No. 18	0.0170	0.0170	0.0174	0.0634	0.0228	0.0169	–	0.0323	0.0201
Average	**0.0144**	0.0160	0.0423	0.0878	0.0155	0.0210	0.0210	0.0493	0.0157

attention mechanism, the average prediction RMSE reduces by 83.60%. Moreover, it is evident that the presented GCAN-TCNR outperforms other methods used in the literature in many situations. For instance, with respect to Battery No. 5, the RMSE of the presented GCAN-TCNR is 0.0122 and the RMSE of the gradient boost decision tree is 0.0192. For Battery No. 6, the RMSE of the presented GCAN-TCNR is 0.0169 and the RMSE of the logic regression with the Gaussian process is 0.0292.

6. Conclusions

In this work, a novel graph convolutional attention network with temporal convolution-aware nested residual connections is developed to predict the RUL of bearings, aircraft engines, and batteries. The proposed method aims at addressing two primary issues in the traditional graph convolutional networks: (1) repeated aggregation operations in GCN layers can severely destroy the temporal correlation in condition monitoring data; (2) repeated aggregation and projection operations may generate less significant features and reduce feature correlation, resulting in poor RUL prediction performance. To deal with the first issue, a temporal convolutional operation is introduced to extract temporal features prior to the repeated aggregation operations in GCN layers, and an external residual connection is created to preserve these extracted features. To deal with the second issue, an internal residual connection is created among multiple GCN layers to skip a few aggregation and projection operations to reduce the negative impact of less significant features; in addition, an attention mechanism is integrated with some GCN layers to reduce the impact of less significant features. Three case studies were conducted to demonstrated the proposed method, where three publicly available datasets were used, including one bearing dataset, one engine dataset, and one lithium-ion battery dataset. The numerical results have shown that the nested residual connections and graph convolution attention network can significantly improve the RUL prediction accuracy in all three case studies. Moreover, the proposed method outperforms other data-driven methods reported in the literature. In the future, we will evaluate the performance of the proposed method on more datasets from different applications.

CRediT authorship contribution statement

Yupeng Wei: Conceptualization, Methodology, Software, Investigation, Validation, Visualization, Writing – original draft. **Dazhong Wu:** Conceptualization, Methodology, Writing – review & editing. **Janis Terpenny:** Conceptualization, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] Li Xiao, An Songyang, Shi Yuanyuan, Huang Yizhe. Remaining useful life estimation of rolling bearing based on SOA-SVM algorithm. Machines 2022;10(9):729.
[2] Wang Zhijian, Ta Yuntian, Cai Wenan, Li Yanfeng. Research on a remaining useful life prediction method for degradation angle identification two-stage degradation process. Mech Syst Signal Process 2023;184:109747.
[3] Han Xuebing, Lu Languang, Zheng Yuejiu, Feng Xuning, Li Zhe, Li Jianqiu, et al. A review on the key issues of the lithium ion battery degradation among the whole life cycle. ETransportation 2019;1:100005.
[4] Wang Cunsong, Zhu Zhenghong, Lu Ningyun, Cheng Yuehua, Jiang Bin. A data-driven degradation prognostic strategy for aero-engine under various operational conditions. Neurocomputing 2021;462:195–207.
[5] Rezaeianjouybari Behnoush, Shang Yi. Deep learning for prognostics and health management: State of the art, challenges, and opportunities. Measurement 2020;163:107929.
[6] Yu Wennian, Kim II Yong, Mechefske Chris. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. Reliab Eng Syst Saf 2020;199:106926.
[7] Chen Zhipeng, Zhu Haiping, Wu Jun, Fan Liangzhi. Health indicator construction for degradation assessment by embedded LSTM–CNN autoencoder and growing self-organized map. Knowl-Based Syst 2022;252:109399.
[8] Huang Yu, Tang Yufei, VanZwieten James. Prognostics with variational autoencoder by generative adversarial learning. IEEE Trans Ind Electron 2021;69(1):856–67.
[9] Yao Dechen, Li Boyang, Liu Hengchang, Yang Jianwei, Jia Limin. Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. Measurement 2021;175:109166.
[10] Wei Yupeng, Wu Dazhong, Terpenny Janis. Learning the health index of complex systems using dynamic conditional variational autoencoders. Reliab Eng Syst Saf 2021;216:108004.
[11] Catelani Marcantonio, Ciani Lorenzo, Fantacci Romano, Patrizi Gabriele, Picano Benedetta. Remaining useful life estimation for prognostics of lithium-ion batteries based on recurrent neural network. IEEE Trans Instrum Meas 2021;70:1–11.
[12] Huang Cheng-Geng, Huang Hong-Zhong, Li Yan-Feng. A bidirectional LSTM prognostics method under multiple operational conditions. IEEE Trans Ind Electron 2019;66(11):8792–802.
[13] Chen Zhen, Xia Tangbin, Li Yanting, Pan Ershun. A hybrid prognostic method based on gated recurrent unit network and an adaptive Wiener process model considering measurement errors. Mech Syst Signal Process 2021;158:107785.

[14] Shi Zunya, Chehade Abdallah. A dual-LSTM framework combining change point detection and remaining useful life prediction. Reliab Eng Syst Saf 2021;205:107257.
[15] Cao Yudong, Jia Minping, Ding Peng, Ding Yifei. Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. Measurement 2021;178:109287.
[16] Wei Yupeng, Wu Dazhong. Remaining useful life prediction of bearings with attention-awared graph convolutional network. Adv Eng Inform 2023;58:102143.
[17] Li Tianfu, Zhao Zhibin, Sun Chuang, Yan Ruqiang, Chen Xuefeng. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction. Reliab Eng Syst Saf 2021;215:107878.
[18] Li Tianfu, Zhao Zhibin, Sun Chuang, Yan Ruqiang, Chen Xuefeng. Multireceptive field graph convolutional networks for machine fault diagnosis. IEEE Trans Ind Electron 2020;68(12):12739–49.
[19] Xie Zongliang, Chen Jinglong, Feng Yong, He Shuilong. Semi-supervised multi-scale attention-aware graph convolution network for intelligent fault diagnosis of machine under extremely-limited labeled samples. J Manuf Syst 2022;64:561–77.
[20] Dai Minyi, Demirel Mehmet F, Liang Yingyu, Hu Jia-Mian. Graph neural networks for an accurate and interpretable prediction of the properties of polycrystalline materials. npj Comput Mater 2021;7(1):1–9.
[21] Wang Lei, Cao Hongrui, Xu Hao, Liu Haichen. A gated graph convolutional network with multi-sensor signals for remaining useful life prediction. Knowl-Based Syst 2022;252:109340.
[22] Shen Yunhang, Ji Rongrong, Wang Yan, Chen Zhiwei, Zheng Feng, Huang Feiyue, et al. Enabling deep residual networks for weakly supervised object detection. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16. Springer; 2020, p. 118–36.
[23] Wei Yupeng, Wu Dazhong. Prediction of state of health and remaining useful life of lithium-ion battery using graph convolutional network with dual attention mechanisms. Reliab Eng Syst Saf 2022;108947.
[24] Wei Yupeng, Wu Dazhong. State of health and remaining useful life prediction of lithium-ion batteries with conditional graph convolutional network. Expert Syst Appl 2023;122041.
[25] Wei Yupeng, Wu Dazhong. Model-based real-time prediction of surface roughness in fused deposition modeling with graph convolutional network-based error correction. J Manuf Syst 2023;71:286–97.
[26] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[27] Lea Colin, Flynn Michael D, Vidal Rene, Reiter Austin, Hager Gregory D. Temporal convolutional networks for action segmentation and detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 156–65.
[28] Nectoux Patrick, Gouriveau Rafael, Medjaher Kamal, Ramasso Emmanuel, Chebel-Morello Brigitte, Zerhouni Noureddine, et al. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In: IEEE international conference on prognostics and health management. IEEE Catalog Number: CPF12PHM-CDR; 2012, p. 1–8.
[29] Shi Junchuan, Yu Tianyu, Goebel Kai, Wu Dazhong. Remaining useful life prediction of bearings using ensemble learning: The impact of diversity in base learners and features. J Comput Inf Sci Eng 2021;21(2).
[30] Lavielle Marc. Using penalized contrasts for the change-point problem. Signal Process 2005;85(8):1501–10.
[31] Javed Kamran, Gouriveau Rafael, Zerhouni Noureddine, Nectoux Patrick. A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling. In: 2013 IEEE conference on prognostics and health management. IEEE; 2013, p. 1–7.
[32] Wan Shaoke, Li Xiaohu, Zhang Yanfei, Liu Shijie, Hong Jun, Wang Dongfeng. Bearing remaining useful life prediction with convolutional long short-term memory fusion networks. Reliab Eng Syst Saf 2022;224:108528.
[33] Li Xiang, Zhang Wei, Ding Qian. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Safety 2019;182:208–18.
[34] Li Xiang, Zhang Wei, Ma Hui, Luo Zhong, Li Xu. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. Knowl-Based Syst 2020;197:105843.
[35] Suh Sungho, Lukowicz Paul, Lee Yong Oh. Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks. Knowl-Based Syst 2022;237:107866.
[36] Wei Yupeng, Wu Dazhong, Terpenny Janis. Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism. Mech Syst Signal Process 2023;188:110010.
[37] Frederick Dean K, DeCastro Jonathan A, Litt Jonathan S. User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS). Technical report, 2007.
[38] Saxena Abhinav, Goebel Kai, Simon Don, Eklund Neil. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International conference on prognostics and health management. IEEE; 2008, p. 1–9.
[39] Wei Yupeng, Wu Dazhong, Terpenny Janis. Constructing robust and reliable health indices and improving the accuracy of remaining useful life prediction. J Nondestruct Eval Diagn Progn Eng Syst 2022;5(2):021009.
[40] Saha Bhaskar, Goebel Kai. Battery data set. 2007, NASA AMES Prognostics Data Repository.
[41] Shi Junchuan, Rivera Alexis, Wu Dazhong. Battery health management using physics-informed machine learning: Online degradation modeling and remaining useful life prediction. Mech Syst Signal Process 2022;179:109347.
[42] Zheng Xueying, Deng Xiaogang. State-of-health prediction for lithium-ion batteries with multiple gaussian process regression model. IEEE Access 2019;7:150383–94.
[43] Yu Jianbo. State of health prediction of lithium-ion batteries: Multiscale logic regression and Gaussian process regression ensemble. Reliab Eng Syst Saf 2018;174:82–95.
[44] Qin Pengliang, Zhao Linhui, Liu Zhiyuan. State of health prediction for lithium-ion battery using a gradient boosting-based data-driven method. J Energy Storage 2022;47:103644.