8-28-2024

# FPCA-SETCN: A Novel Deep Learning Framework for Remaining Useful Life Prediction

Junde Chen

Yuxin Wen

Xuxue Sun

Adnan Zeb

Mohammad Saleh Meiabadi

*See next page for additional authors*

# FPCA-SETCN: A Novel Deep Learning Framework for Remaining Useful Life Prediction




## Authors

Junde Chen, Yuxin Wen, Xuxue Sun, Adnan Zeb, Mohammad Saleh Meiabadi, and Sasan Sattarpanah Karganroudi

# FPCA-SETCN: A Novel Deep Learning Framework for Remaining Useful Life Prediction

Junde Chen, Yuxin Wen, Xuxue Sun, Adnan Zeb, Mohammad Saleh Meiabadi, Sasan Sattarpanah Karganroudi

***Abstract*— The accurate prediction of remaining useful life (RUL) can serve as a reliable foundation for equipment maintenance, thereby effectively reducing the incidence of failure and maintenance costs. In this study, a novel deep learning (DL) framework that incorporates functional principal component analysis (FPCA) and enhanced temporal convolutional network (TCN) is proposed for RUL prediction. Precisely, FPCA is employed to capture the changing patterns in multi-stream degradation trajectories. Subsequently, the reconstructed signals from FPCA are fed into a convolutional block for extracting deep-level features. An enhanced Squeeze-and-Excitation (ESE) block is then incorporated into the network for adaptive feature recalibration, enhancing the network's ability to focus on the most relevant information. The framework includes a TCN module augmented with hybrid attention mechanisms, comprising ESE and spatial attention (SA) blocks, to optimally capture forward and backward sequence information of the feature tensor. The efficiency and feasibility of the proposed approach are demonstrated through case studies on both the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) and Center for Advanced Life Cycle Engineering (CALCE) battery datasets. The proposed method achieves the lowest root mean square error (RMSE) of 15.56 on the C-MAPSS dataset and 0.03 on the CALCE dataset. The comparative studies highlight the superiority of the proposed FPCA-SETCN network over existing deep learning algorithms.**

***Index Terms*— Remaining useful life prediction, Squeeze-and-Excitation block, Temporal convolutional network, Attention mechanism, Functional principal component analysis.**

## I. Introduction

REMAINING useful life (RUL) prediction is an essential component of prognostics and health management (PHM) for mechanical systems. Precise RUL prediction enables the design of maintenance schedules that ensure optimal operational performance of machinery or its components. This helps prevent severe performance degradation, and even mitigating the risk of catastrophic system failures [1]. To attain this objective, a wide range of innovative solutions have been developed, which can be broadly categorized into physical-based and data-driven models [2], [3]. In contrast to physical models, which require a profound understanding of engineering domain knowledge and systematic structures, data-driven models have the capacity to uncover potential relationships within the data without domain knowledge. With the rapid advancement of sensing techniques and computing capacity, massive degradation signals have become available [4]. The data-driven approaches are increasingly gaining popularity for prognostics in various engineering domains.

The data-driven approaches can be loosely categorized into statistical-based models and machine learning models [3], [5]. A typical statistical-based degradation model, such as general path model and stochastic process model, is to identify a suitable mathematical form to model the degradation trajectories over time and then predict RUL [2]. However, most existing statistical-based methods are inadequate for real-world systems, which are often too complex and intrinsically difficult to model, especially for multi-stream degradation signals due to the data dependencies among numerous sensors [6]. Conversely, the basic idea of a machine learning model is to directly construct an end-to-end relationship between the degradation signals and the corresponding RUL of the mechanical system. Over the past decade, shallow machine learning models for RUL prediction have been widely used, such as Support Vector Regression (SVR), Random Forest (RF), Logistic Regression (LR), and Artificial Neural Networks (ANN) [7]. Wang et al. [8] presented a method for predicting the RUL of lithium-ion batteries using improved ant lion optimization and SVR, aiming to enhance the accuracy of RUL prediction, which is crucial for ensuring optimal performance and safety of battery systems. Medjaher et al. [9] performed RUL estimation for critical components of bearings by applying Dynamic Bayesian Networks (DBNs).

*Corresponding author: Yuxin Wen*

J. Chen and Y. Wen are with the Dale E. and Sarah Ann Fowler School of Engineering, Chapman University, Orange, CA 92866 USA (e-mail:jundchen@chapman.edu; yuwen@chapman.edu).

X. Sun is with College of Media Engineering, Communication University of Zhejiang, Hangzhou, Zhejiang 310019 China (email:sunxuxue@zju.edu.cn).

A. Zeb is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518000, Guangdong, China (e-mail: adnanzeb@sustech.edu.cn).

M. Meiabadi is with Department of Mechanical Engineering, École de Technologie Supérieure, 1100 Notre Dame West, Montreal, QC H3C1K3 Canada (e-mail: mohammadsaleh.sheikhmohammadmeiabadi.1@ens.etsmtl.ca).

S. Karganroudi is with Department of Mechanical Engineering, Université du Québec à Trois-Rivières, 3351 Bd des Forges, Trois-Rivières, QC G8Z4M3 Canada (e-mail: Sasan.Sattarpanah.Karganroudi@uqtr.ca).

Nevertheless, these methods are completely dependent upon manual feature extraction and involve a certain degree of subjectivity.

Recently, in view of the increasing volume of data, deep learning-based approaches, especially convolutional neural network (CNN) [10], Long Short-Term Memory (LSTM) [11], Bi-directional Long-Short Term Memory (BiLSTM) [12], and Gated Recurrent Unit (GRU) [13], have gained considerable attention in prognostics due to their accomplishment in handling complex systems [1]. Sateesh et al. [10] proposed a deep CNN model for the prediction of engine gas path faults. In their model, convolution and pooling layers were employed to capture significant patterns of degradation signals at different time scales. Zhu et al. [6] presented a multiscale convolutional neural network (MSCNN) for predicting the RUL of bearings. Instead of using raw vibration signals, they extracted time-frequency characteristics of vibration signals as inputs for MSCNN. Similarly, Wang et al. [14] proposed a multiscale convolutional attention network for predicting the RUL of machinery, embedding two self-attention modules into the networks. Despite their impressive performance, both methods share a common limitation: the local connection structure of CNNs restricts their ability to capture long-term degradation features. Additionally, the incorporation of attention modules in Wang et al. [14]'s method increases the model's parameters, leading to higher computational costs.

Considering the temporality in degradation signals, Park et al. [15] proposed a LSTM model to perform RUL prediction of batteries. Song et al. [16] reported that using a BiLSTM framework to predict the RUL of airplane engines can obtain higher prediction accuracy compared with the SVR, multi-layer perceptron (MLP), CNN, and LSTM methods. Although LSTM and its variants have demonstrated their capability in the application of RUL prediction, they usually require significantly more computations than CNN due to the cascading connection structure. To overcome these issues, some researchers proposed hybrid methods by combining CNN and LSTM sequentially or in parallel, where spatial features and temporal features both can be extracted [17]–[19]. In [17], the authors combined CNN and LSTM in parallel, and then their outputs were concatenated and passed to a fully connected (FC) layer to extract features for lithium-ion battery RUL prediction. Zraibi et al. [18] suggested connecting LSTM layers to the outputs of CNN layers that can improve prediction accuracy with acceptable execution time. In another research, Li et al. [19] trained an enhanced CNN-LSTM RUL prediction model for aircraft engine with convolutional block attention module (CBAM) [20]. While this study reported reasonably good results, the LSTM can only capture one-way sequence dependencies. Moreover, the combined CNN and LSTM architecture has several drawbacks, such as limited processing sequence length and poor parallelism. Additionally, existing attention mechanisms like CBAM are primarily designed for the target detection domain and have high computational complexity.

To better construct the association between CNN and LSTM for sequence modeling problems, temporal convolutional network (TCN), proposed by Bai et al. [21], is the latest improvement. TCN utilizes dilated causal convolution to distill historical information and makes sure no leakage from the future into the past, which is very critical for time-series modeling. It has been proven that TCN has better performance than CNN and LSTM for time series prediction [22]–[25]. Only a few attempts have been made to investigate TCN for RUL prediction. Sun et al. [26] adopted TCN to forecast the RUL of rotating machinery. They first used local mean decomposition to obtain multiscale features, then the extracted features were fed into the TCN. Li et al. [27] utilized a TCN to predict the turbofans' RUL based on the C-MAPSS dataset. Their experimental findings demonstrated that the TCN outperformed three state-of-the-art (SOTA) methods including LSTM, 1DCNN, and deep CNN (DCNN). Sharma et al. [28] trained a TCN model for RUL prediction based on the same dataset. Their results also indicated the superior performance of TCN compared to hybrid architectures like CNN-LSTM and optimized LSTM networks. Despite the promising results reported in the literature, these methods failed to address certain limitations associated with TCN. Specifically, the use of dilated causal convolution in TCN may lead to information loss, where essential features are lost and less significant features are retained during training. This issue poses a potential risk to the accuracy of RUL prediction. Moreover, TCN is susceptible to data noises and may be prone to overfitting. In practical scenarios, signals acquired from operational machines often contain noises and outliers, which can adversely impact the performance of prediction models. Therefore, mitigating these challenges is essential for improving the reliability and robustness of RUL prediction models. To address these concerns, Cao et al. [29] improved the TCN and fused it with a residual self-attention mechanism (TCN-RSA) to extract hidden features. The marginal spectrum was calculated from the vibration signals and treated as the input of their model. Wang et al. [30] introduced a soft threshold TCN with an attention mechanism for machinery prognostics, where a soft thresholding mechanism was used as a flexible activation function and the threshold value was trained by an attention mechanism. Recently, Fu et al. [31] recommended a dual-task TCN with multi-channel attention called MCA-DTCN to perform the prediction of RUL and their proposed method achieved promising performance. In their method, only the channel attention mechanism was considered. In addition, raw signals were directly fed into their model. In practice, transforming degradation signals to reflect intrinsic characteristics, rather than directly feeding raw signals into deep learning models, has great potential to boost prognostic performance [32], [33].

To bridge the aforementioned research gaps, a novel deep learning framework, noted as FPCA-SETCN, is proposed to conduct RUL prediction. First, we consider multi-stream degradation signals as stochastic functions over time, and use functional principal component analysis (FPCA) to distill the changing patterns of multi-stream degradation trajectories. Then, the extracted features are fed into an enhanced TCN framework, noted as SETCN, to further extrapolate the future trajectories and predict the RUL. By employing the FPCA method, the actual degradation process can be reconstructed by the mean trajectory and a set of unit-specific FPC scores. Then

the reconstructed degradation signals are fed into the proposed SETCN model. To prevent losing local information resulting from dilated causal convolution of TCN, an enhanced Squeeze-and-Excitation (ESE) block is integrated into the TCN framework. Subsequently, combining the ESE block with spatial attention (SA), a hybrid attention mechanism is embedded into the network. Following that, the TCN paired with the hybrid attention mechanism is used to obtain sequence information of feature vectors. Concretely, two hybrid attention modules are embedded into the network to highlight the weights of important features while lessening the weights of unwanted features, thereby realizing efficient feature calibration of sequence data.

In brief, the key contributions of this paper can be recapitulated as follows.

- A TCN module is used in the network to obtain the forward and backward sequence information of the features, and a hybrid attention mechanism is integrated into the TCN module for capturing the most relevant features.
- The traditional SE-block is optimized by substituting FC layers with one-dimensional convolutional layers and ReLU activation function. The enhanced SE-block mitigates unnecessary information loss, reduces computational overhead, and it is more sensitive to vibration signals, effectively improving the model accuracy.
- The FPCA method is employed to reconstruct the degradation signals, thereby eliminating irrelevant information and noise. This transformation of degradation signals to reflect intrinsic characteristics, rather than directly inputting raw signals into deep learning models, holds significant potential for enhancing prognostic performance.

The rest of the article is organized as follows. Section II discusses the methodology to perform the task of RUL prediction. Section III focuses on the algorithm experiments, where a series of experiments are conducted and the performance is evaluated and analyzed. Section IV concludes the paper and gives suggestions for further work.

## II. Methodology

In this section, the proposed FPCA-SETCN framework is presented in detail.

### A. Problem Formulation

For the establishment of a prognostic model, the acquisition of run-to-failure data for $N$ training entities is assumed. Let $i$ denote the number of historical units ($i$ = 1, 2, ..., $N$), and $T_i$ represent the total life cycle of the $i$th training unit, the corresponding multi-stream degradation signals can be expressed as $\boldsymbol{D} = \{\boldsymbol{X_i}, i = 1, 2, ..., N\}$, where $\boldsymbol{X_i}$ is the multi-dimensional data of size $T_i \times S$ for unit $i$. The $\boldsymbol{X_i}$ can be written as

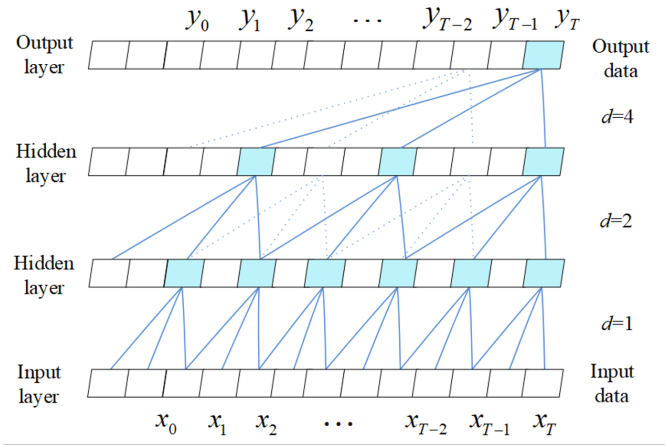
$$\begin{aligned}\boldsymbol{X_i} &= (\boldsymbol{x_{i,1}}, \boldsymbol{x_{i,2}}, ..., \boldsymbol{x_{i,S}}) \\ &= \begin{bmatrix} x_{i,1,1} & x_{i,2,1} & \cdots & x_{i,S,1} \\ x_{i,1,2} & x_{i,2,2} & \cdots & x_{i,S,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1,T_i} & x_{i,2,T_i} & \cdots & x_{i,S,T_i} \end{bmatrix}\end{aligned} \tag{1}$$

In Eq. (1), $x_{i,s} = (x_{i,s,1},\ x_{i,s,2},\ ...,\ x_{i,s,T_i})'$ ($s$ = 1, 2, ..., $S$) is the $s$th features for unit $i$, each element in $x_{i,s}$ denotes one observation, and $S$ indicates the total number of features. This study is dedicated to building a nonlinear prognostic model $f(\cdot)$, which can effectively integrate multi-dimensional data for unit $i$ to accurately predict the RUL, as expressed in

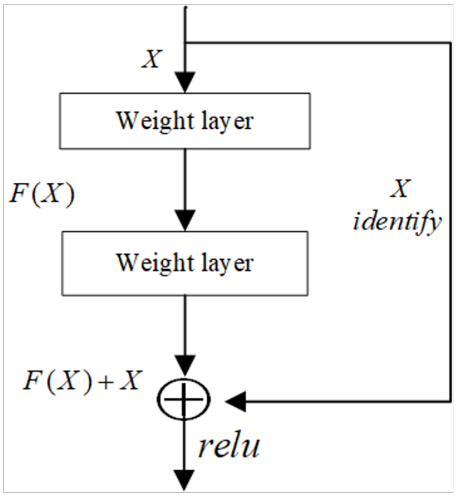
$$h_i(t) = f(x_{i,1,t}, x_{i,2,t}, ..., x_{i,S,t}) \tag{2}$$

### B. Temporal Convolutional Network

We adopt a temporal convolutional network (TCN) as the core component of the proposed model. TCN is a residual network-based CNN designed for handling sequence data [21]. Different from conventional CNNs, TCN consists of three major distinguishing characteristics, including causal convolution, dilated convolution, and residual connections. Among them, causal convolution is a strict time-constrained operation, which is responsible for addressing sequence data and preventing information leakage from future to past. Compared with standard




(a) The dilated causal convolution




(b) The residual block

Fig. 1. The architectural elements in a TCN.

convolution, the dilated convolution has a larger convolution receptive field and it is suitable for handling long-distance dependencies. The fusion of dilated and causal convolutions increases the receptive field and preserves the ordering patterns of sequential input. Besides, residual connections are utilized to address potential issues with gradient disappearance and explosion, which may arise due to the increased depth of the network. The detailed description of these operations is presented as follows.

1) Causal convolution. The role of causal convolution can be abstracted to predict $y_t$ for a given sequence data $X = \{x_0, x_1, ..., x_t\}$ and corresponding filter $F = \{f_0, f_1, ..., f_K\}$. Basically, given $X$, the causal convolution can be expressed by

$$y_t = (F * X)_{(x_t)} = \sum_{k=1}^{K} f_k x_{t-K+k}, \tag{3}$$

where $K$ is the number of filters.

2) Dilated convolution. Given a set of input sequence $X = \{x_0, x_1, ..., x_t\}$ and convolution kernel $f : \{0, ..., k-1\} \rightarrow R$, the dilated convolution can be defined by

$$(F *_d X)_{(x_t)} = \sum_{k=1}^{K} f_k x_{t-(K-k)d} \tag{4}$$

where $d$ represents the expansion coefficient. In practice, as the number of network layers increases, $d$ increases exponentially by 2, e.g., 1, 2, 4 in sequence. When $d = 1$, a dilated convolution reduces to a regular convolution.

3) Residual connection. The residual connection first applies identity mapping to $x$, then fuses the input $x$ into the output $F(x)$ of the model to gain the final TCN output $o$. It can be written as

$$o = \phi(x + F(x)) \tag{5}$$

where $\phi(\cdot)$ is an activation function. The residual connection gives the network a better ability to remember historical information. A dilated causal convolution is a type of causal convolution that applies a filter over a region that exceeds its length, achieved by selectively skipping certain input values with a defined step. Fig. 1 depicts a typical TCN architecture, where Fig. 1(a) and Fig. 1(b) illustrate the dilated casual convolution and the residual block, respectively. Recent research in a wide range of fields has proven that TCN outperforms standard cyclic network architectures such as LSTM and GRUs [21]. Despite the promising performance, the traditional TCN has some weaknesses, e.g., although expanded by the dilated causal convolution, the receptive field of TCN is still limited and it is prone to ignore crucial features.

### C. Enhanced SE-Block

The attention mechanism in deep CNNs can allow models to concentrate on important features while suppressing noise or irrelevant information. With this benefit, much research has been done on attention mechanism, which can be generally divided into channel-wise attention (CA), spatial attention [20], time attention mechanism, and others [34]. Among them, the Squeeze-and-Excitation (SE) network is an influential CA network by explicitly modeling the interdependencies between channels. The core component of the SE network (SE-Net) is the SE block, which is composed of the squeeze, excitation, and reweight operations [35]. Suppose that an intermediate feature map $u \in R^{h \times w \times c}$ is the input in the network, the squeeze operation compresses it by encoding the entire spatial information across channels as a global feature. To produce channel-specific statistics, global average pooling is employed during its execution, defined as

$$z_k = F_{sq}(u_k) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_k(i, j), \tag{6}$$

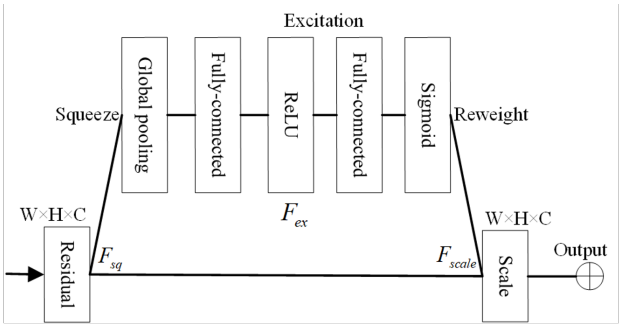
where $z_k$ denotes the $k$-th statistic value, $u_k$ is the $k$-th feature map of the input matrix, $F_{sq}$ implies the squeeze operation, $H$ and $W$ indicate the height and width of $u_k$, respectively. For the global features obtained from the $F_{sq}$, the excitation implements a series of FC, ReLU, FC, and Sigmoid operations to automatically adjust the relation among channels, thereby generating weights for each feature channel. The formula of the excitation operation is written as

$$s = F_{ex}(W, z) = \sigma(f(W, z)) = \sigma(W_2 \vartheta(W_1 z)) \tag{7}$$

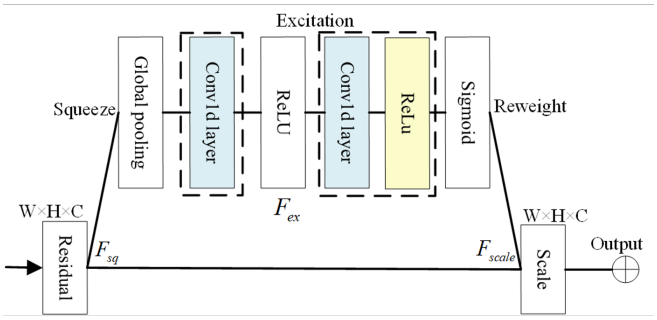
In Eq. (7), $\sigma$ represents the sigmoid activation, $\vartheta$ denotes the ReLU function, $W_1 \in R^{\frac{k}{r} \times K}$, and $W_2 \in R^{K \times \frac{k}{r}}$ (Here $r$ implies a shrinking ratio). It is noteworthy that the weights $W_1$ and $W_2$ are inferred using two non-linear FC layers. At last, the reweight operation uses the activation $s$ to rescale $u_k$ channel-by-channel for performing the recalibration of the original features in channel dimension, expressed as

$$\tilde{x}_k = F_{scale}(s_k, u_k) = s_k * u_k \tag{8}$$

where $[\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_c]$ stands for the final output $\tilde{X}$, and * symbolizes dot products of elements. To our knowledge, the




(a) The traditional SE structure




(b) The enhanced SE structure (ESE block)

Fig. 2. The enhanced squeeze-and-excitation block.

FC layers of the networks contain a considerable number of parameters. Therefore, to compress the model volume and enhance the sensitivity of the SE block for degradation signals, two one-dimensional convolutional layers in the excitation process are utilized to substitute for the existing FC layers. Convolutional layers are more parameter-efficient compared to FC layers as convolutional layers infer local patterns using small filters applied across the input space whereas the FC layers learn global patterns which require more parameters. Moreover, convolutional layers also have a weight-sharing technique that allows multiple sets of features within an input vector to be retrieved by sliding a kernel with the same set of weights on the input vector. Therefore, two one-dimensional convolutional layers are used to replace the existing FC layers of the SE-block since one-dimensional tabular data are processed in this study. In addition, considering the numerous exponential calculations of the sigmoid function ($f(x) = 1/(1+e^{-x})$) and the information loss that may occur when the derivative approaches zero, a ReLU function ($f(x) = max(0,x)$) before the sigmoid layer is added to reduce the amount of calculations and avoid unnecessary information loss. Fig. 2 portrays the structure comparison of the traditional SE block and the enhanced one.

### D. Functional Principal Component Analysis

Degradation signals inevitably are contaminated by noise, especially in high-dimensional data, which may have negative effects on the prognostic model. Extracting the features from original data that reflect the intrinsic trend has great potential to boost prediction performance. In this paper, FPCA is employed for this purpose. FPCA is a statistical technique employed to characterize a set of random trajectories through decomposition into a mean function and principal component functions, which are essentially eigenfunctions of the covariance operator. These eigenfunctions create a compact, orthonormal basis that can be used to represent the actual observed trajectories, and correspond to the fundamental patterns of variation presented within the random trajectories.

In mathematical notation, assume that longitudinal signal $x_i(t), i = 1,2,...,N$ is generated by a square-integrable random process in a given time domain $T$. The sequence $x_i(t)$ can be expressed as $x_i(t) = m_i(t) + \varepsilon_i(t)$, where $m_i(t)$ is a random function with fixed but unknown mean $\mu_i(t)$ and covariance $Cov(x(t),x(t')) = G(t,t')$ for $t,t' \in T$. $\varepsilon_i(t)$ denotes the measurement noise. Based on Mercer's theorem, $G(t,t') = \sum_{k=1}^{\infty} \lambda_{ik}\varphi_k(t)\varphi_k(t')$, where $\varphi_k(t)$ is the associated $k$-th orthogonal eigenfunction and $\lambda_1 > \lambda_2 > \cdots$ are the ordered nonnegative eigenvalues [32]. Therefore, $x_i(t)$ can be broken down into linear combinations of the orthogonal basis function as

$$x_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}\varphi_k(t) + \varepsilon_i(t) \tag{9}$$

In Eq. (9), $\mu(t)$ represents the mean function summarizing the average characteristics of all degradation signals, $\sum_{k=1}^{\infty} \xi_{ik}\varphi_k(t)$ reflects the random effects that distinguish stochastic deviations among different degradation signals, $\xi_{ik}$ are the FPC scores that are mutually uncorrelated with mean zero and variance $var(\xi_{ik}) = \lambda_k$, and $cov(\xi_{ik},\xi_{ik'}) = 0$ for $k \neq k'$. Note that, in most cases, the eigenvalues exhibit rapid decrease and only a few of them are capable of grasping the most crucial information of degradation signals. The rest eigenvalues become negligible. In fact, only considering the first $K$ eigenvalues is sufficient to accurately approximate the signals [33], [36]. As a consequence, $x_i(t)$ can be approximated as

$$x_i(t) = \mu(t) + \sum_{k=1}^{K} \xi_{ik}\varphi_k(t) + \varepsilon_i(t) \tag{10}$$

The value of $K$ can typically be determined using model selection criteria such as a fraction of variance, cross validation (CV), Akaike information criterion (AIC) or others [33], [36].

To solve the FPCA, the mean degradation trend $\mu(t)$, covariance $G(t,t')$, eigenfunctions $\varphi_k(t)$, eigenvalues $\lambda_k$ and FPC scores $\xi_i(k)$ need to be estimated. it can be done by operating directly with the raw data. First, the mean function and raw covariance function can be calculated by $\hat{\mu}(t) = 1/N \sum_{i=1}^{N} x_i(t)$ and $\hat{Cov}(x_i(t),x_i(t')) = (x_i(t)-\hat{\mu}(t))((x_i(t')-\hat{\mu}(t')))$. Note that the diagonal elements of $\hat{Cov}(x_i(t),x_i(t'))$ should be removed because they contain measurement noise. Then the smoothed $\hat{G}(t,t')$ from $\hat{Cov}(x_i(t),x_i(t'))$ can be utilized for estimating the eigenfunction and eigenvalue at any given time $t$ by solving the following expression

$$\int_T \hat{G}(t,t')\varphi_k(t)dt = \hat{\lambda}_k\hat{\varphi}_k(t'), \tag{11}$$

where the $\hat{\varphi}_k$ are subject to $\int_T \hat{\varphi}_k(t)^2 dt = 1$ and $\int_T \hat{\varphi}_k(t)\hat{\varphi}_m(t)dt = 0$ if $m < k$. Then the $k$th FPC score $\xi_i(k)$ can be estimated by

$$\xi_i(k) = \int \varphi_k(t)(x_i(t) - \mu(t))dt, \tag{12}$$

which can be estimated by numerical integration. If the time points vary across units, the raw signals can be first transformed into a basis representation using smooth basis functions, such as B-splines or Fourier basis, then the empirical estimators of the mean and covariance based on the smoothed functional data can be used to conduct FPCA [37].

With the estimated components, the reconstructed signals $m_i(t) = \hat{\mu}(t) + \sum_{k=1}^{K} \hat{\xi}_{ik}\hat{\varphi}_k(t)$ will be used as input for the deep learning model. It is worth noting that the traditional method such as principal component analysis can also be used. However, it encounters some bottlenecks due to data quality issues. For example, in the process of data acquisition, some original data may be missing at some time points. The traditional method is not sufficient to solve these problems. By contrast, FPCA can overcome the defects of traditional methods by detecting the dominant modes of variation in a set of functions using functional data analysis [38].

### E. Proposed FPCA-SETCN Model

The main structure of the proposed FPCA-SETCN, which primarily comprises two modules of feature extraction and RUL prediction, is illustrated in Fig. 3. First, the FPCA method is performed on the original data for reducing noises, as described in Section II $D$. Here, the B-spline basis function is

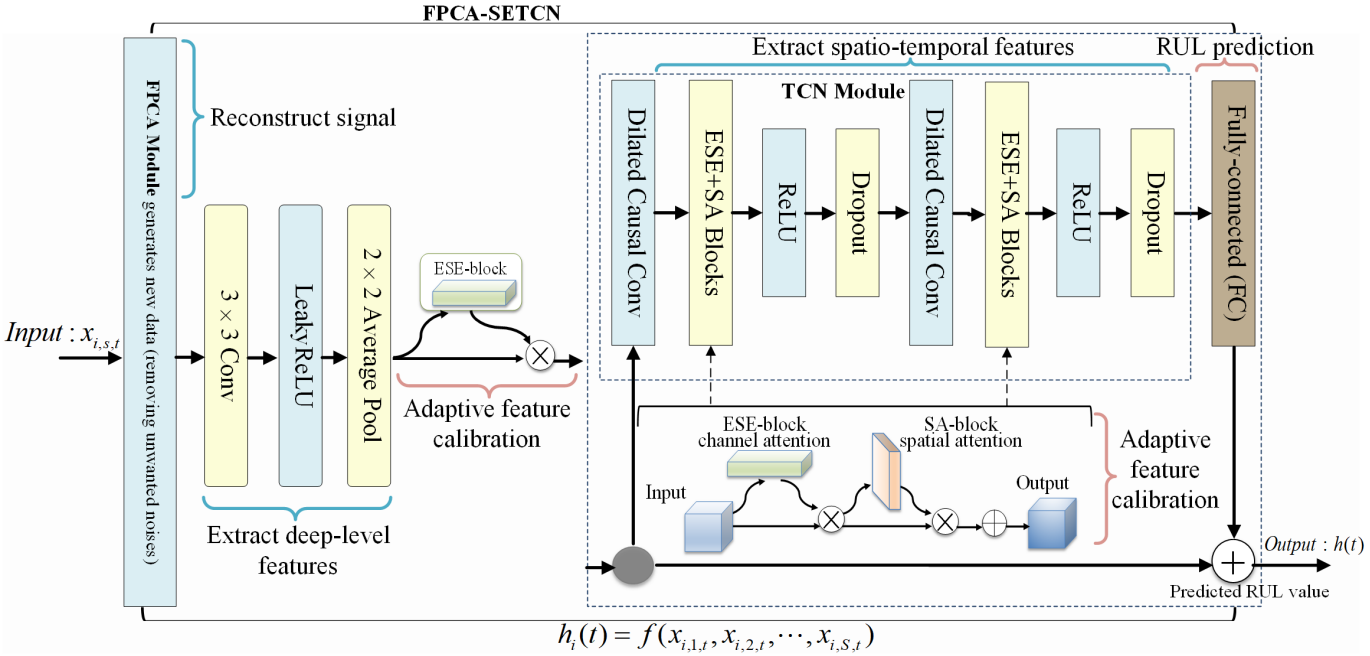



Fig. 3. The architecture of the proposed FPCA-SETCN.

TABLE I
THE PRIMARY PARAMETERS OF THE MODEL

| Layer (Module) | Input shape | No. of filters | Kernel size | Output shape | Param number | Repeated |
|---|---|---|---|---|---|---|
| FPCA | (1, 1, 40, 17) | - | - | (1, 1, 40, 4) | - | 1 |
| Conv2d | (1, 1, 40, 4) | 20 | 3 | (1, 20, 42, 6) | 200 | 1 |
| LeakyReLU | (1, 20, 42, 6) | - | - | (1, 20, 42, 6) | - | 1 |
| MaxPool2d | (1, 20, 42, 6) | - | - | (1, 20, 21, 3) | - | 1 |
| Dropout | (1, 20, 21, 3) | - | - | (1, 20, 21, 3) | - | 1 |
| ESE-Block (AvgPool) | (1, 20, 21, 3) | - | - | (1, 20, 1, 1) | - | 1 |
| Conv1d | (1, 1, 20) | 1 | 2 | (1, 1, 20) | 3 | 2 |
| Sigmoid | (1, 20, 1, 1) | - | - | (1, 20, 1, 1) | - | 1 |
| TCN-Block (Conv2d) | (1, 20, 21, 3) | 20 | 3 | (1, 20, 21, 3) | 3620 | 2 |
| ESE+SA Blocks | (1, 20, 21, 3) | - | - | (1, 20, 21, 3) | - | 2 |
| Conv2d | (1, 20, 21, 3) | 20 | 3 | (1, 20, 21, 3) | 3620 | 2 |
| Add_layer | 2 × (1, 20, 21, 3) | - | - | (1, 20, 21, 3) | - | 1 |
| Flatten_layer | (1, 20, 21, 3) | | | (1, 1260) | - | 1 |
| Linear | (1, 1260) | - | - | (1, 1) | 1260 | 1 |

utilized and the number of basis functions is set to 4 according to cross-validation. Then, the reconstructed signals are fed into a convolutional block for extracting deep-level features, which consists of a 3 × 3 convolutional layer, a Leaky ReLu activation function, an average pooling layer, and a dropout layer. In the model, the convolutional layer is used to extract high-level feature representations and preserve their spatial relationship. Global average pooling is utilized to consolidate global information and simultaneously reduce the number of parameters in the process. Dropout, on the other hand, is used to suppress overfitting risk. The Leaky ReLu activation function is used in this convolutional block and a non-zero slope is assigned to all negative values, namely

$$y_i = \begin{cases} x_i, & if \quad x_i \geq 0 \\ \frac{x_i}{a_i}, & if \quad x_i < 0 \end{cases} \tag{13}$$

In Eq. (13), the fixed parameter $a_i$ is greater than 1, thus a small positive slope in the negative region is applied, which enables it to conduct backpropagation even for negative input values. In the paper, $1/a_i = 0.01$ is adopted. Following the convolutional block, an enhanced SE-block is embedded into the network for adaptive feature calibration. The data dimension does not change when the feature tensor flows through the enhanced SE-block, and then the output is fed

into the TCN module.

TCN cleverly integrates the causal and dilated convolutions, which can extract advanced temporal and spatial features for the RUL prediction task. Besides, the residual connections guarantee a long-term valid usage of long-distance sequence data. Nevertheless, not all features are equally important for message transformation. Some features are crucial for RUL prediction, while others may be negligible. In view of this, the hybrid attention modules comprised of the ESE and SA blocks are incorporated into the TCN module for grasping the optimal features. In the TCN block, two hybrid attention modules are embedded into the network to heighten the weights of the important features while lessening the weights of negligible features. It is worth mentioning that spatial attention is computed by implementing the average-pooling and max-pooling operations. These two are concatenated and convolved using a standard convolutional layer, resulting in the generation of a spatial attention map. The formula of spatial attention can be written by

$$M_s(f') = \sigma(f^{n\times n}([avgpool(f'); maxpool(f')])) \quad (14)$$

where $f'$ indicates the intermediate features and $f^{n\times n}$ represents a convolution operation with a kernel size of $n \times n$. In brief, the output of the whole attention mechanism can be mathematically expressed as

$$O_{c,s}(x) = T_{c,s}(x) * M_{c,s}(x) = T_{c,s}(x) * M_s(M_c(x)) \quad (15)$$

where $c \in \{1, 2, ..., C\}$ is the index of the channel, $x$ indicates the input features, $s$ ranges over all spatial positions, $T_{c,s}$ represents the feature vector extracted through convolution operation, and $M_{c,s}$ refers to the output weight of the whole attention mechanism ($M_c(x)$ denotes the output of channel attention). Note that the order of channel-first then spatial attention is used in our scheme.

After that, normalization of the output feature $O_{c,s}$ is done using a sigmoid activation function, which is applied to all channels and spatial positions without any additional constraints. The formula of the sigmoid activation function is written by

$$\sigma(O_{c,s}) = 1/(1 + e^{(-\frac{O_{c,s}-avg_c}{std_c})}) \quad (16)$$

where $avg$ and $std$ imply the average value and standard deviation of the feature vector, respectively. At last, the output of the TCN is operated by a FC layer, which is employed to implement the RUL prediction task. Mathematically, the FC operation can be described as

$$h^l(x) = \delta(W^l h^{l-1}(x) + b^l) \quad (17)$$

where $l$ indexes the layers, $h$ is the output of the networks, $W$ is the weight matrix, $b$ is a bias vector, and $\delta(\cdot)$ implies the element-wise nonlinear function. Table I summarizes the major parameters of the proposed FPCA-SETCN.

## III. Case Study

### A. Experiment Setup and Evaluation Criteria

To evaluate the efficiency of the proposed method, we conduct the experiments on both the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset from the National Aeronautics and Space Administration (NASA) [7] and the lithium-ion battery dataset from the University of Maryland Center for Advanced Life Cycle Engineering (CALCE) [39]. The C-MAPSS and CALCE datasets are used for the RUL prediction of turbofan engine and lithium-ion battery, respectively. To further verify the validity of the proposed approach, the state-of-the-art RUL prediction methods are chosen for comparison study. All experiments are carried on a computer equipped with AMD Ryzen 9 5900HX processor, 32GB RAM, RTX 3070 GPU.

To investigate the prediction performance of the proposed approach, four evaluation metrics, including the mean absolute error ($MAE$), the root mean square error ($RMSE$), the coefficient of determination R-squared ($R^2$), and the explained

TABLE II
DESCRIPTION OF THE 21 SENSORS AVAILABLE IN THE C-MAPSS DATASET [7].

| ID | Symbols | Description of the characteristic | Units |
|---|---|---|---|
| 1 | T2 | Total temperature at fan inlet | °R |
| 2 | T24 | Total temperature at LPC outlet | °R |
| 3 | T30 | Total temperature at HPC outlet | °R |
| 4 | T50 | Total temperature at LPT outlet | °R |
| 5 | P2 | Pressure at fan inlet | psia |
| 6 | P15 | Total pressure in bypass-duct | psia |
| 7 | P30 | Total pressure at HPC outlet | psia |
| 8 | Nf | Physical fan speed | rpm |
| 9 | Nc | Physical core speed | rpm |
| 10 | Cpr | Engine pressure ratio | - |
| 11 | Ps30 | Static pressure at HPC outlet | psia |
| 12 | Phi | Ratio of fuel flow to static pressure at HPC outlet | Pps/psi |
| 13 | NRf | Corrected fan speed | rpm |
| 14 | NRc | Corrected core speed | rpm |
| 15 | BPR | Bypass ratio | - |
| 16 | farB | Burner fuel-air ratio | - |
| 17 | htBlccd | Bleed enthalpy | - |
| 18 | Nf_dmd | Demanded fan speed | rpm |
| 19 | PCNfR_dmd | Demanded corrected fan speed | rpm |
| 20 | W31 | HPC coolant bleed | Lbm/s |
| 21 | W32 | LPT coolant bleed | Lbm/s |

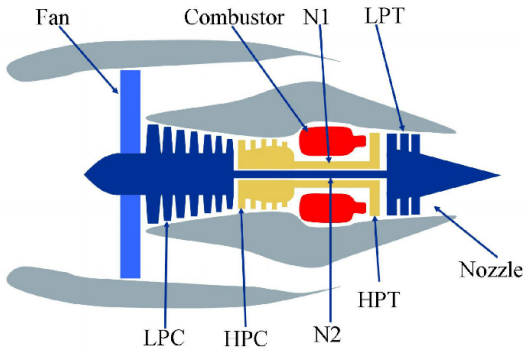



Fig. 4. Engine diagram simulated for the C-MAPSS dataset.

TABLE III
THE RUL PREDICTION ACCURACY ON THE C-MAPSS DATASET.

| Models | Prediction accuracy on training set | | | | Prediction accuracy on test set | | | | Training time |
|---|---|---|---|---|---|---|---|---|---|
| | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | |
| Random Forest | 12.67 | 20.34 | **0.92** | 0.93 | 49.16 | 56.22 | -0.83 | 0.45 | 15s |
| Logistic Regression | 28.82 | 42.02 | 0.63 | 0.64 | 42.07 | 50.72 | -0.48 | 0.24 | 23s |
| LSTM | 28.30 | 44.39 | 0.58 | 0.59 | 27.92 | 34.38 | 0.31 | 0.46 | 95s |
| RF-VAR | 12.57 | 20.25 | 0.91 | 0.91 | 22.76 | 28.41 | 0.53 | 0.62 | 14s |
| LSTM-VAR | 28.41 | 44.28 | 0.58 | 0.60 | 23.15 | 29.30 | 0.50 | 0.57 | 100s |
| This study | **10.02** | **12.45** | **0.92** | **0.95** | **12.05** | **15.56** | **0.79** | **0.79** | 55s |

variance ($E_{VAR}$) are used to facilitate the comparison with benchmark methods. The expressions of these metrics are presented as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{19}$$

$$R^2 = 1 - \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 / \sum_{i=1}^{N}(y_i - \bar{y})^2, \tag{20}$$

$$E_{VAR} = 1 - var(y_i - \hat{y}_i)/var(y_i), \tag{21}$$

where $\hat{y}_i$, $y_i$, and $\bar{y}$ are the predicted value, observed value, and the average of the observed value, respectively. $var(\cdot)$ calculates the empirical variance relative to the test dataset. Among these measure params, $MAE$ assesses the deviation between the observed and predicted values, $RMSE$ depicts the quadratic mean of the differences between the predicted values and actual observations, $R^2$ is a statistical metric to measure the goodness of fit of a model, and $E_{VAR}$ implies the explanatory power of the model. For both $R^2$ and $E_{VAR}$, the best possible value is 1, while greater values are worse for $MAE$ and $RMSE$. To set up the experiment, the learning rate is set to $2 \times 10^{-3}$ based on empirical evidence [21], the Adam (adaptive moment estimation) optimizer is chosen for its adaptive learning rate capabilities and robustness [1], [40], and the model is trained for 100 epochs to ensure stable convergence without overfitting. Further, to verify the validity of the proposed approach, five influential algorithms including Random Forest, Logistic Regression, LSTM, Random Forest with Vector Autoregression (RF-VAR), and LSTM with Vector Autoregression (LSTM-VAR) [41] are selected as the baseline methods to compare models.

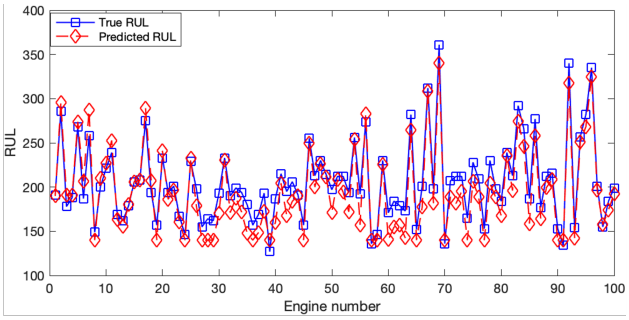
### B. Experiments on the C-MAPSS dataset

*1) Dataset Description:* The C-MAPSS dataset provides a realistic simulation of a commercial turbofan engine with multiple embedded sensors continuously monitor the engine's health status [7]. Fig. 4 presents a simplified diagram of a commercial aircraft gas turbine engine, which typically consists of five modules: high-pressure turbine (HPT), low-pressure turbine (LPT), fan, high-pressure compressor (HPC), and low-pressure compressor (LPC). The dataset is divided into 26 columns, which include the index number of engines, the number of operational cycles, 3 operational settings, and 21 sensor measurements, respectively. Table II presents the description for 21 sensors. A total of four datasets with the corresponding failure modes and operational conditions were generated. In the paper, we only consider the data with a single failure mode and a single operating condition to evaluate the performance. The data contains 100 training units and 100 testing units with 20,631 observation epochs in the training dataset and 13,096 observation epochs in the test dataset. In the training dataset, each unit runs until it fails, whereas in the testing dataset, the operation of each unit terminates at some random time before system failure. Readings from 21 sensors are collected at each observation epoch for each unit. Among the 21 sensors, some sensors have constant outputs with almost no changes throughout the whole life cycle, which cannot provide any useful information for RUL prediction [42]. Therefore, as did in [43] and [42], we remove the outputs of these sensors with indices of 1, 5, 6, 10, 16, 18, and 19 from the C-MAPSS dataset, and the outputs of other sensors with indices of 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21 are selected in this work. Consequently, the outputs of 14 sensors and 3 operational settings, a total of 17 indicators are used as the feature variables and fed into the proposed FPCA-SETCN model for the RUL prediction on the C-MAPSS dataset. Before being fed into the proposed framework, all the data is normalized into the interval of [0,1].

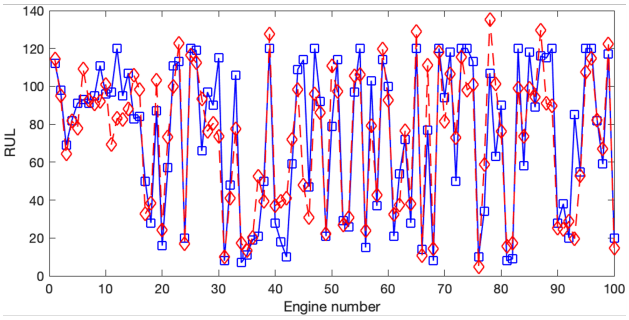
*2) Experimental Results:* Using the proposed method described in Section II $E$, we perform the model training and test on the experimental datasets. Table III presents the prediction performance on the training and test datasets, respectively. For the comparison purpose, the results based on benchmark methods are also provided here. From Table III, it can be observed that the proposed approach achieves the $MAE$, $RMSE$, $R^2$, and $E_{VAR}$ of 10.02, 12.45, 0.92, and 0.95 on the training dataset, and 12.05, 15.56, 0.79, 0.79 on the test dataset, respectively, which are the best performance of all the metrics for the proposed approach compared with those of the benchmark methods. It should be noted that, for the $R^2$ value, the Random Forest algorithm also demonstrates competitive performance with the $R^2$ value of 0.92. Random Forest is an ensemble learning method comprised of multiple decision trees. The computational time for 100 epochs of training is

TABLE IV
COMPARISON WITH RECENT WORK

| ID | References | Year | Description | $RMSE$ |
|---|---|---|---|---|
| 1 | Sateesh et al. [10] | 2016 | Relevance Vector Regression (RVR) | 23.80 |
| 2 | Sateesh et al. [10] | 2016 | CNN | 18.45 |
| 3 | Zhang et al. [42] | 2016 | Extreme learning machine (ELM) | 19.40 |
| 4 | Zhang et al. [42] | 2016 | Deep belief network (DBN) | 18.48 |
| 5 | Zhang et al. [42] | 2016 | Multiobjective deep belief networks ensemble (MODBNE) | 17.96 |
| 6 | Kong et al. [44] | 2019 | CNN_LSTM | 16.13 |
| 7 | Chen et al. [1] | 2021 | Hybrid model | **14.53** |
| 8 | Falcon et al. [45] | 2022 | Deep LSTM | 16.14 |
| 9 | Falcon et al. [45] | 2022 | Neural Turing Machine (NTM) | 15.89 |
| 10 | Costa et al. [46] | 2022 | Variational Autoencoder Recurrent Neural Network (VAE+RNN) | 15.81 |
| 11 | Zhang et al. [47] | 2023 | Multi-task learning-boosted method (MTLTrans) | 16.63 |
| 12 | Azyus et al. [48] | 2023 | CNN-GRU | 16.29 |
| 13 | This study | 2024 | FPCA-SETCN | 15.56 |




(a) RUL prediction on the training dataset




(b) RUL prediction on the test dataset

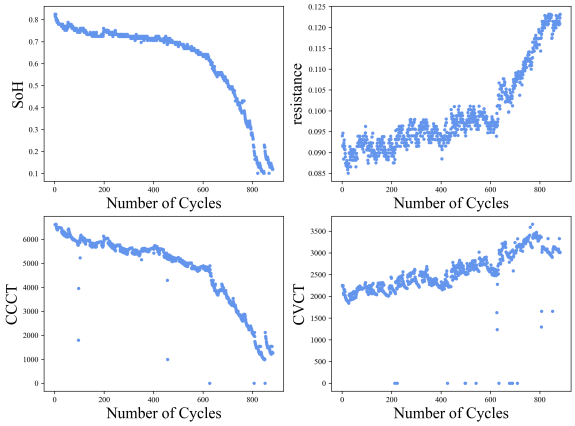
Fig. 5. The RUL prediction results of the proposed method

also reported in Table III, It can be observed that the proposed method is more efficient than LSTM and LSTM-VAR, and slightly worse than LR, RF, and RF-VAR. The training time consumption of the proposed method is 55 seconds, which is close to the average time consumption of the comparison methods. Fig. 5 demonstrates the predicted RUL versus actual RUL for all units using the proposed method both on the training dataset and testing dataset, respectively. The predicted RULs for all units are very close to the actual RULs as shown in Fig. 5. Therefore, based on the experimental findings, it can be concluded that the proposed FPCN-SETCN has presented competitive advantages in both the accuracy and efficiency for the RUL prediction task.

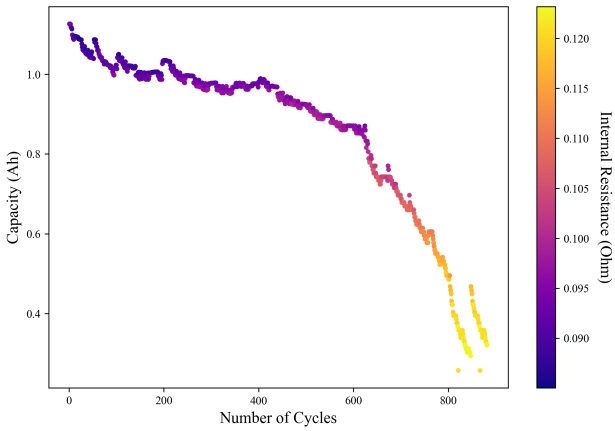
Furthermore, the performance investigation is carried out by comparing our results to those of recent research studies as demonstrated in Table IV. It can be observed that the proposed approach has delivered a competitive performance and outperforms most of the existing methods except the one used in reference [1]. In [1], a hybrid method that combines shallow learning and deep learning was proposed, which contains multiple different modules, such as LSTM, handcrafted features, fully connected layers, and regression modules, which unavoidably makes the model more complex. By contrast, the proposed method is a single model and it achieves a comparable result relative to the hybrid model used in reference [1]. The experiment results suggest that the proposed model effectively captured the underlying patterns in the data and made highly accurate RUL predictions. This highlights the efficacy and superiority of our proposed approach in modeling the complex relationships within the dataset.

### C. Experiments on the CALCE battery dataset

To ensure the scalability and generalizability of the proposed method, we further conducted experiments on the CALCE battery dataset. The CALCE battery dataset contains data from charging and discharging tests of different lithium-ion batteries. The battery cell has a rated capacity of 1.1A/h. In the charging process, the batteries were charged at a constant current (CC) of 0.55A until the battery voltage reached 4.2 V, and then entered a constant voltage (CV) mode until the battery current dropped below 0.05A. The discharge process of these batteries involved a CC of 0.55A until the voltage of the four battery groups dropped to 2.7V. The CS2_35 battery was selected in our experiment. A total of 882 instances are included in this dataset, where 617 samples are for training and 265 samples for test. The capacity of batteries is tested as the target variable, while the other indicators such as the cycle, resistance, state of health (SOH), constant current charge time (CCCT), and constant voltage charge time (CVCT) are used as the input variables. Fig. 6 depicts the metrics distribution and capacity degradation change features of the battery. Table V




(a) Features for input of model.

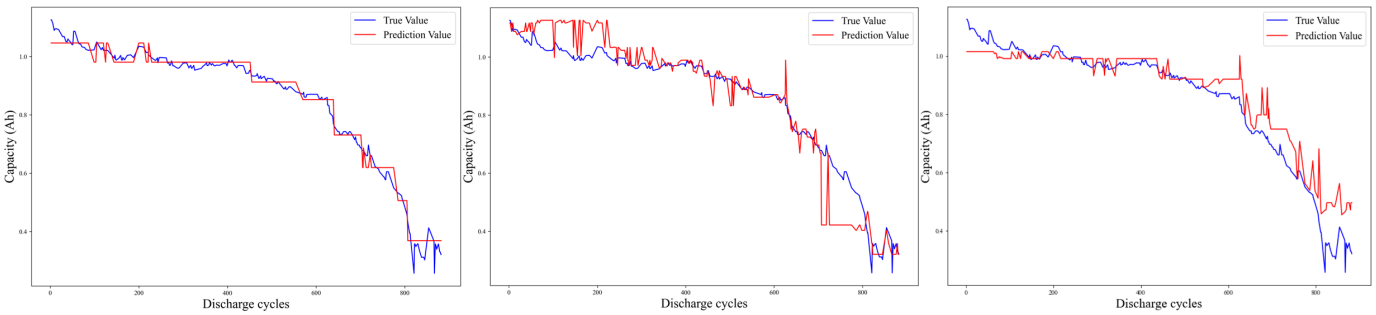



(b) Capacity degradation curve of battery.

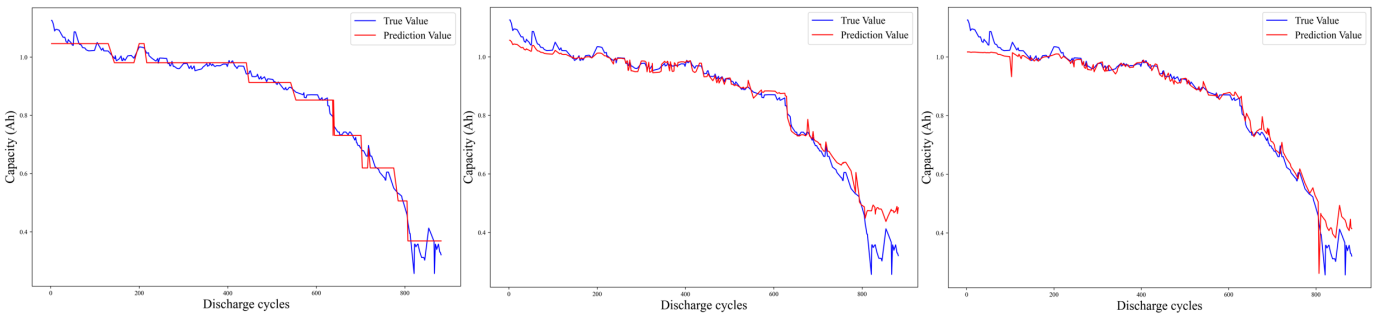
Fig. 6. The characteristics of signals collected from battery.

TABLE V
THE RUL PREDICTION ACCURACY ON THE CALCE BATTERY DATASET

| Models | Prediction accuracy on training set | | | | Prediction accuracy on test set | | | | Trainingtime |
|---|---|---|---|---|---|---|---|---|---|
| | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | |
| Random Forest | **0.02** | **0.03** | **0.98** | **0.98** | **0.02** | **0.03** | **0.98** | **0.98** | 2s |
| Logistic Regression | 0.05 | 0.08 | 0.82 | 0.83 | 0.05 | 0.07 | 0.89 | 0.89 | 3s |
| LSTM | 0.04 | 0.06 | 0.90 | 0.92 | 0.05 | 0.07 | 0.91 | 0.92 | 77s |
| RF-VAR | **0.02** | **0.03** | **0.98** | **0.98** | **0.02** | **0.03** | **0.98** | **0.98** | 3s |
| LSTM-VAR | 0.03 | 0.04 | 0.95 | 0.96 | 0.03 | 0.05 | 0.94 | 0.95 | 113s |
| This study | **0.02** | 0.04 | 0.95 | 0.96 | **0.02** | **0.03** | 0.97 | **0.98** | 106s |




(a) Random Forest (b) Logistic Regression (c) LSTM




(d) RF-VAR (e) LSTM-VAR (f) This study

Fig. 7. The prediction performance of different methods on the CALCE dataset.

TABLE VI
ABLATION EXPERIMENTS ON THE C-MAPSS DATASET

| Ablation approach | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | Time (s) |
|---|---|---|---|---|---|
| Traditional SE-block | 12.73 | 17.60 | 0.73 | 0.74 | 63 |
| Delete ESE-block | 13.97 | 18.46 | 0.77 | 0.78 | 43 |
| Delete ESE+SA | 18.76 | 23.50 | 0.52 | 0.71 | **22** |
| Delete FPCA | 13.57 | 18.38 | 0.76 | 0.76 | 190 |
| This study | **12.05** | **15.56** | **0.79** | **0.79** | 55 |

TABLE VII
ABLATION EXPERIMENTS ON THE CALCE DATASET

| Ablation approach | $MAE$ | $RMSE$ | $R^2$ | $E_{VAR}$ | Time (s) |
|---|---|---|---|---|---|
| Traditional SE-block | 0.03 | 0.04 | 0.96 | 0.96 | 126 |
| Delete ESE-block | 0.03 | 0.05 | 0.95 | 0.95 | 76 |
| Delete ESE+SA | 0.04 | 0.05 | 0.95 | 0.95 | **65** |
| Delete FPCA | 0.05 | 0.05 | 0.94 | 0.96 | 115 |
| This study | **0.02** | **0.03** | **0.97** | **0.98** | 106 |

summarizes the training and test accuracy of different methods on the CALCE battery dataset, and the comparative analysis of test results for different methods are portrayed in Fig. 7.

From Table V it can be observed that the proposed approach has attained a promising performance compared to that of other SOTA methods, even though the optimum regressor algorithms are adopted. On the training set, the proposed approach has achieved the average $MAE$, $RMSE$, $R^2$, and $E_{VAR}$ of 0.02, 0.04, 0.95, and 0.96, where the $MAE$ of 0.02 is the top performance of all the compared methods. On the test dataset, the proposed approach has attained the average $MAE$, $RMSE$, $R^2$, $E_{VAR}$ of 0.02, 0.03, 0.97, and 0.98, respectively, which are superior to those of other compared methods except for the RF and RF-VAR. Nonetheless, the RF is an ensemble learning (EL) method, which consists of multiple decision tree (DT) algorithms. In contrast, the proposed FPCA-SETCN is an independent network method and it attains competitive performance in the experiments of the CALCE battery dataset, which can also be reflected in Fig. 7. From Fig. 7 it can observed that the predicted curve of the capacity degradation is generally consistent with their true curve of the battery capacity degradation change, and the efficacy of the proposed approach is superior to that of all other compared methods, which demonstrates the validity and feasibility of the proposed approach. Besides, it is worth mentioning that the training time consumption of the proposed method is 1 minute and 46 seconds, which is more efficient than LSTM-VAR and slightly worse than the other compared methods. The key explanation for this is that the computational complexity of the model is increased somewhat due to the newly added modules, such as the ESE and SA modules, which results in a slight increase in training time consumption. Nevertheless, the largest discrepancy in training time between our method and the benchmark methods is no more than 2 minutes, which does not pose a significant challenge for the current hardware level.

### *D. Ablation study*

To further evaluate the prediction performance of the proposed approach, we implement ablation study on the model, where we analyze the efficacy of the newly added modules on the original TCN and FPCA component on the test set of both the C-MAPSS and CALCE battery datasets. We separately remove the newly added attention mechanism and the FPCA module from the proposed FPCA-SETCN architecture

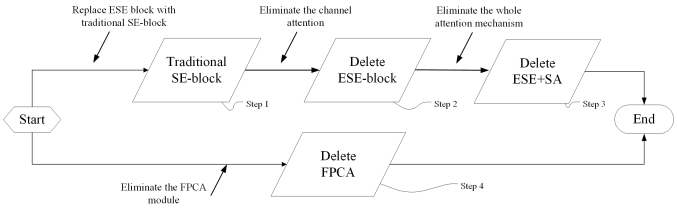



Fig. 8. The flowchart of ablation experiments.

to investigate the performance of the ablation models. Table VI and Table VII display the ablation experiment results on the C-MAPSS and CALCE datasets, respectively. The computational time on training is also provided in both Tables. Fig. 8 depicts a flowchart of the ablation experiments, and the detailed procedure and result analysis are presented as follows.

*1) Ablation on improvement of SE-block:* First, we remove the enhanced SE block and use the traditional one to incorporate it into the network for investigating the efficacy of the enhanced SE block. We notice a slight decrease in the result of the ablated model, where the test $MAE$ and $RMSE$ values of using the traditional SE block rise to 12.73 and 17.60 (increased by 0.68 and 2.04) on the C-MAPSS dataset. The test $MAE$ and $RMSE$ values on the CALCE battery dataset also rise to 0.03 and 0.04 (both increased by 0.01), and the test $R^2$ and $E_{VAR}$ drop to 0.96 (decreased by 0.01 and 0.02), respectively. It is evident that the ablated model experiences a notable decline compared to the proposed architecture of FPCA-SETCN. In terms of computational time, we observe that the running time with the traditional SE block is 8 seconds slower on the C-MAPSS dataset and 20 seconds slower on the CALCE battery dataset compared to using the optimized SE block we proposed. Results indicate that the utilization of the optimized SE block significantly reduces computational overhead. These findings highlight the potential of the proposed model for accelerating model training and inference processes.

*2) Ablation on attention modules:* Subsequently, the ESE block is deleted in the network. An obvious drop in accuracy on this ablation model and the $MAE$ and $RMSE$ values on test dataset increase to 13.97 and 18.46 (increased by 1.92 and 2.90) on the C-MAPSS dataset, as shown in Table VI. Similar phenomena are observed on the CALCE battery dataset. Next, we further remove the whole attention mechanism, including the ESE and SA modules. The $MAE$ value of this ablation model rises to 18.76 and increases by 6.71 relative to that of the proposed approach. The $RMSE$

value of this ablated model also rises to 23.50 and increases by 7.94 on the C-MAPSS dataset. Likewise, the test $MAE$ and $RMSE$ values of this ablated model increase to 0.04 and 0.05 (both increased by 0.02), and the test $R^2$ as well as $E_{VAR}$ drop to 0.95 (decreased by 0.02 and 0.03) on the CALCE battery dataset. Based on the experimental results, it is evident that the attention modules significantly contribute to the performance enhancement of the proposed approach. The absence of the ESE+SA block results in a notable decrease in accuracy compared to the FPCA-SETCN model. However, it is important to acknowledge that, there is an increase in training time. While the incorporation of attention modules introduces a sacrifice in running time, the overall efficiency gains achieved during model prediction are substantial.

*3) Ablation on FPCA:* The third part of the ablation experiment is to evaluate the effect of the FPCA in the RUL prediction. To do so, we eliminate the FPCA module and only use the SETCN to compare the models. We have observed that the test accuracy of the ablated model has a significant decrease on the C-MAPSS dataset, where the $R^2$ drops to 0.76 (decrease by 0.03) and the $RMSE$ rises to 18.38 (increase by 2.82). Similarly, on the CALCE battery dataset, the test $MAE$ and $RMSE$ values of the ablated model also rise to 0.05 (increased by 0.03 and 0.02), and the test $R^2$ and $E_{VAR}$ drop to 0.94 and 0.96 (decreased by 0.03 and 0.02), respectively. The results of the ablation experiment indicate that the performance of the proposed approach is significantly affected by the FPCA module, compared to the performance achieved with the aggregated FPCA module.

### E. Discussion and Analysis

This study has presented a novel end-to-end neural network architecture, namely FPCA-SETCN, to perform the RUL prediction task. In contrast to recent studies, the accuracy of the proposed approach outperforms the results of most existing methods on both the C-MAPSS and CALCE battery datasets. The proposed approach has shown superior advantages and it achieves favorable results in extensive experiments of RUL prediction. The significant improvements in performance achieved by the proposed approach can be attributed to the optimized TCN module, which incorporates Leaky ReLu activations and effectively integrates causal and dilated convolutions. This integration prevents information leakage from the future to the past and facilitates backpropagation, even for negative input values. Besides, the residual connections in the TCN module ensure a long-term valid usage of long-distance sequence data. Moreover, the hybrid attention mechanism that contains the ESE and SA blocks is embedded into the network, boosting its ability to extract multidimensional features and effectively leverage inter-channel correlations and spatial point characteristics. Beyond that, instead of directly feeding raw signals into deep learning models, the degradation signals have been transformed to reflect intrinsic characteristics by FPCA, which has also potentially boosted the prognostic performance of the proposed approach. Conversely, the other methods are commonly-used machine learning methods or single neural networks. The inadequacy of feature extraction in the models constrains the effectiveness of the prediction. Based on the experimental analysis, it can be inferred that the proposed approach has achieved a higher level of effectiveness than traditional methods, exhibiting an impressive performance for RUL prediction.

## IV. Conclusion

Accurately predicting the RUL of mechanical systems is paramount to their maintenance and health management in industrial fields. In this article, a novel FPCA-SETCN deep learning framework was proposed to achieve a reliable and robust RUL prediction. Our approach leverages FPCA to extract changing patterns of multi-stream degradation trajectories and an enhanced TCN with ESE and SA hybrid attention blocks for adaptive feature calibration and optimal feature selection. Concretely, the FPCA method is first used for dimensionality reduction to remove useless information and noise, thereby extracting the crucial features of data. Then, the TCN module is used as the backbone network, and the traditional SE-block is enhanced by substituting the FC layers with the one-dimensional convolutional layers and ReLU activation layer, which prevents unnecessary information loss and decreases the amount of calculations. Combining the ESE and SA blocks, a hybrid attention mechanism is incorporated into the TCN to grasp the optimal features and obtain the sequence information of feature vectors. The proposed framework has the capability to amplify the important features while lessening the weights of negligible features, thus the prediction accuracy can be notably enhanced. Through comprehensive case studies based on the C-MAPSS and CALCE datasets, we validated the effectiveness and feasibility of our approach and demonstrated its superiority over existing deep learning algorithms. However, it is important to acknowledge that, our proposed method may exhibit certain limitations. One potential shortcoming is the computational complexity. Advanced model pruning algorithms can be incorporated to mitigate the computational complexity of the model while maintaining its impressive prediction performance. Moreover, the proposed method can be further extended to other engineering domains and explore the use of additional attention mechanisms to further enhance the precision of RUL prediction.

## References

[1] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2020.

[2] Y. Wen, M. F. Rahman, H. Xu, and T.-L. B. Tseng, "Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective," *Measurement*, vol. 187, p. 110 276, 2022.

[3] M. Kordestani, M. Saif, M. E. Orchard, R. Razavi-Far, and K. Khorasani, "Failure prognosis and applications—a survey of recent literature," *IEEE transactions on reliability*, vol. 70, no. 2, pp. 728–748, 2019.

[4] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical systems and signal processing*, vol. 104, pp. 799–834, 2018.

[5] Y. Wen, J. Wu, and Y. Yuan, "Multiple-phase modeling of degradation signal for condition monitoring and remaining useful life prediction," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 924–938, 2017.

[6] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3208–3216, 2018.

[7] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 international conference on prognostics and health management*, IEEE, 2008, pp. 1–9.

[8] Y. Wang, Y. Ni, N. Li, *et al.*, "A method based on improved ant lion optimization and support vector regression for remaining useful life estimation of lithium-ion batteries," *Energy science & engineering*, vol. 7, no. 6, pp. 2797–2813, 2019.

[9] K. Medjaher, D. A. Tobon-Mejia, and N. Zerhouni, "Remaining useful life estimation of critical components with application to bearings," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 292–302, 2012.

[10] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, April 16-19, 2016, Proceedings, Part I 21*, Springer, 2016, pp. 214–228.

[11] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5695–5705, 2018.

[12] K. Hu, Y. Cheng, J. Wu, H. Zhu, and X. Shao, "Deep bidirectional recurrent neural networks ensemble for remaining useful life prediction of aircraft engine," *IEEE Transactions on Cybernetics*, 2021.

[13] J. Chen, H. Jing, Y. Chang, and Q. Liu, "Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process," *Reliability Engineering & System Safety*, vol. 185, pp. 372–382, 2019.

[14] B. Wang, Y. Lei, N. Li, and W. Wang, "Multiscale convolutional attention network for predicting remaining useful life of machinery," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7496–7504, 2020.

[15] K. Park, Y. Choi, W. J. Choi, H.-Y. Ryu, and H. Kim, "Lstm-based battery remaining useful life prediction with multi-channel charging profiles," *Ieee Access*, vol. 8, pp. 20 786–20 798, 2020.

[16] Y. Song, G. Shi, L. Chen, X. Huang, and T. Xia, "Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory," *Journal of Shanghai Jiaotong University (Science)*, vol. 23, pp. 85–94, 2018.

[17] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, and M. J. Deen, "A data-driven auto-cnn-lstm prediction model for lithium-ion battery remaining useful life," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3478–3487, 2020.

[18] B. Zraibi, C. Okar, H. Chaoui, and M. Mansouri, "Remaining useful life assessment for lithium-ion batteries using cnn-lstm-dnn hybrid method," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4252–4261, 2021.

[19] H. Li, Z. Wang, and Z. Li, "An enhanced cnn-lstm remaining useful life prediction model for aircraft engine with attention mechanism," *PeerJ Computer Science*, vol. 8, e1084, 2022.

[20] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[21] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[22] K. He, Z. Su, X. Tian, H. Yu, and M. Luo, "Rul prediction of wind turbine gearbox bearings based on self-calibration temporal convolutional network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[23] K. Yu, D. Wang, and H. Li, "A prediction model for remaining useful life of turbofan engines by fusing broad learning system and temporal convolutional network," in *2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, IEEE, 2021, pp. 137–142.

[24] Q. Chen, Y.-B. Liu, M.-F. Ge, J. Liu, and L. Wang, "A novel bayesian-optimization-based adversarial tcn for rul prediction of bearings," *IEEE Sensors Journal*, vol. 22, no. 21, pp. 20 968–20 977, 2022.

[25] F. Deng, Y. Bi, Y. Liu, and S. Yang, "Remaining useful life prediction of machinery: A new multiscale temporal convolutional network framework," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[26] H. Sun, M. Xia, Y. Hu, S. Lu, Y. Liu, and Q. Wang, "A new sorting feature-based temporal convolutional network for remaining useful life prediction of rotating machinery," *Computers and Electrical Engineering*, vol. 95, p. 107 413, 2021.

[27] R. Li, Z. Chu, W. Jin, Y. Wang, and X. Hu, "Temporal convolutional network based regression approach for estimation of remaining useful life," in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, IEEE, 2021, pp. 1–10.

[28] D. K. Sharma, S. Brahmachari, K. Singhal, and D. Gupta, "Data driven predictive maintenance applications for industrial systems with temporal convolutional networks," *Computers & Industrial Engineering*, vol. 169, p. 108 213, 2022.

[29] Y. Cao, Y. Ding, M. Jia, and R. Tian, "A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings," *Reliability Engineering & System Safety*, vol. 215, p. 107 813, 2021.

[30] Y. Wang, L. Deng, L. Zheng, and R. X. Gao, "Temporal convolutional network with soft thresholding and attention mechanism for machinery prognostics," *Journal of Manufacturing Systems*, vol. 60, pp. 512–526, 2021.

[31] S. Fu, L. Lin, Y. Wang, *et al.*, "Mca-dtcn: A novel dual-task temporal convolutional network with multi-channel attention for first prediction time detection and remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 241, p. 109 696, 2024.

[32] S. Chung and R. Kontar, "Functional principal component analysis for extrapolating multistream longitudinal data," *IEEE Transactions on Reliability*, vol. 70, no. 4, pp. 1321–1331, 2020.

[33] F. Yan, X. Lin, R. Li, and X. Huang, "Functional principal components analysis on moving time windows of longitudinal data," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 67, no. 4, pp. 961–978, 2018.

[34] J. Chen, W. Chen, A. Zeb, and D. Zhang, "Segmentation of medical images using an attention embedded lightweight network," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105 416, 2022.

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[36] X. Fang, K. Paynabar, and N. Gebraeel, "Multistream sensor fusion-based prognostics model for systems with single failure modes," *Reliability Engineering & System Safety*, vol. 159, pp. 322–331, 2017.

[37] F. Yao, H.-G. Müller, and J.-L. Wang, "Functional data analysis for sparse longitudinal data," *Journal of the American statistical association*, vol. 100, no. 470, pp. 577–590, 2005.

[38] T. Perrin, O. Roustant, J. Rohmer, *et al.*, "Functional principal component analysis for global sensitivity analysis of model with spatial output," *Reliability Engineering & System Safety*, vol. 211, p. 107 522, 2021.

[39] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on dempster–shafer theory and the bayesian monte carlo method," *Journal of Power Sources*, vol. 196, no. 23, pp. 10 314–10 321, 2011.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] R. Thombre, S. Gajbhiye, and M. Dhabu, "Remaining useful life estimation in prognostics and health management using lstm neural network and vector autoregression models," in *Proceedings of ACM/CSI/EEECS Research & Industry Symposium on IoT Cloud For Societal Applications (IoTCloud'21)*, 2021.

[42] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2306–2318, 2016.

[43] Z. Li, J. Wu, and X. Yue, "A shape-constrained neural data fusion network for health index construction and residual life prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 5022–5033, 2020.

[44] Z. Kong, Y. Cui, Z. Xia, and H. Lv, "Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics," *Applied Sciences*, vol. 9, no. 19, p. 4156, 2019.

[45] A. Falcon, G. D'Agostino, O. Lanz, G. Brajnik, C. Tasso, and G. Serra, "Neural turing machines for the remaining useful life estimation problem," *Computers in Industry*, vol. 143, p. 103 762, 2022.

[46] N. Costa and L. Sánchez, "Variational encoding approach for interpretable assessment of remaining useful life estimation," *Reliability Engineering & System Safety*, vol. 222, p. 108 353, 2022.

[47] Z. Zhang, X. Chen, E. Zio, and L. Li, "Multi-task learning boosted predictions of the remaining useful life of aero-engines under scenarios of working-condition shift," *Reliability Engineering & System Safety*, vol. 237, p. 109 350, 2023.

[48] A. F. Azyus, S. K. Wijaya, and M. Naved, "Prediction of remaining useful life using the cnn-gru network: A study on maintenance management," *Software Impacts*, vol. 17, p. 100 535, 2023.