IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 26, 2019, accepted September 15, 2019, date of publication September 23, 2019,
date of current version November 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942991

Remaining Useful Life Estimation Using CNN-XGB With Extended Time Window

XIAOYONG ZHANG[1,2], PENGCHENG XIAO[1,2], YINGZE YANG[1,2], (Member, IEEE), YIJUN CHENG[1,2], BIN CHEN[1,3], DIANZHU GAO[1,2], WEIRONG LIU[1,2], (Member, IEEE), AND ZHIWU HUANG[1,3], (Member, IEEE)

[1]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[2]Hunan Engineering Laboratory of Rail Vehicles Braking Technology, Central South University, Changsha 410083, China
[3]School of Automation, Central South University, Changsha 410083, China

Corresponding author: Yingze Yang (yangyingze@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873353, Grant 61672537, and Grant 61672539, and in part by the 111 Project under Grant B17048.

ABSTRACT The remaining useful life estimation has been widely studied for engineering systems. A system commonly works under varying operating conditions, which may affect the system degradation trajectory differently and consequently reduce the accuracy of remaining useful life estimation. In this paper, we propose CNN-XGB with extended time window to tackle this issue. Firstly, the extended time window is created by feature extension and time window processing in data preprocessing. In feature extension, multiple degradation features are extracted by an improved differential method, and these features are appended to the raw data as additional features. To make the time window cover more information for better prognostic accuracy, a time window padding method is used considering the problem of missing data in some samples. Secondly, a convolutional neural network architecture with multichannel $1 * 1$ filter kernel is proposed considering the effect of varying operating conditions. Furthermore, to improve the prognostic robustness and avoid the sensitivity to the abnormal data, convolutional neural network and extreme gradient boosting are fused by model averaging (CNN-XGB). The validity of the proposed method is verified using aero-engine datasets from NASA.

INDEX TERMS Neural networks, lifetime estimation, time series analysis, prognostics and health management.

I. INTRODUCTION

Prognostic and health management (PHM) has become one of the focuses of preventive maintenance in the systems or components [1], [2]. Anomaly detection and a timely early warning of failure plays an important role in traditional preventive maintenance [3], [4]. The PHM aims to monitor the reliability and security of an engineering system, which improves the maximum operating availability and reduces maintenance cost [5]. The prognostics provide an estimation of the remaining useful life (RUL) of a degradation system or component based upon knowledge of operating history, current state and future operating conditions [6].

The RUL describes the time remaining of an equipment before one or more fault modes occurs [7]. However, many systems usually work under varying operational conditions due to multiple factors such as the ambient environments, operational profiles or workloads [8]. Since different operating conditions may affect the degradation trajectory of a system differently [8], it is difficult that the monitoring data of the system with varying operational conditions (*e.g.* aero-engines [9] and avionics system [10]) are used to estimate RUL. Thus, it is still a challenge to develop an accurate RUL estimation method for a system with varying operating conditions.

For the RUL estimation of a system, the model-based methods describe the fault mechanism and failure process of the system [11]. The commonly used model-based methods include weibull distribution [12] and particle filtering [13]. However, the model-based methods require the online unit to work under the same conditions as the offline units, and require a lot of prior knowledge [14]. Data-driven methods build estimation models based on historical data, which successfully avoid the limitations of getting prior knowledge.

The associate editor coordinating the review of this manuscript and approving it for publication was Nagarajan Raghavan.

Data-driven methods can be classified into direct and indirect approaches [7], [15]. Indirect methods build health indicator and then maps the health indicator to RUL value through degradation model, where similarity measurement is used to match the most similarity degradation model. Direct methods is able to build the RUL estimation model directly from raw data without building health indicator [7].

Indirect approaches have been widely studied for the RUL estimation of system [16]. Wang *et al.* [17] constructed different estimation models for aero-engines under six operating conditions, where the health indicator is constructed by linear regression for each operating condition. But the health indicator constructed by linear regression is not robustness. In recent years, many scholars attempted different methods to improve the quality of health indicator. Yan *et al.* [18] proposed a data fusion methodology, which integrates the information from multiple sensors to construct a health indicator. Gugulothu and Gugulothu [19] proposed Embed-RUL, which uses a sequence-to-sequence model based on recurrent neural networks (RNN) to generate embeddings for sub-sequences. Ramasso [20] proposed remaining useful life estimation based on imprecise health indicator modeled by planar polygons and similarity-based reasoning. Hu *et al.* [21] proposed an ensemble model which combines multiple member algorithms with a weighted-sum formulation.

Overall, indirect methods have strong interpretability because it analyzes the degradation processes of different units, and construct the candidate sets of models for different units. However, the prognostic accuracy heavily depends on the construction of health indicator and degradation model, and the high model complexity results in the longer training time consumption. Considering the limitations of indirect approaches, direct approaches are used in this paper.

Direct approaches focus on the effect of time series information. Khelif *et al.* [7] used support vector regression to estimate RUL directly, where the trend coefficient and the average value are extracted from the time window to create sample sets. Compared with the time window dataset, the training time consumption of [7] is greatly reduced while the certain performance is guaranteed. However, the analysis of time series in [7] is imperfect because it may cause the loss of degradation information.

To ensure the integrity of degradation information, the different direct approaches capable of processing time series are used. Zheng *et al.* [22] applied deep long short-term memory neural network (LSTM) to estimate RUL, which used sensor sequence information to reveal hidden characteristic of the data with multiple operating conditions to some extent. In order to better learn sequence information, Wu *et al.* [23] proposed an improved LSTM, where the dynamic differential features are appended to the raw data to improve the expression ability of data with multiple operating conditions. However, the dynamic differential method did not consider the cycle interval between sensor data in the same operating condition, so it is difficult to enough describe the degradation characteristic of sensor data under varying operating conditions. In this paper, we improve the dynamic differential method to fully dig the degradation information of data with multiple operating conditions.

To make use of sequence information more convenient, the convolution neural network (CNN) based on time window is used for RUL estimation. The deep CNN architecture is designed to learn the features of time window, which established a good mapping between the raw data and RUL value [5], [24]. But they did not consider the impacts of differences in data distributions under varying operating conditions. Besides, the training time consumptions of deep architectures is serious, and their hyper-parameter selection is difficult. Considering the complexity of model and the degradation characteristics of data with varying operating conditions, we develop a CNN based on multichannel $1 * 1$ filter kernel with extended time window, where the extended time window contains sensor data and additional features. However, the multichannel $1 * 1$ filter kernel is focused on the features of one moment, it may be sensitive to the abnormal data.

To improve the prognostic accuracy and robustness, ensemble learning is used for RUL estimation [25]. The fusing of multiple models can effectively reduce noise impact and improve prognostic accuracy [26], [27], which is promising for RUL estimation [28]. In recent years, extreme gradient boosting (XGBoost) has been widely used by data scientists [29], [30] and has achieved promising results in many machine learning challenges [31].

Comparing with the current ensemble of decision trees algorithm, such as AdaBoost, Gradient Boosting Decision Tree, Random Forest, etc., XGBoost is an improved boosting decision tree [32], which adopts the second-order Taylor expansion to improve the computing speed and generalization ability. Then it has the advantages of both boost and RF by boosting multiple trees and subsampling, which can reduce over-fitting and noise interference effectively. But its parameter selection is difficult.

To reduce high dependence on the parameters of single model and combine the advantages of multiple models, we fuse the CNN with XGBoost through model averaging. The proposed method is validated on C-MAPSS turbofan aero-engine datasets from NASA in this paper. Nonetheless, the proposed method can be adapted for other industrial applications.

In summary, the existing data-driven methods for RUL estimation have the problems of high training time consumption or poor accuracy. In this paper, we propose CNN-XGB with extended time window to estimate the remaining useful life of system with varying operating conditions. The contributions of this paper are the three-fold:

- Multiple additional features extracted by improved differential method are appended to the raw data to improve the expression ability of degradation information for sample data. Moreover, a time window padding method is used for the trade-off between prognostic accuracy

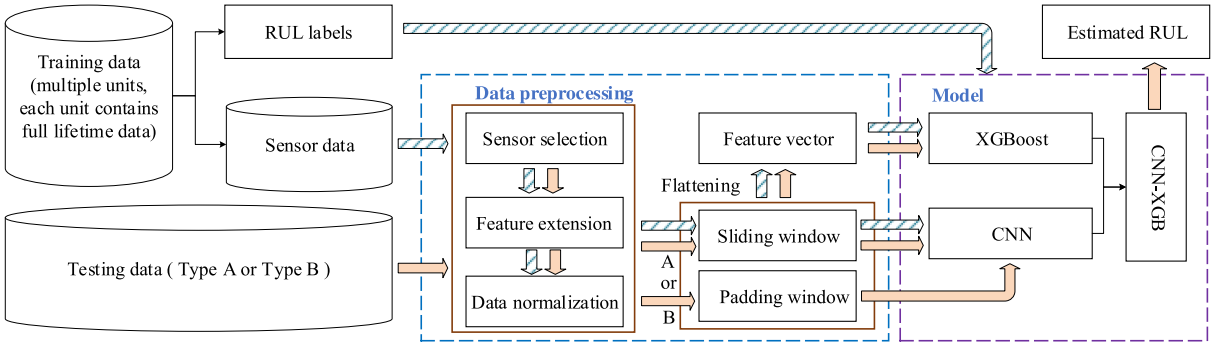


FIGURE 1. The proposed remaining useful life estimation method, where the blue arrow describes the process of modeling, and the orange arrow indicates the prognostic procedure of testing data.

and the problem of missing data, which considers the degradation characteristics of different unit;

- A convolutional neural network based on multichannel $1*1$ filter kernel with extended time window is proposed to estimate RUL, which reduces the impact of varying operating conditions;
- Since the multichannel $1 * 1$ filter kernel is mainly focused on the features of one moment, it may be sensitive to the abnormal data. To improve the prognostic accuracy and robustness, an ensemble model (CNN-XGB) is designed to combine XGBoost with CNN, where the extended time window serves as the input.

The remainder of the paper is organized as follows. Section II describes the remaining useful life estimation method. Section III uses the C-MAPSS aero-engine dataset from NASA to carry out experimental verification of the proposed method. Finally, conclusions are described in Section IV.

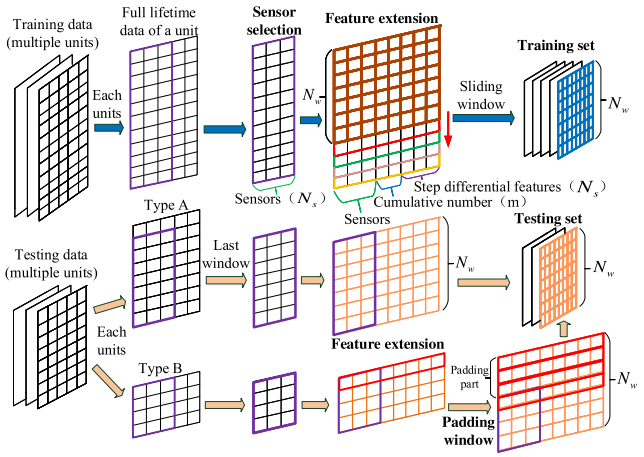


FIGURE 2. The process of data preprocessing, where the training data uses the sliding window to create training set, and the testing data uses the sliding window or padding window to get the testing instance.

II. RUL ESTIMATION METHOD

The proposed remaining useful life estimation method is schematically represented in Fig. 1. The testing data have two types, type A and type B. For the type A, all features corresponding to the real sensing data. While for some test engines, the data recorded before the collection point is less than the length of the slide windows, it is necessary to add the padding data when the sensing data is not sufficient for a slide window, forming the type B. Then, an ensemble model (CNN-XGB) is designed to combine XGBoost with CNN.

A. DATA PREPROCESSING

In this paper, the diagram of data preprocessing is illustrated in Fig. 2. The blue arrow describes the treatment of data preprocessing for training data, and the orange arrow indicates testing data. The rectangle with purple board is the raw sensor data within one slide window. Then the original sensing data is extended by feature extensions, presenting by the red one dimension vector, which are consisted of raw sensor data and extended features. The blue square on the top right of the figure is corresponding to the constructed vector as the training data of learning model. For type B, The red one dimension vector is copied to pad the slide window as the input features of the learning model.

The training data contains the full lifetime data of multiple units, and the lifespan of each unit is different. The testing data contains the recorded cycle data of multiple units, which have two types (see Fig. 1). Firstly, the sensors with the obvious monotonous trends along time are selected as the basic features, which can reflect the aggravation tendency. Secondly, feature extension is used to extract the degradation features for the operation switching. Thirdly, the data are normalized. Finally, sliding window and padding window are used to construct extended time window datasets.

1) FEATURE EXTENSION

Monitoring data usually contains operational history and sensor data [23]. These features may can build RUL estimation model when a system only works in single operating condition. However, it is difficult to build an effective RUL estimation model for the system with multiple operating conditions [23]. To tackle this problem, we extract the effective

features by analyzing the characteristics of sensors data, and the extracted features are appended to the raw data.

In [23], Wu *et al.* proposed a dynamic differential method to extract the inter-frame information from the raw data, where the extracted feature is defined as dynamic differential features. The dynamic differential features can solve the challenge of multiple operating conditions reasonably. However, during the lifespan of a system, the operating conditions are varying, that is, the cycle intervals between sampling data under the same operating condition are varying, which are continuous or discrete. Due to the uncertainty of cycle interval and noise interference, it is imperfect that the differential values of inter-frame data under the same operating condition are used to describe the degradation information of samples.

In this paper, to amplify degeneration trends among sensor data, the step differential method is utilized to extend the feature of the raw sensor data. The step differential method is an improvement on conventional dynamic differential method. In the feature extension part, the differential values is not the current sample data minus last sample data, but the data before n_{step} cycles, and $n_{step} > 1$, which is a constant. The differential values between the sensor values at the current time and the sensor values of past n_{step} cycles have better degradation characteristic. The diagram of step differential method is shown in Fig. 3. In the figure, the condition 1, 2, 3 . . . 6 meaning the different operational modes, which are automatically classified by utilizing clustering algorithm on raw sensing data.

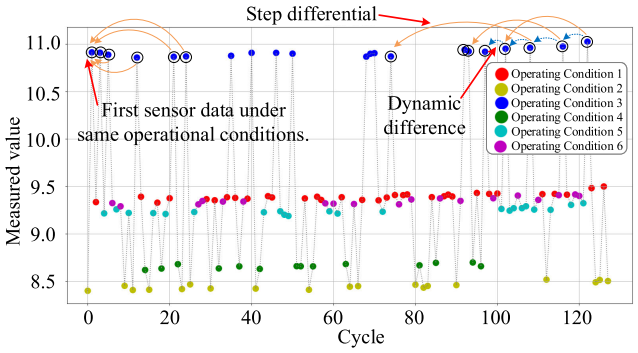


FIGURE 3. The process of step differential method, where the difference step threshold n_{step} is assumed to be 20.

During the lifespan of a system, the operational conditions of system are switched frequently. We extract the differential features of data under different operational conditions separately. In a specific operating condition, if the cycle of sampling data is less than n_{step} at the current time, the differential values between the current sampling data and first sensor data are taken. Otherwise, if the cycle interval between current sampling data and previous is greater than n_{step}, the differential values of them are taken. By contrary, the earlier sensor data is considered, until the cycle interval between the current data and the past data is greater than n_{step}.

The steps of feature extraction are as follows. Firstly, sub-datasets of different operational conditions are clustered by K-means algorithm. Secondly, the cumulative number features of different operating conditions of each unit are extracted, and these features are appended to the raw data as additional features. Thirdly, the step differential values of each cycle data are obtained under the same operational condition, and the step differential value of each sensor data is attached successively behind the cumulative number features as additional feature. Finally, the sub-datasets of different operating conditions are merged by the order of recorded cycle. Given the number of sensors N_s, and the number of operational conditions m. Using the above process of feature extension, the dimension of feature vector for one cycle data becomes $1 \times (2N_s + m)$. When the system only works in single operating condition, the dimension of feature vector is $1 \times 2N_s$ because the cumulative number features is useless.

2) DATA NORMALIZATION

The range of the each feature is normalized to $[-1,1]$ by using min-max normalization,

$$x_{norm}^{i,j} = \frac{2\left(x^{i,j} - x_{\min}^{j}\right)}{x_{\max}^{j} - x_{\min}^{j}} - 1, \quad \forall i, j, \tag{1}$$

where $x^{i,j}$ is the j-th feature of the i-th measuring point, $x_{norm}^{i,j}$ is the $x^{i,j}$ standardized result, and $x_{\max}^{j}$ and $x_{\min}^{j}$ are the maximum and minimum values of the j-th feature respectively.

3) SLIDING WINDOW AND PADDING WINDOW

Sliding window is used to create sample containing information of time series [5], [24], [33]. Suppose the i-th time window is denoted as

$$\begin{aligned} \boldsymbol{x(i)} &= \begin{bmatrix} x^{i-N_w+1,1} & \cdots & x^{i-N_w+1,2N_s+m} \\ \vdots & \ddots & \vdots \\ x^{i,1} & \cdots & x^{i,2N_s+m} \end{bmatrix}^{(N_w \times (2N_s+m))} \\ &= \left[x^{i-N_w+1,:}, x^{i-N_w+2,:}, \cdots, x^{i,:}\right]^T. \end{aligned} \tag{2}$$

where N_w is the size of time window. A new window is generated by shifting one cycle continuously. Thus, the $(i+1)$-th sliding window is represented as

$$\boldsymbol{x(i+1)} = \left[x^{i-N_w+2,:}, x^{i-N_w+3,:}, \cdots, x^{i+1,:}\right]^T \tag{3}$$

Suppose the size of time sequence in testing instance is $\tilde{N}_w$, where $\tilde{N}_w < N_w$. Since the degradation of system is low in the early stage, the degradation path of sensor data is imperceptible. The strategy of repeatedly sampling the feature vector of first cycle $(N_w - \tilde{N}_w)$ times is used to pad the time window. Therefore, the padding window is represented as

$$\hat{\boldsymbol{x}} = [\underbrace{x^{1,:}, \cdots, x^{1,:}}_{N_w - \tilde{N}_w + 1}, x^{2,:}, \cdots, x^{\tilde{N}_w,:}]^T. \tag{4}$$

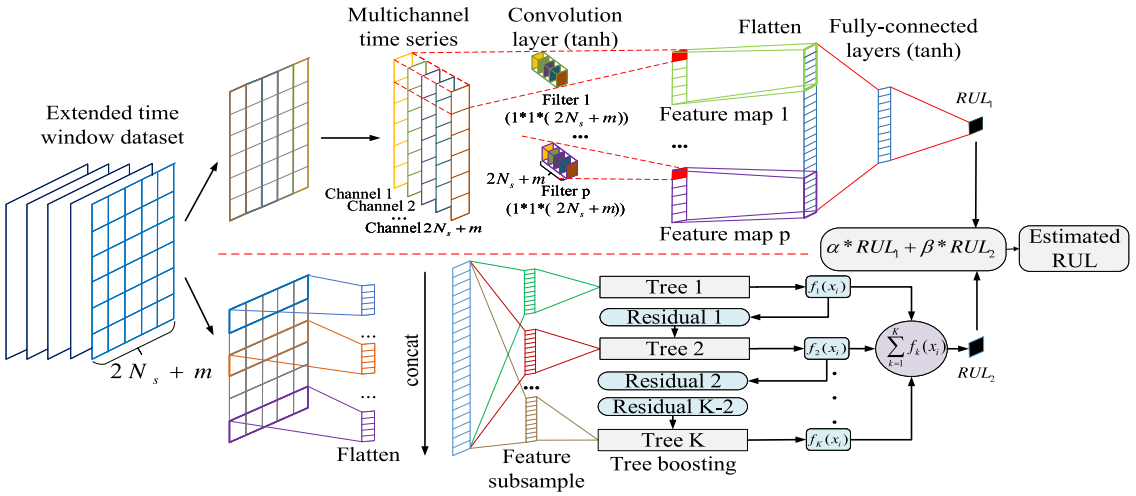


FIGURE 4. The architecture of CNN-XGB using extended time window, where the top part displays the CNN architecture, and the bottom part displays the XGBoost architecture.

B. PROPOSED CNN-XGB

1) CONVOLUTIONAL NEURAL NETWORK

In this paper, each sample is an extended time window, where the vertical is the time series (length N_w), the horizontal is the feature points (width $2N_s + m$). In [5], a one-dimensional convolutional kernel is chosen because the correlation between different sensors are not significant. However, in data preprocessing, we have enriched the feature points of each moment in time window, thus enhancing the correlation between features. Considering the difference and relevance of the representations of different features, a time window can be converted to a multichannel time series [34], where the number of channel is $2N_s + m$. Each channel denotes a time series of one feature. Multichannel convolutional can fuse the multiple features into one feature map, thus reducing the training time consumption. The time series of the q-th channel is represented as

$$x_q = [x^{1,q}, \cdots, x^{N_w,q}]^T, \quad \forall q = 1, \cdots, (2N_s + m). \tag{5}$$

The convolutional neural network provides multiple filter kernels in convolutional to better learn features. Filtering in the convolutional layer can be expressed as the inner product between a filter kernel w and a concatenation vector [5]. In this paper, the convolutional kernel w is a three-dimensional tensor, which is successively expressed as the length, width and channel. Suppose the p-th filter kernel is $w_p (w_p \in \Re^{F_L \times 1 \times (2N_s+m)})$, and the i-th subsequence of the q-th channel is $x^{i:i+F_L-1,q}$, which can be represent as

$$x^{i:i+F_L-1,q} = x^{i,q} \oplus x^{i+1,q} \oplus \cdots \oplus x^{i+F_L-1,q}, \tag{6}$$

where F_L is the length of the filter kernel, and $\oplus$ is a concatenation operation on data points. The output $z^{i,p}$ represents the i-th feature of the multichannel time series through the p-th filter kernel, which is given by

$$z^{i,p} = \varphi\left(b_p + \sum_{q=1}^{2N_s+m} {w_p}^T x^{i:i+F_L-1,q}\right), \tag{7}$$

where ${w_p}^T ({w_p}^T \in \Re^{1 \times F_L \times (2N_s+m)})$ is the matrix transpose of the p-th filter kernel, φ is the activation function, and b_p is the bias term of the p-th filter. Through sliding of the filter kernel from the first point to the last point in the multichannel time series, the feature map of the p-th filter is expressed as

$$z^p = \left[z^{1,p}, z^{2,p}, \cdots, z^{N_w-F_L+1,p}\right]^T. \tag{8}$$

In this paper, compared with convolutional kernel of long sequence, the advantages of $F_L = 1$ are the two-fold:

- The time consumption of convolutional operation is significantly reduced, because each filter kernel has only one parameter in a channel;
- The multichannel $1 * 1$ filter kernel used for time series can reduce the impact of varying operational conditions, because the convolution of each feature points in time series is irrelevant to the previous cycle.

In this paper, the proposed convolutional neural network architecture is schematically described in the top part of Fig. 4. Compared with [5], the architecture of this paper only contains input layer, one convolutional layer, one flatten layer, one full connection layer and output layer. Flatten layer is used for the transition from convolutional layer to full-connection layer, which completes the flattening connection of the feature maps of multiple filter kernels. The full-connected layer performs regression operation using the features learned by convolutional layer. The *tanh* activation function is used for convolutional layer, full connection layer and output layer, and *Xavier* normal initializer is used to the weight initializations [5]. The *Adam* optimizer is used for parameter optimization, and mean squared error is used for the loss function. The number of multichannel $1 * 1$ filter kernel is set as fifteen, and the number of neuron in the full connection layer is set as five.

2) EXTREME GRADIENT BOOSTING

In this paper, the input of XGBoost is a feature vector created by flattening a extended time window. Given the dataset $D = \{(x_i, y_i)\}(x_i \in \Re^{(N_w*(2N_s+m))}, y_i \in \Re, i \leq n)$, where n is the number of samples. XGBoost uses K additive functions to predict the output, which is defined as

$$\hat{y}_i = \varphi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F, \tag{9}$$

where $F = \left\{f(x) = \omega_{q(x)}\right\}\left(q : \Re^m \to T, \omega \in \Re^T\right)$ is a set of regression tree, $\hat{y}_i$ is the predicted value, q represents the structure of each tree, which maps a sample to the corresponding leaf index, T is the number of leaves, and f_k corresponds to a separate tree structure q and weight w. The target function is defined as,

$$L(\varphi) = \sum_i l\left(y_i, \hat{y}_i\right) + \sum_k \Omega(f_k). \tag{10}$$

where l is a differentiable convex loss function and Ω is the regularization term.

In the additive model, the incremental f_t is designed to minimize the prediction error. To quickly optimize the target function in the general case, the second order Taylor expansion is applied. According to reference [31], by using the second-order Taylor expansion of the square loss function to approximate the loss function derivation of the least square method, the solving of the least square method can be removed and the performance of optimization speed and generalization are improved.

$$\begin{aligned} L^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \\ &\simeq \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t), \end{aligned} \tag{11}$$

where $\hat{y}_i^{(t)}$ indicates the prediction of the i-th sample at the t-th iteration, g_i and h_i are first and second order statistics of loss functions respectively. More detailed description of the XGBoost is given in [31]. The XGBoost architecture is schematically described in the bottom part of Fig. 4.

3) MODEL FUSION

The accuracy of fusion model depends on the performance of single model and the diversity of multiple models. In general, the correlations between the different models should be as small as possible, and their performance should not vary much. The commonly used model fusion methods of data-driven are averaging, bagging, boosting and stacking. However, the training time consumptions of bagging, boosting and stacking are all high [35]. In this paper, model averaging is used for model fusion considering the training time consumption. Firstly, the CNN model and the XGBoost model are used for RUL prediction respectively. Then, model averaging is applied to obtain the final RUL value, that is, coefficients α and β are 0.5. The detailed process of the proposed ensemble model (CNN-XGB) is shown in Fig. 4.

III. EXPERIMENTAL STUDY

A. TURBOFAN AERO-ENGINE DATASET

The C-MAPSS turbofan aero-engine datasets from NASA [36] are widely used in prognostic studies, which contains four cases, FD001, FD002, FD003 and FD004 dataset. Each case contains training set, testing set and testing RUL values, and the monitoring data of each case is consist of 21 sensors and 3 operation settings [9].

Training set contains data from normal to failure under the varying operational conditions and fault modes. Each engine unit has varying degrees of initial wear and manufacturing differences. Over time, the engine units degrade until they reach the system failure, that is, the cycle of last data recorded is described as the failure cycle of engine unit. Each recorded cycle is used as the training sample, and is associated with a RUL label. In the testing set, the sensor data for each engine unit is terminated at some cycles before system failure. The last recorded cycle data for each engine unit is used as the testing instance. The goal of the experiment is to estimate the remaining useful life value of each engine unit in the testing set [5]. The detailed information of the C-MAPSS datasets is described in Table 1.

TABLE 1. The detail information of the C-MAPSS dataset.

Dataset	FD001	FD002	FD003	FD004
Engine units (training)	100	260	100	249
Engine units (testing)	100	259	100	248
Operational conditions	1	6	1	6
Fault modes	1	1	2	2
Minimum lifespans (training)	128	128	146	128
Maximum lifespans (training)	362	378	526	544
Minimum cycles recorded (testing)	31	21	38	19
Maximum cycles recorded (testing)	303	367	475	486

B. PERFORMANCE METRICS

In this paper, two metrics are used to estimate prognostic performance, *i.e.* scoring function and root mean square error (RMSE) [5]. The scoring function is formulated as

$$\begin{aligned} s &= \sum_{i=1}^{N} s_i, \\ s_i &= \begin{cases} e^{-\frac{d_i}{13}} - 1, & for\ d_i < 0, \\ e^{\frac{d_i}{10}} - 1, & for\ d_i \geq 0, \end{cases} \end{aligned} \tag{12}$$

where s indicates the score, N indicates the number of testing instances, $d_i = RUL'_i - RUL_i$ indicates the prediction error of the i-th instance, RUL'_i and RUL_i denote the predicted RUL

and the actual RUL value of the i-th instance respectively. Another metric is RMSE, which is given by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2}. \tag{13}$$

.

C. DATA PREPROCESSING

The data preprocessing description in Section II. Fourteen sensors are selected, which retrieval is 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21. Then, FD002 and FD004 datasets are processed with the cumulative number features and the step differential features, while FD001 and FD003 datasets are only processed with the step differential features because they only work in single operational condition. Besides, cut-off point T_{cut} is used to capture valid samples. Compared with maximum cycles recorded of testing engine units, the discarded samples are manifested to its actual RUL value which is too large in training set. The T_{cut} is slightly less than the maximum cycles recorded of testing engine units.

The degradation trajectory of a system is usually expressed as a nonlinear function. In the related research, piecewise linear function have been shown to have excellent performance [1], [5], [20], [22], which is expressed as

$$label_c = \begin{cases} label_{real}, & if\ label_{real} \le R_{early} \\ R_{early}, & if\ label_{real} > R_{early}, \end{cases} \tag{14}$$

where $label_{real}$ denotes the actual remaining useful life value of training set sample, R_{early} is the remaining useful life threshold and $label_c$ denotes the transformed label. The R_{early} is a key parameter in piecewise function processing, which determines the starting time of linear degradation of a system and the upper limit of remaining useful life values.

Furthermore, the label is scaled to [0,1],

$$label_s = \frac{label_c}{R_{early}}, \tag{15}$$

where $label_s$ denotes the scaled label of training set sample. To restore the actual remaining useful life value of testing set instance, the estimated value should be multiplied by R_{early}.

The generating condition of different datasets leads to different parameter settings of data preprocessing. The main parameters are difference step threshold n_{step}, window size N_w, RUL threshold R_{early} and cutoff point T_{cut}. The parameter adjustment range is set by referring to [5, 17, 23]. Based on their research works, the candidate set of each parameter is determined, and the final setting of parameters is carried out by using grid searching. Figure 5 presents the effect of different parameters on experimental results. And the parameters chosen for the datasets are shown in Table 2.

D. EXPERIMENTAL RESULT AND ANALYSIS

The experimental environment is Inter Core i5-4460 (3.20GHz) CPU, 8GB memory, Microsoft Windows 10 ultimate operation system and anaconda3 with python3.6.

TABLE 2. The parameter selection of data preprocessing.

Parameter	Description	FD001	FD002	FD003	FD004
n_{step}	Difference Step threshold	15	50	15	50
N_w	Window size	31	35	38	40
R_{early}	RUL threshold	125	160	125	160
T_{cut}	Cut-off point	300	300	450	450

Besides, the deep learning library "Keras", the machine learning library "scikit-learn" and library "xgboost" are applied.

The experimental results are averaged by ten trials to reduce the randomness. The parameters of proposed ensemble model are determined by 10-fold cross validation. For the CNN model, the learning rate of *Adam* optimizer is 0.0002, batch size is 128 and the number of iterations is 150. For the XGBoost model, the parameters are presented as follows: subsample rate of data is 0.7, subsample rate of features is 0.7, minimum sum of leaf weight is 3, step size shrinkage is 0.1, the number of iterations is 600 and the other are default.

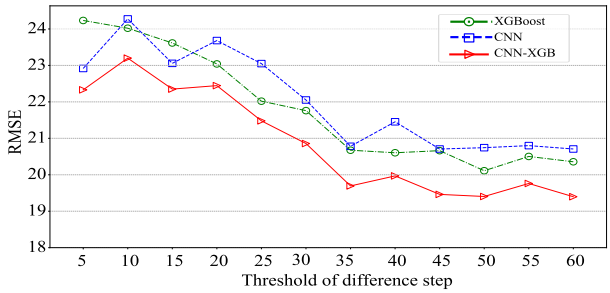
In this paper, a various of experiments have been designed to verify the effectiveness of the proposed RUL estimation method. Firstly, the effects of key parameters are studied on FD004 dataset (six operational conditions and two fault modes), including difference step threshold n_{step}, window size N_w and RUL threshold R_{early}. Secondly, the validity of the step differential features is further analyzed by comparing the dynamic differential features and raw time window for the FD004 dataset. In addition, the performance of multiple convolutional neural network architectures under different strategies of feature extension are discussed. Thirdly, the performance of the proposed method and other popular methods are analyzed in four cases. Finally, the superiority of the proposed method is proved by comparing with other state-of-the-art results in recent years.

1) EFFECTS OF KEY PARAMETERS

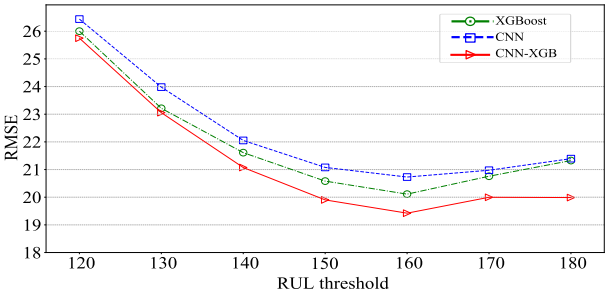
The effects of key parameters of data preprocessing are analyzed with FD004 dataset, for instance. Table 1 describes the detailed information of FD004 dataset, and Fig. 3 shows the data distribution of a sensor in six operational conditions.

The difference step threshold n_{step} is a key parameter in feature extension. Fig. 5(a) shows the effect of the n_{step} on prognostic accuracy. Since the degradation characteristic is increased with cycle interval, the prognostic accuracy is greatly improved when n_{step} increases from 5 to 45, and $n_{step} = 50$ is the best. When the cycle interval is too short, the degradation characteristic of sensor data is imperceptible. However, overlong cycle interval is not practical for real scenes due to the limitation of lifespan. Thus, an appropriate difference step threshold can significantly improve expression ability of degradation information for sample data.

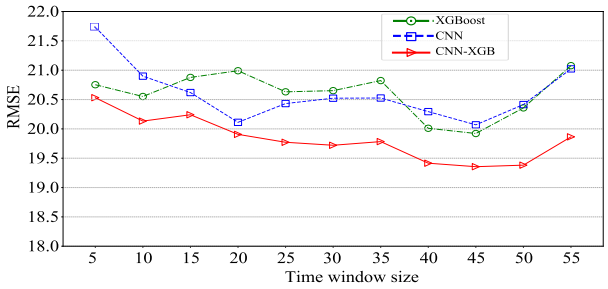
The effect of RUL threshold R_{early} is shown in Fig. 5(b). It is appropriate for $R_{early} = 160$ when dealing with data



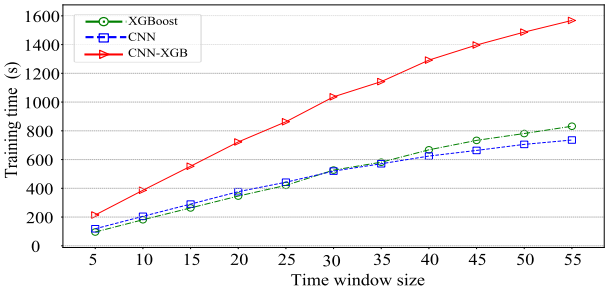
(a) The effect of the parameter n_{step} on prognostic accuracy.



(b) The effect of the parameter R_{early} on prognostic accuracy.



(c) The effect of the parameter N_w on prognostic accuracy.



(d) The effect of the parameter N_w on training time consumption.

FIGURE 5. The effects of the four key parameters for FD004 dataset.

with varying operational conditions. However, since the R_{early} restricts the upper limit of estimated RUL value, the prediction bias is large when the actual RUL of instance is overlong. Note that the R_{early} should be different for four datasets because the generation condition of datasets. The experimental results on four cases show that $R_{early} = 125$ is suitable when the system only works in single operating condition, and $R_{early} = 160$ is more reasonable when the system works in six operating conditions.

Fig. 5(c) and Fig. 5(d) show the effects of the time window size N_w on prognostic accuracy and training time consumption respectively. More information can be covered by larger time window, which can obtain better result. However, the training time consumption increases linearly with window size. When window size increases from 5 to 40, the prognostic accuracy is significantly improved. The $N_w = 40$ is most appropriate for the trade-off between prognostic accuracy and the training time consumption. It is obvious that the padding method has a significant effect because the minimum cycle recorded of the testing unit is 19 for FD004 dataset. Besides, CNN-XGB is distinctly better than the single-model. It is worth noting that when the system requires real-time ability, the proposed convolutional neural network architecture with a shorter time window is a good choice.

In summary, these key parameters have significant effects on prognostic accuracy. However, the parameters are usually related to the generation conditions of datasets, such as the lifespan of different units, the number of units, operational conditions, the failure process of components and so on.

2) COMPARING WITH DIFFERENT STRATEGIES OF FEATURE EXTENSION

In direct approaches, the performance of RUL estimation is heavily determined by the quality of features. To manifest the superiority of step differential features, a series of experiments are designed. Firstly, we design three strategies of feature extension and four convolutional neural network architectures, where the three strategies are time window without additional features, time window with dynamic differential features and time window with step differential features. The four convolutional neural network architectures are similar, one of which is the proposed convolutional neural network architecture in Section II. The convolutional kernels of other architectures are expressed as

- Convolutional neural network architecture A: the convolutional kernel is single channel with $10 * 1$ [5];
- Convolutional neural network architecture B: the convolutional kernel is single channel with $1 * 1$;
- Convolutional neural network architecture C: the convolutional kernel is multichannel with $10 * 1$.

It is worth noting that the inputs of architecture A and B are not converted to a multichannel time series. Table 3 compares the prognostic performance of multiple convolutional neural network architectures under different strategies of feature extension. On the one hand, the validity of the step differential features is verified by comparing the dynamic differential features and raw time window. On the other hand, the effectiveness of the proposed convolutional neural network architecture is proved by comparing with other architectures. It can be observed that the proposed convolutional neural network

TABLE 3. The prognostic performance of multiple convolutional neural network architectures under different strategies of feature extension for FD004 dataset.

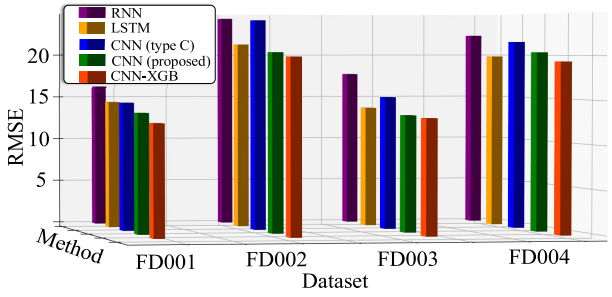
	Time window without additional features				Time window with dynamic differential features				Time window with step differential features			
CNN architectures	A	B	C	**proposed**	A	B	C	**proposed**	A	B	C	**proposed**
RMSE	33.8	28.8	33.2	**28.6**	24.4	24.6	27.6	**22.9**	21.8	20.3	22.20	**20.3**
Scoring	45280.4	10514.5	32378.2	**11086.8**	5925.0	6680.0	13226.8	**4013.7**	6688.1	4408.2	6366.8	**5088.1**
Time	1137.7	928.8	599.3	**306.6**	3009.7	2670.4	1124.8	**648.1**	2843.2	2493.2	1245.4	**621.7**

architecture under the step differential strategy has better performance.

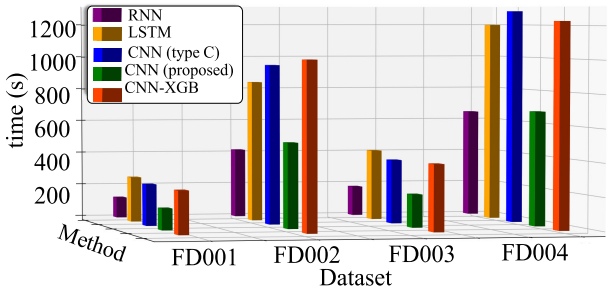
In table 3, our proposed algorithm suffer a little higher value on SCORE when achieving the lowest RMSE. As the Scoring Function has a great punishment on large errors due to its exponential form. While RMSE's punishment increases linearly with deviation, which makes the performance more stable. The test set contains lifetimes of 248 engines. The higher scoring may be caused by certain single deviation in the prediction of individual engines, which may not reflect the overall prediction performance actually.

3) COMPARING WITH OTHER POPULAR METHOD

The results are compared with other popular methods to verify the superiority of proposed method. In four cases, RNN and LSTM with the similar architecture and convolutional neural network architecture C are used. Compared with the CNN architecture, the RNN and LSTM replace one convolutional layer with one corresponding recursive layer with containing 15 neurons.



(a) RMSE



(b) Training time consumption

FIGURE 6. The prognostic accuracy and training time consumption by different methods on four datasets. (a) RMSE. (b) Training time consumption.

The accuracy and training time consumption of different methods are shown in Fig. 6(a) and Fig. 6(b) respectively.

TABLE 4. The performance comparisons of the proposed method and the latest papers on the C-MAPSS dataset.

Method	Metric	FD001	FD002	FD003	FD004	R_{early}
RULCLIPPER [20]	RMSE	13.27	22.89	16.00	24.33	135
	Scoring	216.0	2796.0	317.0	3132.0	
DCNN [24]	RMSE	18.45	30.29	19.82	29.16	No
	Scoring	1286.7	13570.0	1596.2	7886.4	
MODBNE [25]	RMSE	15.04	25.05	12.51	28.66	No
	Scoring	334.2	5585.3	421.9	6557.6	
LSTM [22]	RMSE	16.14	24.49	16.18	28.17	130
	Scoring	338.0	4450.0	852.0	5550.0	
DCNN [5]	RMSE	12.61	22.36	12.64	23.31	125
	Scoring	273.7	10412.0	284.1	12466	
Semi-supervised setup [37]	RMSE	12.56	22.73	12.10	22.66	GA
	Scoring	231.0	3366.0	251.0	2840.0	
Proposed CNN	RMSE	**13.59**	**20.03**	**13.23**	**20.29**	125/160*
	Scoring	**287.91**	**2533.56**	**285.21**	**4892.15**	
Proposed	RMSE	**12.61**	**19.61**	**13.01**	**19.41**	125/160*
CNN-XGB	Score	**224.73**	**2525.99**	**279.36**	**2930.65**	

* The RUL threshold R_{early} of FD001 and FD003 dataset is set as 125, and the RUL threshold R_{early} of FD002 and FD004 dataset is set as 160. GA: A genetic algorithm is used to select R_{early}.

On the one hand, the prognostic accuracy and training time consumption of the proposed convolutional neural network architecture is superior to the architecture C. There are two reasons for this phenomenon. One reason is the number of trainable parameters of proposed convolutional neural network architecture is lower than the architecture C. Another is the convolution of different operational conditions data is not conducive to the expression of degradation information. On the other hand, the prognostic accuracy of RNN is significantly inferior to the proposed convolutional neural network architecture. The prognostic accuracy of LSTM is slightly inferior to the proposed convolutional neural network architecture, but its training time consumption is serious. Besides, the prognostic accuracy of CNN-XGB is obviously superior to the single-model in the four cases, which indicates that the proposed ensemble model is suited for prognostic problem in complex environments. The CNN-XGB provides reliable remaining useful life estimation while the training time consumption is acceptable.

To better analyze prognostic characteristic, the actual RULs and estimated RULs of testing units compared in Fig. 7, in which the testing units are sorted by actual RUL in ascending order. This figure shows the estimated RUL values of proposed convolutional neural network architecture and

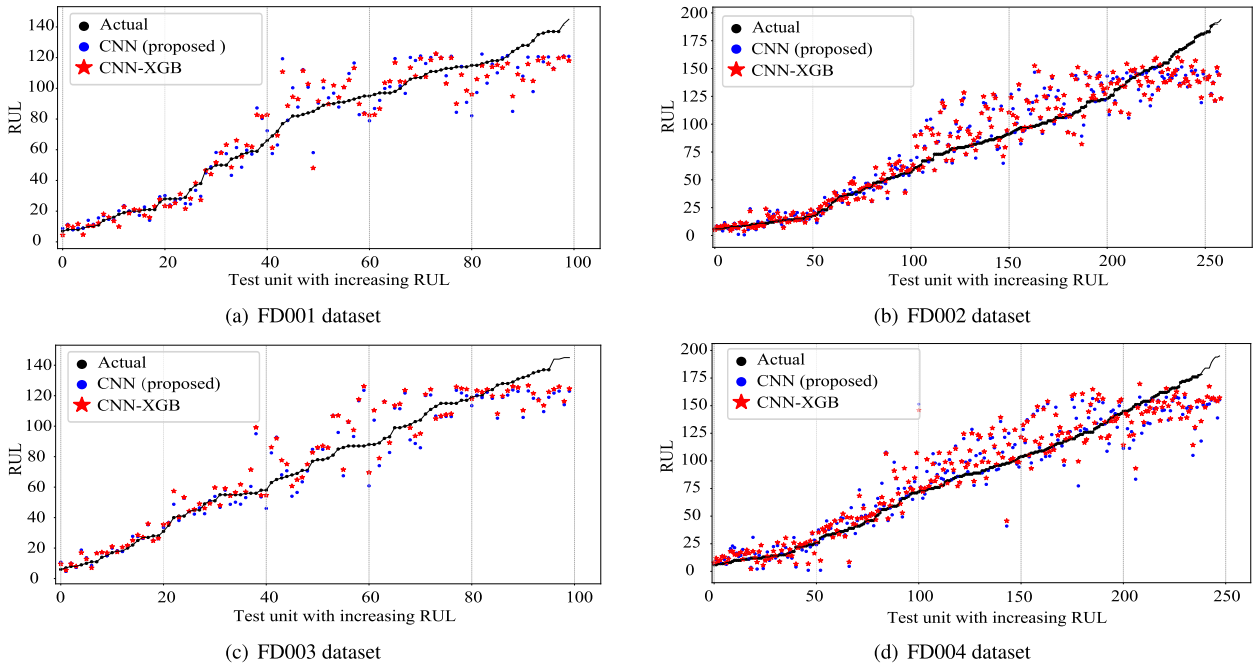


FIGURE 7. The sorted prediction of the testing engine units for the four datasets.

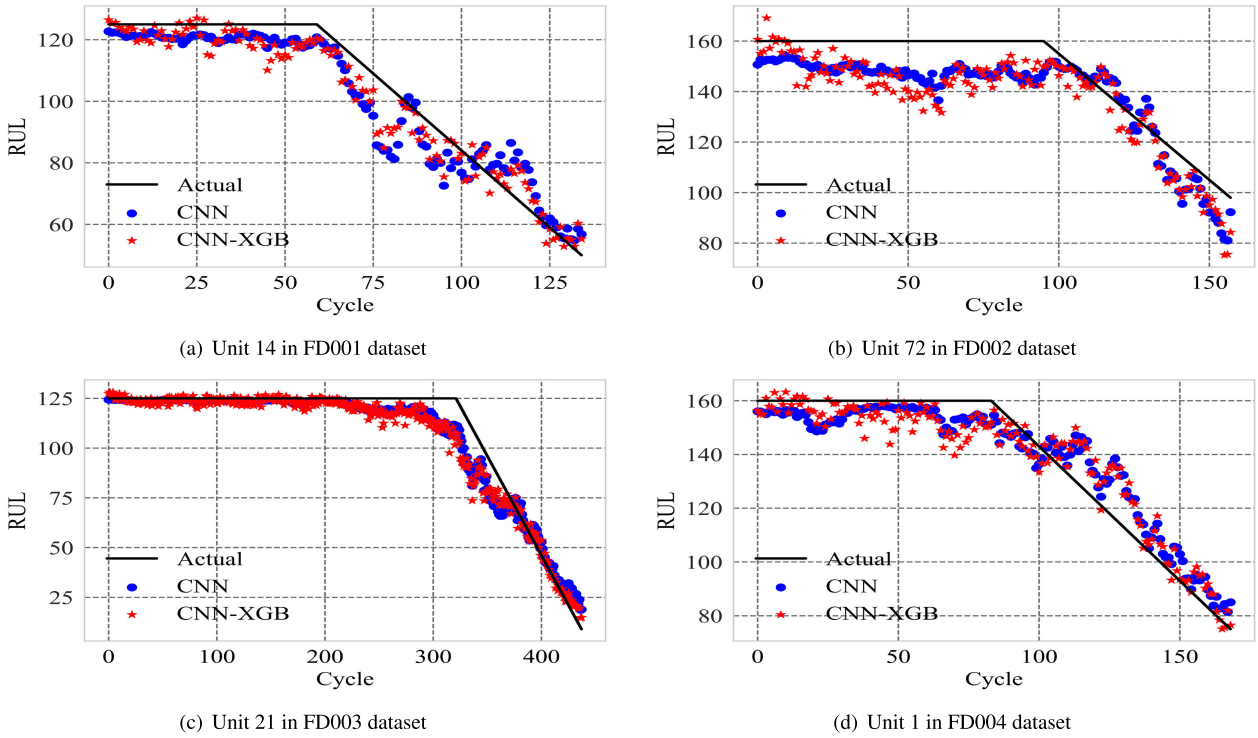


FIGURE 8. The prediction of the selected test engines during lifetime in the four datasets.

CNN-XGB are close to the actual RUL, where CNN-XGB is better than CNN in four cases.

It is worth noting that in Fig. 7, on the right of the figure, the test units run from comparable health status. Then the degradation characteristics are not obvious at that time, which make RUL prediction performance weaker. Close to the left of the figure, the test units aggravate gradually from the start and the degradation characteristics gradually strengthen. Thus all the methods can achieve more accuracy for testing units with low RUL values and the RUL prediction

performance is significantly improved. Since the failure features become more prominent when the working state of the engine is closer to failure.

To illustrate the performance of the proposed algorithm, Fig. 8 presents the relationship between the predicted life RUL and the time of some selected sample test pieces. In the figure, it is shown that similar to Fig. 7, the prediction performance is comparable lower for large real RUL. However, CNN-XGB achieves more accurate prediction performance than CNN during the whole lifetime of test units. It is verified that combining the CNN having simple structure with XGBoost can improve the prediction performance.

4) COMPARING WITH RELATED WORKS

Table 4 summaries the latest research results on the four datasets of C-MAPSS. The proposed methods have obtained prospective compared with the state-of-the-art results. It is worth noting that the proposed convolutional neural network architecture manifests highly competitive in both accuracy and real-time ability. In real application, considering the complexity of model fusion and training time consumption, model averaging of multiple models is practical.

In this paper, compared with the CNN, the accuracy of CNN-XGB is significantly improved. On the other side, although the deep convolutional neural network (DCNN) proposed in [5] has the similar performance to our proposed method, it has more bulky scale and needs more data compared to the ensemble of the CNN and XGBoost. Moreover, compared with other recent advanced works, the proposed CNN-XGB algorithm is competitive on 4 C-MAPSS datasets. Especially on multiple condition datasets, CNN-XGB achieves the best value on RMSE index, which validates its effectiveness on complex industrial applications.

IV. CONCLUSION

In this paper, we propose CNN-XGB with extended time window to estimate the remaining useful life of engineering systems with varying operating conditions. Firstly, the extended time window is created by feature extension and time window processing in data preprocessing. In feature extension, a step differential method is used for extracting degradation features from the raw data, and the step differential features are appended to the raw data as additional features. To make the time window cover more information, a time window padding method is used considering the problem of missing data. Secondly, considering the effect of varying operating conditions, CNN-XGB with extended time window is proposed for accurate RUL estimation, where the CNN architecture is simplified as much as possible while the accuracy is guaranteed. Experimental analysis on the C-MAPSS aero-engine dataset from NASA proves the effectiveness of the proposed method. Compared with the latest advanced methods, the proposed method has shown the excellent prognostic performance. In the future, de-noising algorithm will be embedded in the CNN-XGB.

REFERENCES

[1] Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Rel. Eng. Syst. Safe.*, vol. 164, pp. 74–83, Aug. 2017.

[2] E. Balaban, A. Saxena, S. Narasimhan, I. Roychoudhury, M. Koopmans, C. Ott, and K. Goebe, "Prognostic health-management system development for electromechanical actuators," *J. Aerosp. Inf. Syst.*, vol. 12, no. 3, pp. 329–344, Mar. 2015.

[3] B. Huang, K. Cohen, and Q. Zhao, "Active anomaly detection in heterogeneous processes," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2284–2301, Apr. 2018.

[4] Y. Cheng, J. Peng, X. Gu, X. Zhang, W. Liu, Y. Yang, and Z. Huang, "RLCP: A reinforcement learning method for health stage division using change points," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2018, pp. 1–6.

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

[6] R. Guo and Q. Gan, "Prognostics for a leaking hydraulic actuator based on the f-distribution particle filter," *IEEE Access*, vol. 5, pp. 22409–22420, 2017.

[7] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2276–2285, Mar. 2017.

[8] T. Tao, E. Zio, and W. Zhao, "A novel support vector regression method for online reliability prediction under multi-state varying operating conditions," *Rel. Eng. Syst. Saf.*, vol. 177, pp. 35–49, Sep. 2018.

[9] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–9.

[10] G. P. Pandian, D. Das, L. I. Chuan, E. Zio, and M. Pecht, "A critique of reliability prediction techniques for avionics applications," *Chin. J. Aeronaut.*, vol. 31, no. 1, pp. 10–20, Jan. 2018.

[11] M. J. Daigle, A. Bregon, and I. Roychoudhury, "Distributed prognostics based on structural model decomposition," *IEEE Trans. Rel.*, vol. 63, no. 2, pp. 495–510, Jun. 2014.

[12] Q. Zhang, C. Hua, and G. Xu, "A mixture Weibull proportional hazard model for mechanical system failure prediction utilising lifetime and monitoring data," *Mech. Syst. Signal Process*, vol. 43, pp. 103–112, Feb. 2014.

[13] S. Li, H. Ma, T. K. Saha, Y. Yang, and G. Wu, "On particle filtering for power transformer remaining useful life estimation," *IEEE Trans. Power Del.*, vol. 33, no. 6, pp. 2643–2653, Dec. 2018.

[14] Y. Zhang, R. Xiong, H. He, and M. Pecht, "Lithium-ion battery remaining useful life prediction with Box–Cox transformation and Monte Carlo simulation," *IEEE Trans. Ind. Electron*, vol. 66, no. 2, pp. 1585–1597, Feb. 2018.

[15] F. Yang, M. S. Habibullah, T. Zhang, Z. Xu, P. Lim, and S. Nadarajan, "Health index-based prognostics for remaining useful life predictions in electrical machines," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2633–2644, Apr. 2016.

[16] Z. Liang, J. Gao, H. Jiang, X. Gao, Z. Gao, and R. Wang, "A similarity-based method for remaining useful life prediction based on operational reliability," *Appl. Intell.*, vol. 48, no. 9, pp. 2983–2995, Sep. 2018.

[17] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, pp. 1–6, Oct. 2008.

[18] H. Yan, K. Liu, X. Zhang, and J. Shi, "Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions," *IEEE Trans. Reliab.*, vol. 65, no. 3, pp. 1416–1426, Aug. 2016.

[19] N. Gugulothu, V. Tv, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, "Predicting remaining useful life using time series embeddings based on recurrent neural networks," 2017. *arXiv:1709.01073*. [Online]. Available: https://arxiv.org/abs/1709.01073

[20] E. Ramasso, "Investigating computational geometry for failure prognostics," *Int. J. Prognostics Health Manage.*, vol. 5, no. 1, p. 5, 2014.

[21] C. Hu, B. D. Youn, P. Wang, and J. T. Yoon, "Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life," *Rel. Eng. Syst. Saf.*, vol. 103, pp. 120–135, Jul. 2012.

[22] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2017, pp. 88–95.

[23] Y. T. Wu, M. Yuan, S. Dong, L. Li, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018.
[24] G. S. Babu, P. Zhao, and X. L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2016, pp. 214–228.
[25] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2016.
[26] L. Peel, "Data driven prognostics using a Kalman filter ensemble of neural network models," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–6.
[27] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–6.
[28] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Trans. Rel.*, vol. 63, no. 1, pp. 191–207, Mar. 2014.
[29] W. Li, S. Ding, Y. Chen, and S. Yang, "Heterogeneous ensemble for default prediction of peer-to-peer lending in China," *IEEE Access*, vol. 6, pp. 54396–54406, 2018.
[30] Y. Yang, P. Xiao, Y. Cheng, W. Liu, and Z. Huang, "Ensemble strategy for hard classifying samples in class-imbalanced data set," in *Proc. IEEE Int. Conf. Big Data Smart Comput.*, Jan. 2018, pp. 170–175.
[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. SIGKDD ACM*, Aug. 2016, pp. 785–794.
[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[33] P. Lim, C. K. Goh, and K. C. Tan, "A time window neural network based framework for remaining useful life estimation," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2016, pp. 1746–1753.
[34] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and K. Shonali, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Conf. Joint Artif. Intell.*, Jun. 2015, pp. 3995–4001.
[35] I. Syarif, E. Zaluska, A. Prugelbennett, and G. Wills, "Application of bagging, boosting and stacking to intrusion detection," in *Proc. Int. Conf. Mach. Learn. Data Mining*, 2012, pp. 593–602.
[36] A. Saxena and G. Kai, "Turbofan engine degradation simulation data set," NASA Ames Prognostics Data Repository, NASA Ames Res. Center, Moffett Field, Mountain View, CA, USA, Tech. Rep., 2008. [Online]. Available: http://ti.arc.nasa.gov/project/prognostic-data-repository
[37] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Rel. Eng. Syst. Saf.*, vol. 183, pp. 240–251, Mar. 2019.

XIAOYONG ZHANG received the B.S., M.S. and Ph.D. degrees in computer application technology from Central South University, Changsha, China, in 2000, 2005, and 2009, respectively, where he is currently an Associate Professor with the School of Information Science and Engineering. His current research interests include intelligent transportation, train network control, and prognostics and health management.

PENGCHENG XIAO received the B.Eng. degree in measurement and control technology and instruments from the Huaihai Institute of Technology, Lianyungang, China, in 2016. He is currently pursuing the M.S. degree in electronics science and technology with the School of Information Science and Engineering, Central South University, China. His current research interests include intelligent fault diagnosis, and prognostics and health management.

YINGZE YANG received the B.S. and Ph.D. degrees in control theory and control engineering from Central South University, in 2006 and 2010, respectively, where he is currently a Lecturer with the School of Information Science and Engineering. His current research interests include communication performance optimization, distributed fault diagnosis, and prognostics and health management.

YIJUN CHENG received the B.S. degree from the School of Resources and Safety Engineering, Central South University, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Information Science and Engineering. Her current research interests include fault diagnosis, and prognostics and health management.

BIN CHEN received the B.S. degree from the School of Information Science and Engineering, Central South University, China, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include robotic control, train antiskid control, and prognostics and health management.

DIANZHU GAO received the B.Eng. degree from Southwest Jiaotong University, Chengdu, China, in 1999, and the M.S. degree from Central South University, Changsha, China, in 2009, where he is currently pursuing the Ph.D. degree with the School of Information Science and Engineering. His current research interests include intelligent control technology, intelligent transportation, and prognostics and health management.

WEIRONG LIU (M'14) received the B.S. and M.S. degrees from the School of Information Science and Engineering, Central South University, Changsha, China, and the Ph.D. degree from the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with the School of Information Science and Engineering, Central South University. His special fields of interests include cooperative control, nonlinear control, and wireless sensor networks.

ZHIWU HUANG (M'08) received the B.S. degree from Xiangtan University, in 1987, the M.S. degree from the University of Science and Technology Beijing, in 1989, and the Ph.D. degree from Central South University, China, in 2006. From 2008 to 2009, he was with the School of Computer Science and Electronic Engineering, University of Essex, U.K., as a Visiting Scholar. He is currently a Professor with the School of Information Science and Engineering, Central South University. His research interests include fault diagnostic, co-operative control, and cloud computing.