Bi-LSTM based Two-Stream Network for Machine Remaining Useful Life Prediction

Ruibing Jin, Zhenghua Chen, *Senior Member, IEEE*, Keyu Wu, Min Wu, *Senior Member, IEEE*, Xiaoli Li, *Senior Member, IEEE*, and Ruqiang Yan, *Fellow, IEEE*

***Abstract*—In industry, prognostic health management (PHM) is used to improve the system reliability and efficiency. In PHM, remaining useful life (RUL) prediction plays a key role in preventing machine failure and reducing operation cost. Recently, with the development of deep learning technology, the long short-term memory (LSTM) and convolutional neural networks (CNN) are adopted into many RUL prediction approaches, which show impressive performances. However, existing deep learning based methods directly utilize raw signals. Since noise widely exists in raw signals, the quality of these approaches' feature representation is degraded, which degenerates their RUL prediction accuracy. To address this issue, we firstly propose a series of new handcrafted feature flows (HFFs), which can suppress the raw signal noise, and thus improve the encoded sequential information for the RUL prediction. Additionally, to effectively integrate our proposed HFFs with the raw input signals, a novel Bi-LSTM based two-stream network is proposed. In this novel two-stream network, three different fusion methods are designed to investigate how to combine both streams' feature representations in a reasonable way. To verify our proposed Bi-LSTM based two-stream network, extensive experiments are carried out on the C-MAPSS dataset, showing superior performances over state-of-the-art approaches.**

***Index Terms*—Remaining Useful Life (RUL) Prediction, Bi-directional LSTM, Two-stream Network, Deep Learning, Time Series**

I. INTRODUCTION

Remaining useful life (RUL) prediction acts as a core task in the prognostics and health management (PHM), which predicts the machinery failure, prevents accidents and lowers operation cost. Generally, the RUL prediction refers to use time series data from multiple sensors to predict the remaining life of a machine. Currently, in the RUL prediction task, existing approaches fall into two categories: model based and data driven methods. In the model based methods [1]–[3], the prior knowledge on a mechanical system (or component) is required to formulate the system's degradation characteristics. However, with the rapid progress of industry, mechanical systems become complex, and extensive prior knowledge is required in model based methods. This makes it difficult to apply these methods to predict the RUL. In comparison, data driven methods regard the mechanical system as a black box. Instead of requiring the dynamics of a system, data driven methods only need to collect the sensor data, which makes these methods applicable to complex mechanical systems.

Recently, deep learning technology adventures rapidly and shows its promising performance in many industry applications [4]–[7]. Some data driven methods [4]–[6], [8]–[15] which utilize the deep learning technology, are proposed for the RUL prediction. The approach [8] firstly investigates the effectiveness of the deep learning technology in the RUL prediction, where a convolutional neural network (CNN) is proposed. They extend the convolution and pooling layers to the temporal dimension for capturing the temporal relationship among signals. After that, several CNN based methods [5], [6], [11] are proposed to explore the temporal feature capability for CNN based methods.

Apart from it, long short-term memory (LSTM) is very popular in the nature language processing field, which is good at learning the long-term information or capturing the sequential information. Since sequential information is important for the RUL prediction, LSTM is widely applied in many methods [8]–[10] to predict the RUL. The approach [10] directly utilizes LSTM to predict RUL by exploiting the long-term dependencies. To further improve the capacity of LSTM in capturing the long-term dependencies, [8] proposes a bi-directional LSTM based approach, which can accept time series data from two directions and shows better performances than traditional LSTM based approaches. In [9], the attention mechanism is embedded into the LSTM, which further improves the prediction accuracy further by investigating the relationship across different time steps. In [16], a reconstruction loss is proposed for enhancing the encoded feature representation.

Deep learning technologies like CNN and LSTM are used in many approaches for the RUL prediction. However, these methods are directly applied on the raw signals including random noise. This kind of random noise may affect the performances of these approaches. In Fig. 1, we illustrate four time sequences from sensors. Affected by this random noise, the raw signals fluctuate. The encoded sequential information based on this raw signals is degenerated. Approaches only based on the raw signals cannot accurately predict the RUL.

To solve this problem, we propose a novel Bi-LSTM based two-stream network in this paper. Our proposed Bi-LSTM based two-stream network aims to effectively integrate the raw signals with an additional sequence of signals called handcrafted feature flows (HFFs), which are proposed by us in this paper. Different from raw signals, our proposed HFFs

Ruibing Jin, Zhenghua Chen, Keyu Wu, Min Wu, and Xiaoli Li are with Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, Singapore 138632 (Email: jin_ruibing@i2r.a-star.edu.sg, chen0832@e.ntu.edu.sg, wu_keyu@i2r.a-star.edu.sg, wumin@i2r.a-star.edu.sg, xlli@i2r.a-star.edu.sg).
Ruqiang Yan is with School of Mechanical Engineering at Xi'an Jiaotong University, China (Email: yanruqiang@xjtu.edu.cn).

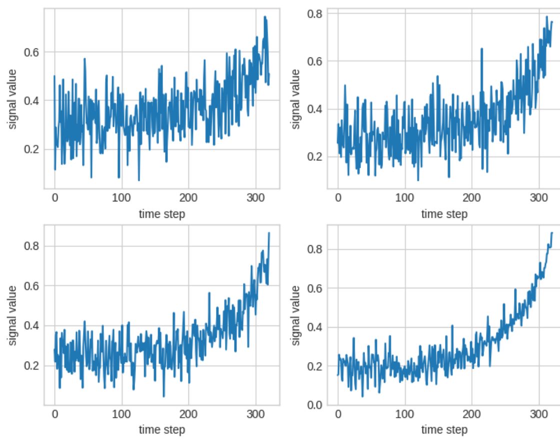


Fig. 1: Four sequences of raw sensor signals are shown. Noise widely exists in these signals.

are computed statistically among the temporal axis, which can suppress the influence caused by the random noise from raw signals. Our proposed HFFs clearly show the sensor signal variation trend. Apart from it, bi-directional long short-term memory (Bi-LSTM) has shown a strong capacity to capture the sequential information. To integrate our HFFs with the raw input signals, we propose a Bi-LSTM based two-stream network. In our proposed Bi-LSTM based network, we regard a Bi-LSTM as an unit, and utilize several Bi-LSTM units to form a stream for processing the input data. Two streams are used to capture the sequential information for raw input signals and the HFFs, respectively. After getting these two produced features, a fusion module is proposed to combine these features together and forward the combined feature to a regressor. To effectively combine these two features, we investigate different fusion methods and propose an effective fusion module for our two-stream network. Extensive experiments are conducted on the well-known C-MAPSS dataset [17], showing state-of-the-art performances.

This work is extended from its preliminary version [18]. We summarize the main changes made to our original work as follows. 1) A series of novel handcrafted feature flows are proposed. Compared with our preliminary handcrafted feature, our proposed handcrafted feature flows not only suppress the raw signal noise, but also provide temporal information for the RUL prediction. 2) Compared with our preliminary work, a new Bi-LSTM based two-stream network is proposed to effectively fuse raw signals and our handcrafted feature flows. With our proposed handcrafted feature flows and the two-stream network, the RUL can be predicted accurately.

The main contributions in this paper can be summarized as follows:

1) Handcrafted feature flows are proposed to suppress the noise existing in raw input signals. With our proposed HFFs, we can easily capture the sequential information, and improves the RUL prediction accuracy.
2) We propose a Bi-LSTM based two-stream network. Several different fusion methods are proposed to effectively integrate our HFFs with the raw input signals.

The remaining of the paper is organized as follows: a comprehensive review on related works is given in Section II. In Section III, we firstly present our proposed handcrafted feature flows and then propose our Bi-LSTM based two-stream network. Experimental settings, network hyper-parameters, evaluation methods, and our experimental results are discussed at Section IV. Following it, we present our conclusion in Section V.

II. Related Work

RUL prediction has been studied for tens of years. Recently, with the remarkable progress of the deep learning technology, many deep learning based methods are proposed for the RUL prediction. In this section, we review some advanced deep learning based methods in the RUL prediction.

In the RUL prediction, deep learning based methods can be roughly divided into three categories: CNN based methods, LSTM based methods and CNN-LSTM based methods. In CNN based methods, [4] firstly explores the effectiveness of CNN on the RUL prediction, showing a significant improvement than existing methods. After that, [5] proposes a time window to augment the training data and uses this augmented data for the neural network training. A multi-scale CNN [19] is proposed to combine both local and global information together for a better performance. The approach [6] proposes a double-CNN to divide the RUL prediction into two stages and provides an accurate RUL prediction. MSCAN [20] proposes to utilize a self-attention scheme to exploit the distinction between multiple sensor data and develops a multi-scale learning strategy to capture the sequential information. Multi-Head Net [21] is proposed to consider the detailed time sequence information for the RUL prediction.

Apart from it, some methods are proposed according to LSTM regarding to its strong capacity in capturing temporal features. [10] proposes a LSTM based method, which tries to fully utilize the sequential dependencies and learns the hidden pattern existing in time series. The method [9] introduces an attention mechanism to the LSTM for further improvement. Although LSTM is able to learn the long-term dependency, it only accepts the time sequence in a single direction and the potential feature information is not fully exploited. To alleviate this issue, [22] adopts the bi-directional LSTM, which learns temporal information from two directions and provides a better performance. In deep learning, gated recurrent unit (GRU) is also widely used to capture the sequential information. The method [23] combines the GRU with an attention scheme together to predict the RUL. Following it, a bi-directional GRU is proposed in BiGRU [24], where an attention scheme and a skip connection are applied to exploit the long term dependencies among multiple sensor data. To further improve the accuracy in the RUL prediction, a type of CNN-LSTM based methods [8], [11], [25] are proposed, where they use a CNN to extract features from raw data and utilize the LSTM to encode the temporal information. These CNN-LSTM based methods produce some impressive performances.

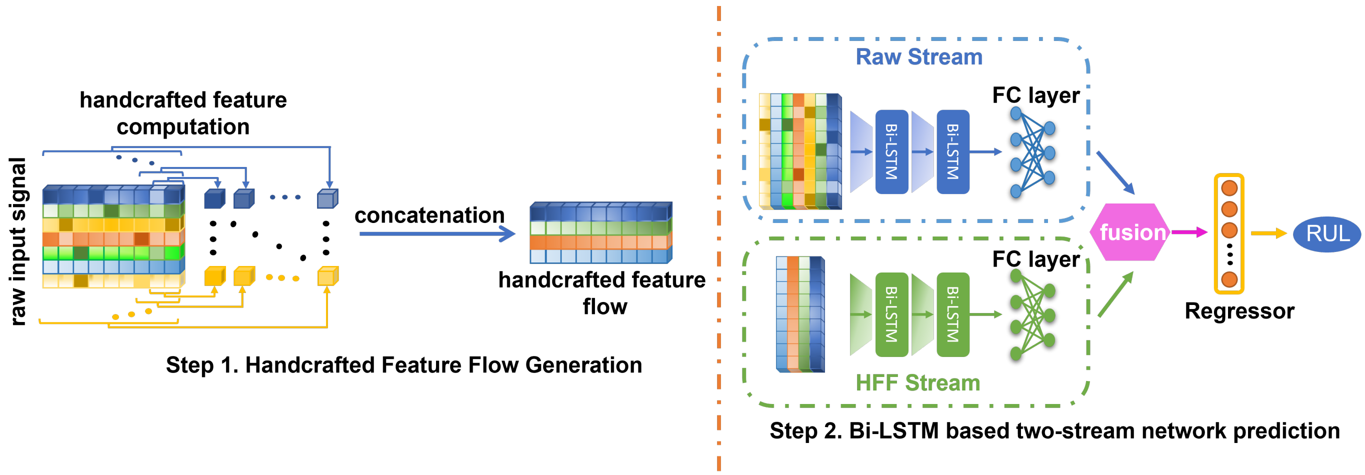


Fig. 2: The overview of our proposed method.

Recently, contrastive learning [26], [27] shows great success in improving the feature representation for deep learning based approaches. Several approaches [28], [29] propose to utilize the contrastive learning to improve the RUL prediction performance. CADA [28] designs a contrastive loss [30] to learn the domain-invariant features. In the method [29], a Siamese network is employed to prevent the neural network from over-fitting during the training process.

Various deep learning based methods have been proposed. However, these methods directly accept the raw sensor data which contains noise. This noise from raw signals may affect the performances for these methods. To solve this problem, we propose the handcrafted feature flows (HFFs). Then, a Bi-LSTM based two-stream network is designed to effectively fuse both raw input signals and our HFFs together, producing state-of-the-art performances.

III. Methodology

In this section, we present our proposed handcrafted feature flows and our bi-directional LSTM (Bi-LSTM) based two-stream neural network. The pipeline of our proposed method is illustrated in Fig. 2. Given the raw sensor signals, $X \in R^{T\times C}$, where T is the length of the signal and C represents the sensor number. We firstly generate the corresponding handcrafted feature flows (HFFs) $\tilde{X}$. Then, both raw input signal X and the HFFs $\tilde{X}$ are forwarded to our proposed Bi-LSTM based two-stream network to predict the RUL prediction.

A. Handcrafted Feature Flows

In this subsection, a series of new handcrafted feature flows (HFFs) are proposed to suppress the noise existing in the raw input signals.

To produce our proposed HFFs, several handcrafted features are computed at first. We adopt two kinds of handcrafted features: amplitude average and the trend coefficient of linear model [18]. Given a specified temporal length t, the amplitude average for the sequence of signals from the sensor c is computed according to Eq. 1.

$$H^c_{avg} = F_{avg}(X^c, t) = \frac{1}{t}\sum_{i=1}^{t} X^c(i), \tag{1}$$

where $X^c(i)$ indicates the signal value from sensor c at ith time step.

Amplitude average only reflects the mean value within a period of time. To provide more information for describing the raw signals, we propose to use a polynomial to fit the raw signals and use the corresponding coefficients to describe the raw signals. This process can be formulated as:

$$H^c_{pf} = F_{pf}(X^c, t). \tag{2}$$

Concretely, we select a simple but effective polynomial model to fit the given raw signals as follows:

$$\hat{y}(i) = w_0 + w_1 x(i). \tag{3}$$

where w_0 and w_1 are the coefficients. To estimate these two coefficients, the least squares method is applied to minimize the difference: $||w_0 + w_1 x(i) - y(i)||^2$, where $y(i)$ represents the real signal value. These coefficients are used as our another handcrafted feature. Generally, to effectively capture the sequential feature, we pre-process the given raw data by normalizing them. Thus, w_0 becomes zero, and we use w_1 as our another handcrafted feature.

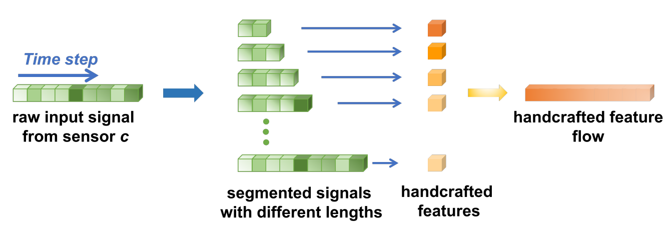


Fig. 3: The pipeline of our proposed handcrafted feature flow generation.

It is challenging to use these two handcrafted features above to indicate the sequential information for time series signals.

To solve this problem, we generate the HFFs by computing these features under different temporal length. As illustrated in Fig. 3, given a sequence of input signal with T temporal length from the sensor c, we firstly compute our handcrafted features H^c_{avg} and H^c_{pf} in varied temporal length t, which increases from 2 to T. After that, we concatenate them together according to Eq. 4.

$$\begin{aligned} F_{hff}(X^c) = \text{concat}(&F_{avg}(X^c,2), F_{pf}(X^c,2), F_{avg}(X^c,3), \\ &F_{pf}(X^c,3), F_{avg}(X^c,4), F_{pf}(X^c,4), \ldots, \\ &F_{avg}(X^c,T), F_{pf}(X^c,T)). \end{aligned} \tag{4}$$

For the input signals from C sensors, we compute C groups of handcrafted feature flows and combine them together via a concatenation operation as follows:

$$\begin{aligned} \tilde{X} = \text{concat}(&F_{hff}(X^1), F_{hff}(X^2), \\ &F_{hff}(X^3), \ldots, F_{hff}(X^C)), \end{aligned} \tag{5}$$

where $\tilde{X}$ indicates the final produced HFFs. Given the raw input signals $X \in R^{T\times C}$, we can obtain our HFFs, $\tilde{X} \in R^{T-1\times 2C}$. T is the window size, and its value is indicated in the Sec. IV

With our proposed handcrafted feature flows, the noise in the raw input signals is suppressed, and the sequential information can be easily captured. This improves the RUL prediction accuracy.

B. Bi-LSTM based Two-stream Network

To effectively combine our proposed HFFs with the raw input signals, we propose a Bi-LSTM based two-stream network in this subsection.

As shown in the right side of Fig. 2, our proposed Bi-LSTM based two-stream network consists of two streams, raw stream and HFF stream. The raw stream is used to capture the sequential information from the raw input signals, and the HFF stream is utilized to process our proposed HFFs. In our proposed two-stream network, we stack two consecutive Bi-LSTM units into a stream for extracting temporal features. After that, two fully connected layers are adopted as a feature decoder. Finally, two encoded feature vectors are combined together in the fusion module, which are used to predict the final results.

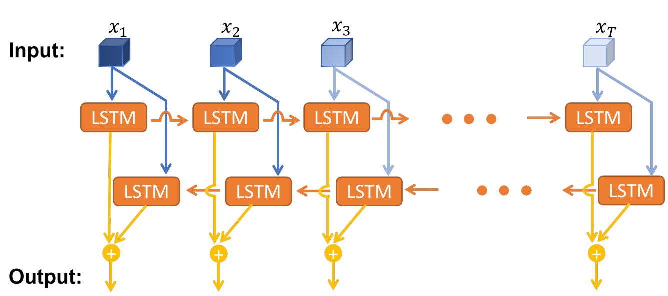


Fig. 4: The illustration for bi-directional LSTM.

Bi-directional LSTM (Bi-LSTM) has been widely used in the RUL prediction task [8], [22], [31], which has shown a strong capacity in capturing the sequential information. As illustrated in Fig. 4, compared with the vanilla LSTM, Bi-LSTM receives the sequence of data in both two directions: forward and backward. For the forward direction, the computation process can be formulated as follows:

$$\begin{aligned} \overrightarrow{h}(t) &= f(\overrightarrow{x}(t), \overrightarrow{h}(t-1), w) \\ &= \begin{cases} \overrightarrow{i}(t) = \sigma(\overrightarrow{w}_{ii}\overrightarrow{x}(t) + \overrightarrow{w}_{hi}\overrightarrow{h}(t-1)) \\ \overrightarrow{f}(t) = \sigma(\overrightarrow{w}_{if}\overrightarrow{x}(t) + \overrightarrow{w}_{hf}\overrightarrow{h}(t-1)) \\ \overrightarrow{g}(t) = \tanh(\overrightarrow{w}_{\text{ig}}\overrightarrow{x}(\text{t}) + \overrightarrow{w}_{\text{hg}}\overrightarrow{h}(\text{t}-1)) \\ \overrightarrow{o}(t) = \sigma(\overrightarrow{w}_{io}\overrightarrow{x}(t) + \overrightarrow{w}_{ho}\overrightarrow{h}(t-1)) \\ \overrightarrow{c}(t) = \overrightarrow{f}(t) \odot \overrightarrow{c}(t-1) + \overrightarrow{i}(t) \odot \overrightarrow{g}(t) \\ \overrightarrow{h}(t) = \overrightarrow{o}(t) \odot \tanh(\overrightarrow{c}(\text{t})) \end{cases}, \end{aligned} \tag{6}$$

where $\overrightarrow{i}(t)$, $\overrightarrow{f}(t)$, $\overrightarrow{g}(t)$, and $\overrightarrow{o}(t)$, indicate the gates for input, forget, cell and output at time t for the forward direction, respectively. $\overrightarrow{h}(t)$ and $\overrightarrow{c}(t)$ represent the hidden and cell state at time t in the forward direction, respectively. σ denotes the sigmoid function, and $\odot$ indicates the Hadamard product. $\overrightarrow{w}_{ii}$, $\overrightarrow{w}_{hi}$, $\overrightarrow{w}_{if}$, $\overrightarrow{w}_{hf}$, $\overrightarrow{w}_{ig}$, $\overrightarrow{w}_{hg}$, $\overrightarrow{w}_{io}$, and $\overrightarrow{w}_{ho}$, are the learnable weights.

Similar to the forward directional process, the backward process is computed as below:

$$\begin{aligned} \overleftarrow{h}(t) &= f(\overleftarrow{x}(t), \overleftarrow{h}(t+1), w) \\ &= \begin{cases} \overleftarrow{i}(t) = \sigma(\overleftarrow{w}_{ii}\overleftarrow{x}(t) + \overleftarrow{w}_{hi}\overleftarrow{h}(t+1)) \\ \overleftarrow{f}(t) = \sigma(\overleftarrow{w}_{if}\overleftarrow{x}(t) + \overleftarrow{w}_{hf}\overleftarrow{h}(t+1)) \\ \overleftarrow{g}(t) = \tanh(\overleftarrow{w}_{\text{ig}}\overleftarrow{x}(\text{t}) + \overleftarrow{w}_{\text{hg}}\overleftarrow{h}(\text{t}+1)) \\ \overleftarrow{o}(t) = \sigma(\overleftarrow{w}_{io}\overleftarrow{x}(t) + \overleftarrow{w}_{ho}\overleftarrow{h}(t+1)) \\ \overleftarrow{c}(t) = \overleftarrow{f}(t) \odot \overleftarrow{c}(t+1) + \overleftarrow{i}(t) \odot \overleftarrow{g}(t) \\ \overleftarrow{h}(t) = \overleftarrow{o}(t) \odot \tanh(\overleftarrow{c}(\text{t})) \end{cases}. \end{aligned} \tag{7}$$

To effectively combine the outputs from two directions, we apply the element-wise addition fusion on the Bi-LSTM:

$$o_{final}(t) = \overrightarrow{o}(t) + \overleftarrow{o}(t). \tag{8}$$

After two consecutive Bi-LSTM layers, two fully connected layers are adopted as feature encoders. Following by it, a fusion module is proposed to fuse both streams.

In our proposed two-stream network, the raw stream predicts the RUL according to the raw input signals, while the HFF stream estimates the RUL based on the HFFs which are produced by aggregating handcrafted features across time steps. Based on the raw signals, the raw stream can rapidly capture the characteristic variation of a machine's RUL, but may be easily affected by the random noise. In comparison, with our proposed HFFs, the HFF stream utilizes the history information to predict the RUL. The influence of random noise can be suppressed, but a late prediction may be produced in the HFF stream. To effectively use these two streams, we proposed three different fusion methods, as shown in Fig. 5.

Late Fusion. An intuitive approach to use these two streams is to weighted sum the predictions from these two streams. In our late fusion, we fuse the two streams at the result level, where the same weight is given on these two predictions for a weighted summation. Concretely, as illustrated in Fig. 5a, we use these two streams to predict the RUL, respectively. Then, we compute the average on these predictions. This process can be formulated as follow:

$$Y_o = Avg(Y_r(f_r(X)), Y_h(f_h(\tilde{X}))), \tag{9}$$

where $f_r(X)$ and $f_h(\tilde{X})$ indicate the produced feature vector for the raw stream and the HFF stream, respectively, and Y_r and Y_h represent the regressor prediction for the raw stream and the HFF stream, respectively. Our late fusion method can be realized easily by running each stream separately. This does not require additional GPU memory. In the late fusion, we assume that the predictions from these two streams are equally important. However, this assumption may be not consistent to the real scenario. Apart from it, a more complex fusion method may be more effective than the addition operation. To combine these two streams better, we propose another two fusion methods.

Concatenation Fusion. Different from the late fusion, our proposed concatenation fusion method combines the two streams on the feature level. We use two streams to encode the useful information, respectively, and then provide these encoded features to a decoder. In this fusion method, we do not specify a specific fusion method, but leave this to the decoder for learning an effective fusion approach. As illustrated in Fig. 5b, given two encoded feature vectors: $f_r \in R^d$ and $f_h \in R^d$, the fused feature vector $f_o \in R^{d'}$ is computed as follow:

$$f_o = Concate(f_r(X), f_h(\tilde{X})). \tag{10}$$

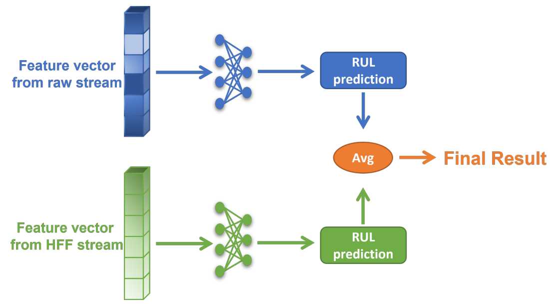
This operation directly concatenate two feature vectors together, which results in the dimension $d' = d + d$. After that, we forward this fused feature vector into a fully connected layer and a regressor to predict the RUL. The concatenation operation increases the dimension of the feature vector from d to d', and expands the model size. This may increase the difficulty in training a neural network. To solve this issue, we propose another fusion method.

Addition Fusion. Different from the concatenation fusion method, we specifically defined an element-wise addition operation to fuse the features from two streams. Since a neural network produces different response values to different input signals, we assume that the encoded feature from each stream is the weighted feature vector. Based on this assumption, we propose the element-wise addition fusion method. As displayed in Fig. 5c, we directly add one feature vector to another feature vector. This process is defined at Eq. 11.

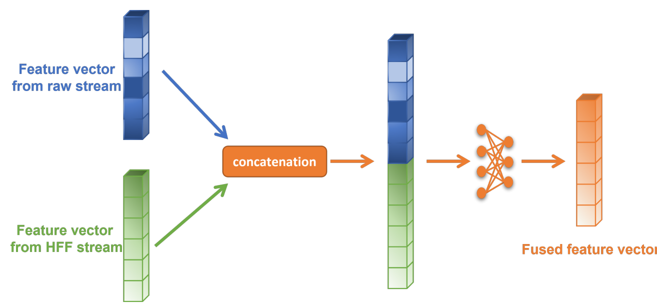
$$f_o = f_r(X) + f_h(\tilde{X}), \tag{11}$$

where both feature vectors are combined together by the element-wise addition operation and the dimension of the fused feature vector is not increased.

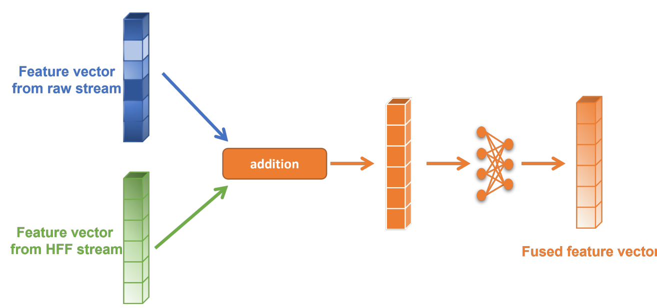
We implement these three different fusion methods in our experiment part and investigate their performances on the RUL prediction. With our proposed Bi-LSTM based two-stream network, both raw input signals and our HFFs are effectively combined, which improves the performance on the RUL prediction.



(a) Late Fusion



(b) Concatenation Fusion



(c) Addtion Fusion

Fig. 5: Our proposed fusion methods.

IV. Experiments

A. Dataset Discreption

A widely used dataset, commercial modular aero propulsion system simulation (C-MAPSS) [17], is used to verify the effectiveness of our proposed modules. The C-MAPSS dataset includes time sequential data from 21 sensors in total, which record the degradation process for aircraft engines as shown in Fig. 6. The information recorded from these sensors includes temperature, pressure, and speed measured by sensors distributed on different locations for the engine. There are four subsets: FD001, FD002, FD003 and FD004, and their details are listed in Table II. As listed in Table II, FD002 and FD004 are more complex than others. They contain more training and testing trajectories, and are involved in six different operation conditions.

TABLE I: Comparison with other methods.

Dataset	FD001		FD002		FD003		FD004	
Evaluation	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
HDNN [32]	13.02	245.00	15.24	1282.42	12.22	287.72	18.16	1527.42
CNN-LSTM [33]	14.40	290.00	27.23	9869.00	14.32	316.00	26.69	6594.00
BLCNN [34]	13.18	302.28	19.09	1557.55	13.75	381.37	20.97	3858.78
DCNN [5]	12.61	273.70	22.36	10412.00	12.64	284.10	23.31	12466.00
BiLSTM-ED [35]	57.00	273.00	49.00	3099.00	42.00	574.00	40.00	3202.00
LSTM-BS [36]	14.89	481.00	26.86	7982.00	15.11	493.00	27.11	5200.00
ATS2S [16]	12.63	243.00	14.65	876.00	**11.44**	263.00	16.66	1074.00
CNN-BiLSTM [8]	12.13	**174.00**	16.01	1230.00	11.96	242.00	18.10	1513.00
MF-LSTM [18]	12.15	233.12	**14.26**	747.97	12.57	254.11	17.63	1174.83
Bi-LSTM based Two-Stream Network (*ours*)	**11.96**	206.33	14.48	**726.82**	13.41	**223.36**	**15.11**	**892.39**

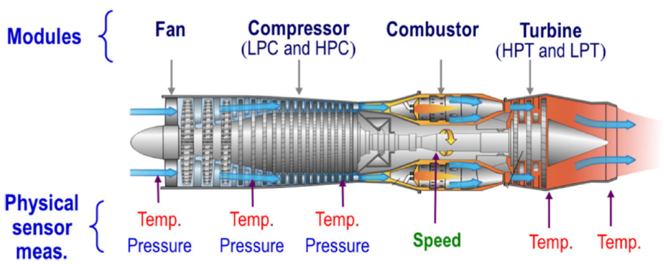


Fig. 6: Illustration for an aircraft engine.

TABLE II: Overview of C-MAPSS Dataset

Subsets	Conditions	Fault Modes	Training trajectories	Testing trajectories
FD001	1	1	100	100
FD002	6	1	260	259
FD003	1	2	100	100
FD004	6	2	248	249

1) Data Setting: Although there are 21 sensors in this dataset, some sensor data remains a constant value, providing less information for this task. Following the protocol in [8], [9], we remove the redundant sensor data during training and testing. There are 14 sensor data used in this paper.

Sliding window [5], [9], [37] is utilized to segment the sensor data, where the window size is set to be 30 in this paper. For the length of the data sequence less than 30, we simply pad the segmented data with its first value. Data normalization has been proven to be important in the RUL prediction. According to [8], the min-max normalization is used to normalize all sensor data. Following [38], we also employ the piece-wise linear for RUL values. Specifically, we set the maximal RUL value as 125. To improve the RUL prediction, we also concatenate the operation settings with the raw inputs signals together and forward them into the raw input signal stream. The network structure for each stream is set as the same. During the training process, the training data is divided into 5 folds. In each training epoch, 4 folds are selected from these 5 folds for training.

B. Network Hyper-parameters

The hidden number for the first and the second Bi-LSTM layer is 16 and 32, respectively. The hidden number for the two fully connected layers in each stream are set as 16. The hidden number of the fully connected layer in the fusion module is set as 8. For data argumentation, the dropout layer is used and the dropout rate is 0.2. Each fully connected layer is followed by a ReLU layer. In the training process, the batch size is 10 and the epoch number is 50. All experiments are conducted with PyTorch and the Ubuntu 20.04 system.

C. Evaluation Methods

According to approaches [8], [9], we choose two metrics, root mean square error (RMSE) and scoring function, to evaluate the performance of our proposed method.

1) RMSE. The RMSE is commonly used to compare the difference between the predicted value and the actual remaining useful life. It is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - g_i)^2}, \tag{12}$$

where N is the number of samples, r_i and g_i are the predicted RUL value and the true RUL value, respectively.

2) Scoring Function. The score function is proposed in the C-MAPSS dataset. It is defined based on the real industry situations. In practice, late predictions may cause more serious consequences than early predictions. In view of this, the scoring function gives a higher penalty on the late estimation than the early estimation. This scoring function is computed as below:

$$S = \begin{cases} \sum_{i=1}^{N}(e^{\frac{g_i - r_i}{13}} - 1), & r_i < g_i \\ \sum_{i=1}^{N}(e^{\frac{r_i - g_i}{10}} - 1), & r_i \geq g_i \end{cases}. \tag{13}$$

D. Comparison with Other Methods

In this subsection, we compare our Bi-LSTM based two-stream network with other state-of-the-art methods in Table I. Among our proposed fusion methods, the element-wise addition method achieves the best performances. We compare our Bi-LSTM based two-stream network which uses the addition fusion module, with other methods.

As listed in Table I, we compare our method with other state-of-the-art methods: HDNN [32], CNN-LSTM [33], BLCNN [34], DCNN [5], BiLSTM-ED [35], LSTM-BS [36], ATS2S [16], and CNN-BiLSTM [8] in four subsets of the C-MAPSS dataset. We re-implement our preliminary method

TABLE III: Ablation study for our proposed Bi-LSTM based two-steam network.

	FD001		FD002		FD003		FD004	
Method	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
Raw Stream (baseline)	12.58	228.07	**14.05**	1040.85	**12.81**	224.74	15.25	916.96
HFF Stream	13.65	298.39	14.36	879.24	15.32	339.53	18.78	1818.05
Late Fusion	13.14	229.19	14.54	797.33	14.06	223.95	18.13	1245.09
Concatenation Fusion	14.69	301.52	14.37	744.51	15.44	312.55	**14.49**	910.53
Addition Fusion	**11.96**	**206.33**	14.48	**726.82**	13.41	**223.36**	15.11	**892.39**

MF-LSTM [18] and improve the data pre-processing part, which further improves the performances of our MF-LSTM. Compared with other methods, our proposed Bi-LSTM based two-stream network surpasses other methods on most subsets, and sets new state-of-the-art performances. The CNN-BiLSTM [8] which is a hybrid method, performs good on the two simple subsets, i.e., FD001 and FD003, where our proposed method can achieve comparable performance. Besides, our proposed method significantly outperforms it on the two challenging and complex subsets, i.e., FD002 and FD004, in terms of both RMSE and score.

E. Ablation Study

To show the effectiveness of our proposed fusion methods, the ablation study is performed in this subsection. The experimental results are listed in Table III. Our proposed Bi-LSTM based two stream network is composed of two streams: raw stream and HFF stream. To show the effectiveness of our proposed fusion method, we firstly show each stream performances on four different subsets. Then, we list the three different fusion method performances.

As listed in Table III, our raw stream is treated as our baseline, which is composed of two consecutive Bi-LSTM layer and a fully connected layer. After that, another fully connected layer is used as a regressor to produce the RUL prediction. Benefited from our improved data pre-processing, our baseline Bi-LSTM shows a good RUL prediction accuracy on four subsets. FD001 and FD003 are relatively simple. Our baseline achieves a RMSE of 12.58 and a score of 228.07 on FD001, and a RMSE of 12.81 and a score of 224.74 on FD003. In FD002 and FD004 subsets, more conditions and fault modes are included, which increases the difficulty in the RUL prediction. In the FD002 subset, our baseline gets a RMSE of 14.05 and a score of 1040.85. In the FD004 subset, our baseline achieves a RMSE of 15.25 and a score of 916.96.

Our HFF stream only utilizes our proposed HFFs to predict the RUL. Since our HFFs are generated according to the aggregation across a sequence of signals, the signal values at the last time step are not available. This makes the HFF stream difficult to predict the RUL accurately. However, our HFF is able to suppress the noise existing in the raw signals, which enables it to achieve satisfactory performances and to even surpass our raw stream on the FD002 subset (879.24 Score v.s. 1040.85 Score).

To integrate our proposed HFFs with the raw input signals, three fusion methods are proposed by us. These fusion methods predict the RUL based on the two streams, and these fusion performances are affected by the RUL prediction accuracy of each stream. In this paper, the fusion method is expected to extract the useful information, filter out the wrong features, and achieve a better performance than each single stream. We will evaluate the effectiveness of our three fusion methods with the aspects discussed above.

According to Table III, our proposed late fusion method performs better than each single stream on FD001, FD002, and FD003. However, it performs inferior to our raw stream on the FD004. This may be because that our HFF stream performs worse than our raw stream on the FD004, which affects the late fusion method performances. Our late fusion method only combines both stream on the result level, which limits its fusion effectiveness. To solve this problem, we investigate another two feature level fusion methods: concatenation fusion and element-wise addition fusion.

As listed in Table III, after applying our concatenation fusion method, the performances on the FD002 and FD004 are improved. This indicates that the our concatenation fusion method can more effectively integrate our HFFs with the raw input signals by combining two streams on the feature level. However, the performances of our concatenation fusion method on the FD001 and FD003 are surpassed by our late fusion methods. The reason may be that the concatenation operation expands the feature dimension, and increases the training difficulty. Since the training trajectory number in FD001 and FD003 are smaller than that in FD002 and FD004, our concatenation fusion method is not fully trained. This limits our concatenation fusion performances. To mitigate this issue, we propose the element-wise addition fusion method. From Table III, it can be seen that our addition fusion method shows the best performances among four subsets. Especially for the FD004, our addition fusion method effective combines features from both streams and sets state-of-the-art performances on the C-MAPSS dataset.

F. RUL Prediction Result Analysis

To analyze the RUL accuracy on the C-MAPSS dataset in details, we visualize the detailed prediction results of our proposed method on the C-MAPSS dataset.

In Fig. 7, the prediction results for all test engines on four subsets are visualized. As shown in Fig. 7, the RUL prediction accuracy of our method increases when the true RUL decreases to zero. This character fulfills the industrial requirement in practice, that the RUL prediction system should accurately raise an alert when the industry tends to fall into failures. In the C-MAPSS dataset, compared with FD002 and FD004, FD001 and FD003 subsets involve fewer test trajectories. Our approach shows better performances on FD001 and FD003

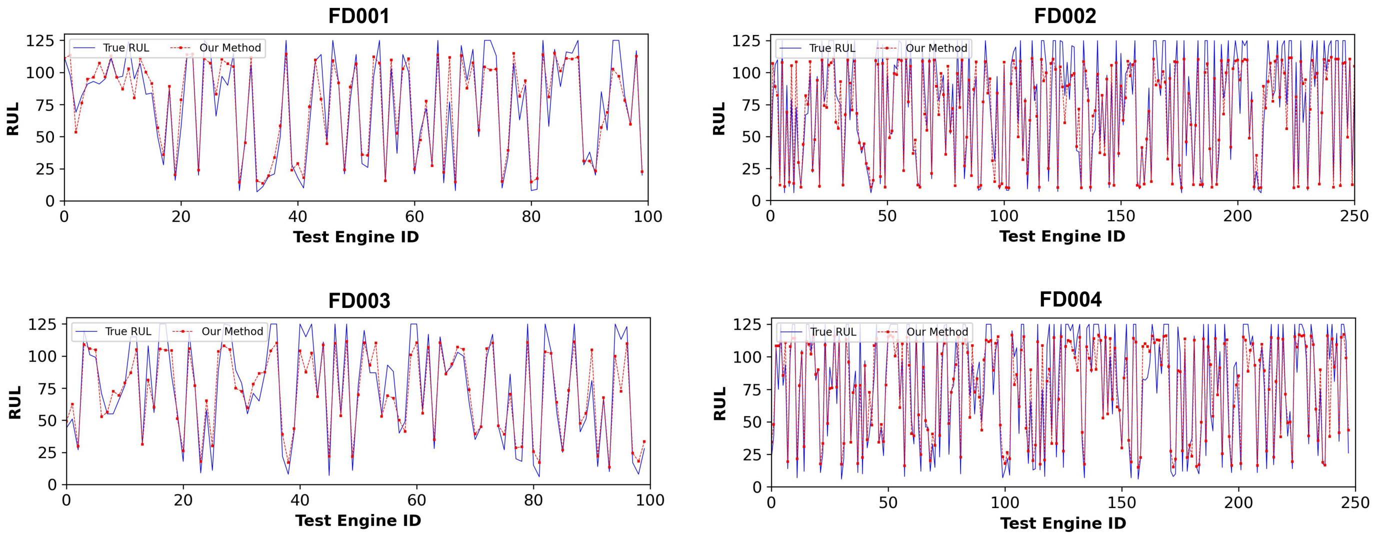


Fig. 7: Illustration for the RUL prediction results on test sets.

than FD002 and FD004. Since FD002 and FD004 include more complex scenarios, the RUL prediction accuracy is not stable. We will investigate to solve this problem by considering condition information in the future.

Apart from it, we also visualize several examples of life-time prediction results in Fig. 8. Among four subsets in the C-MAPSS, FD002 and FD004 are difficult and complex, which include different conditions. To show the effectiveness of our approach, we visualize two examples for the life-time predictions on these two subsets, respectively, in Fig. 8. In Fig. 8, it can be found that our proposed Bi-LSTM based two-stream network is able to predict the RUL accurately under various scenarios.

V. Conclusion and Future Work

This paper has proposed a series of new handcrafted feature flows (HFFs) and a novel Bi-LSTM based two-stream network for the RUL prediction. The proposed HFFs not only suppress the noise in the raw input signals, but also provides clear sequential information. To effectively integrate our HFFs, three different fusion methods are proposed. Through experiments, our proposed method is able to effectively improve the RUL prediction, and sets a new state-of-the-art performance compared with other benchmark methods on the C-MAPSS dataset.

In this paper, our handcrafted features have been shown to be effective in the RUL prediction. In future works, more effective handcrafted features will be investigated for further improving the performance of RUL prediction. Besides, the CNN-LSTM based approaches [8] have shown better feature representation capability than the LSTM based methods on the RUL prediction task. Thus, we will try to integrate the CNN-LSTM based neural network with our proposed two-stream network architecture to boost the RUL prediction performance.

References

[1] M. Pecht and J. Gu, "Physics-of-failure-based prognostics for electronic products," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 309–322, 2009.

[2] Y. Qian, R. Yan, and R. X. Gao, "A multi-time scale approach to remaining useful life prediction in rolling bearing," *Mechanical Systems and Signal Processing*, vol. 83, pp. 549–567, 2017.

[3] Q. Zhai and Z.-S. Ye, "Rul prediction of deteriorating products using an adaptive wiener process model," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2911–2921, 2017.

[4] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*. Springer, 2016, pp. 214–228.

[5] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[6] B. Yang, R. Liu, and E. Zio, "Remaining useful life prediction based on a double-convolutional neural network architecture," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9521–9530, 2019.

[7] C. Zhu, Z. Chen, R. Zhao, J. Wang, and R. Yan, "Decoupled feature-temporal cnn: Explaining deep learning-based machine health monitoring," *IEEE Transactions on Instrumentation and Measurement*, 2021.

[8] J. W. Song, Y. I. Park, J.-J. Hong, S.-G. Kim, and S.-J. Kang, "Attention-based bidirectional lstm-cnn model for remaining useful life estimation," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[9] 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.

[10] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, 2017, pp. 88–95.

[11] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. N. Low, and S. S. Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," in *2019 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2019, pp. 915–920.

[12] W. Mao, J. He, and M. J. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1594–1608, 2020.

[13] Y. Ding, M. Jia, and Y. Cao, "Remaining useful life estimation under multiple operating conditions via deep subdomain adaptation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

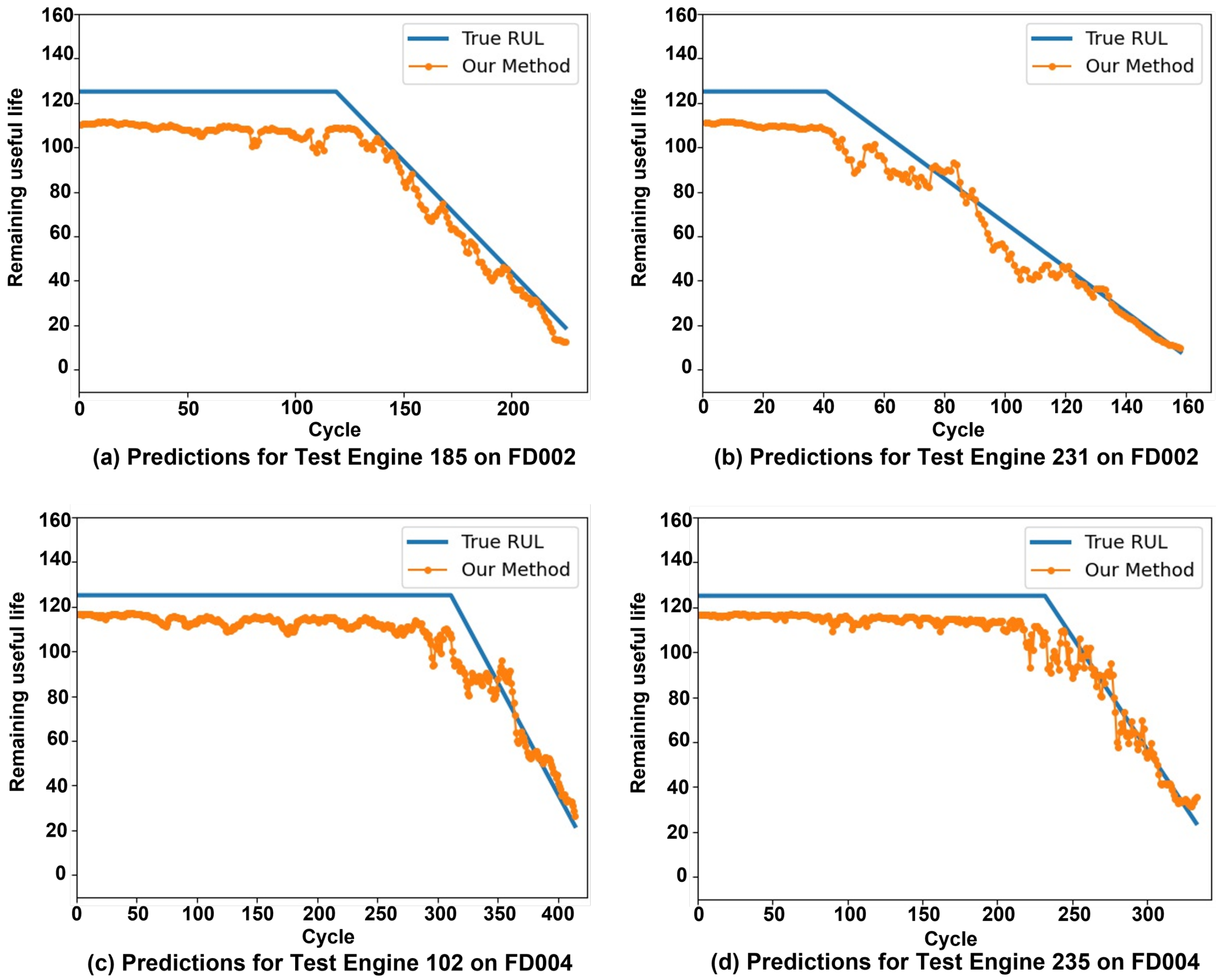


Fig. 8: Illustration for the time-life RUL prediction results on four test engines.

[14] J.-Y. Wu, M. Wu, Z. Chen, X.-L. Li, and R. Yan, "Degradation-aware remaining useful life prediction with lstm autoencoder," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[15] Z. Que, X. Jin, and Z. Xu, "Remaining useful life prediction for bearings based on a gated recurrent unit," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[16] M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, R. Yan, and X. Li, "Attention-based sequence to sequence model for machine remaining useful life prediction," *Neurocomputing*, vol. 466, pp. 58–68, 2021.

[17] 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.

[18] R. Jin, Z. Chen, K. Wu, M. Wu, X. Li, and R. Yan, "Multi-feature fused bidirectional long short-term memory for remaining useful life prediction," in *the second International Conference on Sensing, Measurement and Data Analytics in the era of Artificial Intelligence (ICSMD 2021)*. IEEE, 2021.

[19] 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.

[20] 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.

[21] L. Ren, H. Qin, Z. Xie, B. Li, and K. Xu, "Aero-engine remaining useful life estimation based on multi-head networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.

[22] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bidirectional lstm prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.

[23] Y. Qin, D. Chen, S. Xiang, and C. Zhu, "Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings," *IEEE Transactions on Industrial Informatics*, 2020.

[24] Y. Duan, H. Li, M. He, and D. Zhao, "A bigru autoencoder remaining useful life prediction scheme with attention mechanism and skip connection," *IEEE Sensors Journal*, vol. 21, no. 9, pp. 10 905–10 914, 2021.

[25] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. ICML, 2020, pp. 1597–1607.

[27] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[28] M. Ragab, Z. Chen, M. Wu, C. S. Foo, C. K. Kwoh, R. Yan, and X. Li, "Contrastive adversarial domain adaptation for machine remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5239–5249, 2021.

[29] J. Jang and C. O. Kim, "Siamese network-based health representation learning and robust reference-based remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by

learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[31] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *Journal of manufacturing systems*, vol. 48, pp. 78–86, 2018.

[32] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Computers in Industry*, vol. 108, pp. 186–196, 2019.

[33] Z. Wu, S. Yu, X. Zhu, Y. Ji, and M. Pecht, "A weighted deep domain adaptation method for industrial fault prognostics according to prior distribution of complex working conditions," *IEEE Access*, vol. 7, pp. 139 802–139 814, 2019.

[34] H. Liu, Z. Liu, W. Jia, and X. Lin, "A novel deep learning-based encoder-decoder model for remaining useful life prediction," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[35] W. Yu, I. Y. Kim, and C. Mechefske, "Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme," *Mechanical Systems and Signal Processing*, vol. 129, pp. 764–780, 2019.

[36] Y. Liao, L. Zhang, and C. Liu, "Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method," in *2018 ieee international conference on prognostics and health management (icphm)*. IEEE, 2018, pp. 1–8.

[37] 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.

[38] Q. Xu, Z. Chen, K. Wu, C. Wang, M. Wu, and X. Li, "Kdnet-rul: A knowledge distillation framework to compress deep neural networks for machine remaining useful life prediction," *IEEE Transactions on Industrial Electronics*, 2021.