

Dual Aspect Self-Attention based on Transformer for Remaining Useful Life Prediction

Zhizheng Zhang, Wen Song, Qiqiang Li

Abstract—Remaining useful life prediction (RUL) is one of the key technologies of condition-based maintenance, which is important to maintain the reliability and safety of industrial equipments. While deep learning has achieved great success in RUL prediction, existing methods have difficulties in processing long sequences and extracting information from the sensor and time step aspects. In this paper, we propose Dual Aspect Self-attention based on Transformer (DAST), a novel deep RUL prediction method. DAST consists of two encoders, which work in parallel to simultaneously extract features of different sensors and time steps. Solely based on self-attention, the DAST encoders are more effective in processing long data sequences, and are capable of adaptively learning to focus on more important parts of input. Moreover, the parallel feature extraction design avoids mutual influence of information from two aspects. Experiments on two widely used turbofan engines datasets show that our method significantly outperforms the state-of-the-art RUL prediction methods.

Index Terms—Remaining useful life, Transformer, Dual aspect self-attention, Feature fusion.

I. INTRODUCTION

MAINTENANCE management plays a very important role in the operation of modern large mechanical equipment [1]. Compared with the traditional preventive maintenance strategy, condition-based maintenance (CBM) is more effective in reality due to the utilization of real-time system health information, hence is widely used in the maintenance of modern industrial equipment [2]. CBM involves predicting the remaining useful life (RUL) and potential faults of the equipment according to the real-time operational status, based on which the maintenance decisions can be made on an as-needed basis according to the prediction information. Obviously, RUL prediction is one of the most critical technologies for effective implementation of CBM. If the RUL of mechanical equipment is predicted according to the current or historical operation information, the time of failure can be accurately known [3]. Therefore, RUL prediction is of great importance to researchers in the field of CBM.

In general, RUL prediction methods can be roughly divided into traditional model-based methods, data-driven methods and

hybrid methods. The model-based method requires accurate dynamic modeling of mechanical equipment or components to describe the degradation trend of components [4]. However, the structure of modern industrial large-scale equipment is becoming more and more complex, with miscellaneous nonlinear relationships between various systems and parts. Therefore, it is unrealistic to establish an accurate model.

The goal of data-driven RUL prediction method is to establish the mapping relationship between RUL and features of the target equipment [5]. It does not require extensive expert knowledge and physical modeling for complex mechanical equipment [6]. In the literature, some traditional machine learning algorithms have been used for RUL prediction, such as support vector regression (SVR) [7], random forest (RF) [1], and extreme learning machine (ELM) [8]. However, these methods rely on tedious feature engineering. In contrast, deep learning based methods can automatically extract valuable features from the original CBM data, and achieve much better prediction performance. Consequently, deep learning based RUL prediction methods have a wider range of applicability and have received increasing attention recently [9].

The RUL prediction of mechanical equipment is essentially a multivariate time series regression task. The strong temporal and spatial correlation in the condition monitoring signals can be effectively captured by modern deep architectures such as recurrent neural network (RNN) [10] and convolution neural network (CNN) [11], which have been widely used in RUL prediction. RNN based methods employ sequence components such as long short-term memory (LSTM) [12] and gated recurrent unit (GRU) [13] to analyze the signal data sequences. However, due to the existence of recurrent structures in RNN, sequence data needs to pass through each processing unit in turn to extract useful features, which inevitably causes the problem of forgetting important information and is less effective in learning long-term dependencies. CNN based methods normally apply one-dimensional convolution and pooling filters along the time dimension over all sensors to extract feature information [14]. However, when processing long time sequences, CNN based methods need to continuously increase the size of convolution kernels to obtain larger time step receptive field which contain more sequence information. That is, the ability of CNN based methods in capturing long-term dependent information in sequence data is also limited.

For RUL prediction, another key issue is that more attention should be paid to the important features that contain more degradation information. Attention mechanism [15] is an effective method to learn such dependencies, i.e. weights among

Zhizheng Zhang and Wen Song are with Institute of Marine Science and Technology, Shandong University, China (Email: 202020861@mail.sdu.edu.cn, wensong@email.sdu.edu.cn).

Qiqiang Li is with School of Control Science and Engineering, Shandong University, China (Email: qqli@sdu.edu.cn).

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

different time steps and sensors. Recently, several works attempt to combine the attention mechanism with RNN/CNN based structure to predict RUL [16], [17]. However, there are two major shortcomings in these methods. Firstly, the inefficiency of RNN/CNN in capturing long-term dependencies still can not be avoided. Secondly, the input data enters the attention modules and RNN/CNN modules sequentially, which causes the mutual influence between the extracted feature information, thereby affecting the RUL prediction performance.

Transformer [18] is a recently proposed sequence modeling architecture. It makes use of the self-attention mechanism to capture the long-term dependencies between elements in a sequence without considering their distance, so that it is less affected by the increase of sequence length compared with traditional methods such as RNN and LSTM. However, for RUL prediction, the vanilla transformer architecture only attention the weights of different time steps and ignores the importance of different sensors in the CBM data stream, which is crucial for the overall prediction performance.

To overcome the above issues, in this paper, we propose a novel deep RUL prediction method named Dual Aspect Self-attention based on Transformer (DAST). In this method, we apply the Transformer architecture [18] to RUL prediction for the first time, which is an encoder-decoder structure purely based self-attention, without any RNN/CNN module. DAST consists of two encoders, i.e. the sensor encoder and time step encoder. Each of the two DAST encoders employs the self-attention mechanism to process all CBM sequence data, and automatically learns to pay different attentions to different sensors and time steps. Specifically, the two decoders work *in parallel* in the process of feature extraction, therefore the mutual influence of the two aspect information is avoided. The features extracted by the two encoders are fused together and fed into the self-attention based decoder to obtain the RUL prediction. The main contributions of this paper are summarized as follows:

- 1) We propose a novel end-to-end deep RUL prediction method based on the Transformer architecture. Experiments on two widely used NASA's turbofan engine datasets show that our method significantly outperforms the state-of-the-art RUL prediction methods.
- 2) Based on self-attention, our method is able to automatically pay more attentions to the features that are more important without any domain knowledge, and is more effective in handling long CBM data sequences than RNN/CNN based methods. Our novel dual-aspect design enables extracting features from the sensor and time step dimensions simultaneously, which overcomes the limitation of vanilla Transformer and effectively improves the RUL prediction performance.
- 3) The weights of different sensors and time steps learned by our method is intuitive and interpretable to the maintenance personnel, so that they can formulate better maintenance strategies to improve maintenance efficiency.

The rest of this paper is organized as follows. The second section introduces the related literature review. The third section introduces the proposed method. In addition, the fourth

part also demonstrates the effectiveness and superiority of this method. Finally, the fifth part summarizes the paper.

II. RELATED WORKS

In this section, we briefly review recent data-driven methods for RUL prediction.

Due to its advantages in processing sequence data, RNN architecture and its variants (e.g. LSTM and GRU) have been widely used in RUL prediction. Zheng et al. [12] used the LSTM network prediction model to verify the effectiveness of LSTM compared to RNN. Chen et al. [13] proposed a GRU based method for predicting nonlinear deterioration process. Besides RNN, CNN architecture has also been applied to the RUL prediction task. Li et al. [14] proposed a RUL prediction method based on deep convolution neural network (DCNN), which directly uses normalized raw data as input and performs convolution operation along the time dimension. Zhu et al. [19] proposed the multi-scale convolutional neural network (MSCNN) for RUL prediction, which keeps the global and local information synchronously compared to traditional CNN.

Recently, there are several works employ the attention mechanism to improve the performance of RNN/CNN based models. For example, Liu et al. [16] proposed a RUL prediction method based on the combination of attention mechanism, GRU and CNN. The features extracted by the attention mechanism are fed into the bidirectional GRU and CNN network to predict RUL. Xiang et al. [17] proposed a gear RUL prediction method based on LSTM and attention mechanism (LSTM-AON). Experimental results prove that the attention mechanism can effectively improve the accuracy of RUL prediction. However, in most of these studies, attention mechanism is used in combination with RNN/CNN architecture, hence the limitation in processing long sequences still exists.

Owing to its effectiveness in modeling long sequences, Transformer has been employed in time-series related tasks recently. Zhou et al. [20] studied the application of Transformer in long sequence time-series prediction and proposed the ProbSparse self-attention mechanism to reduce the time complexity and memory usage. Beltagy et al. [21] proposed the Longformer with an attention mechanism that scales linearly with sequence length, making it easy to process long sequence. A recent survey of transformer variants can be found in [22]. However, most of existing studies only consider capturing the dependencies between time steps. In the RUL prediction, the weight information between different sensors also has a great influence on the final RUL prediction. Therefore, the above-mentioned research is not suitable for RUL prediction. At present, there are few studies on the application of the Transformer architecture to RUL prediction. In this paper, we explore this direction and propose a dual aspect self-attention design to capturing the weight information of different time steps and sensors at the same time, which makes it better to the RUL prediction task.

III. METHODOLOGY

In this section, we first describe the RUL prediction problem, and then present the proposed DAST method in detail, including its architecture and key components.

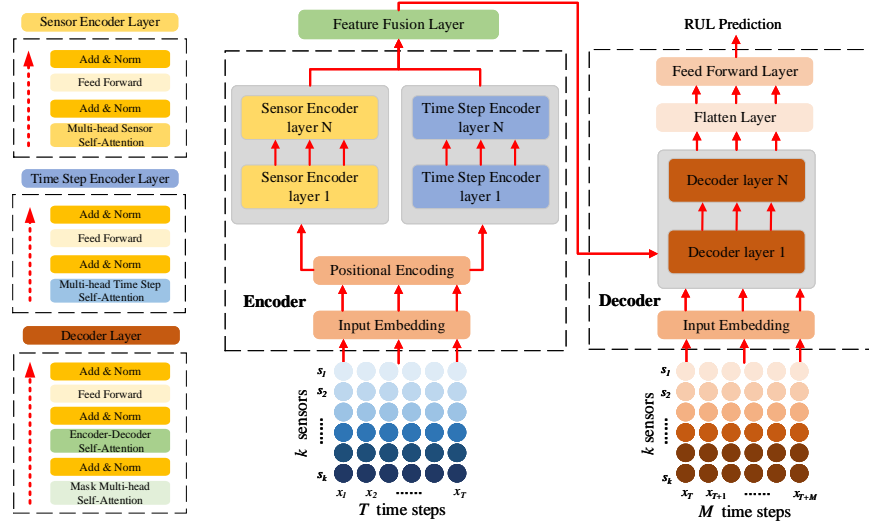


Fig. 1. Architecture of DAST.

A. Problem Description

In this paper, the CBM data collected by sensors during the process of turbofan engine operation are used. The RUL prediction problem can be formally defined as follows. The input is $X_t \in \mathbb{R}^k$, $t = (1, 2, \dots, T)$, where T is the length of time steps and k is the number of sensors. The corresponding output is the predicted RUL Y_t for each time step. Our purpose is to predict real-time RUL by establishing the mapping relationship between RUL and CBM data, expressed as follows:

$$Y_t = f(X_t), \quad (1)$$

where X_t is the real-time CBM data during operation of the turbofan engine, f is the mapping function, and Y_t is the real-time RUL predicted by f . In this paper, we design a Transformer based deep architecture to establish the mapping, which will be detailed in the following subsections.

B. Model Architecture

Fig. 1 shows the architecture of our RUL prediction method, Dual Aspect Self-attention based on Transformer (DAST). DAST follows the encoder-decoder structure in the original Transformer, and consists of three main substructures in the framework: encoder layer (including sensor encoder layer and time step encoder layer), feature fusion layer and decoder layer. Different from the RUL prediction method based on RNN and CNN architecture, DAST captures the long-term dependence information between the inputs and outputs of sequence through self-attention mechanism without considering the distance, so that the importance of each work cycle information will not be reduced due to the increase of time step length. Based on the Transformer architecture, we propose a novel feature extract and fusion approach that is more suitable for the RUL prediction task. It enables learning the weights of different sensors and time steps at the same time, and obtain more valuable feature information of turbofan engine by fusing the feature information of the two parts.

The workflow of our DAST model is as follows. First, it performs feature extraction on CBM data collected by multiple sensors. Specifically, we design a dual aspect encoding mechanism, which applies the sensor encoder and time step encoder that work in parallel to capture the weight features of different sensors and time steps. Both encoders are designed based on multi-head self-attention mechanism. Second, the features extracted from the two aspects are integrated by the feature fusion layer to get a new feature map with importance information of different sensors and time steps. Finally, the fused feature map is sent to the decoder, which adopts the self-attention mechanism to realize the attention of current work cycle information and the previous different time steps and sensors information and outputs the predicted RUL through a fully connected feed-forward network (FFN). Next, we will discuss each of the above mentioned substructures in detail.

C. Encoder of DAST

The encoder of DAST is mainly composed of an input embedding layer, a positional encoding layer, and multiple sensor encoder layers and time step encoder layers. The input embedding layer maps the input state monitoring data to a vector of D_{model} dimension through a FFN, in order to prepare for the following feature extraction process. The remaining components of encoder is described below.

1) Positional encoding layer

As mentioned, our DAST model does not contain structure based on RNN or CNN. Consequently, we need to inject some relative position tokens into the sequence, so that the model can make full use of the position information of the sequence. There are currently a variety of positional encoding methods to choose. In this paper, we use sine and cosine functions of different frequencies [18]:

$$P_t(2k) = \sin(t/10000^{2k/D_{\text{model}}}) \quad (2)$$

$$P_t(2k+1) = \cos(t/10000^{2k/D_{\text{model}}}) \quad (3)$$

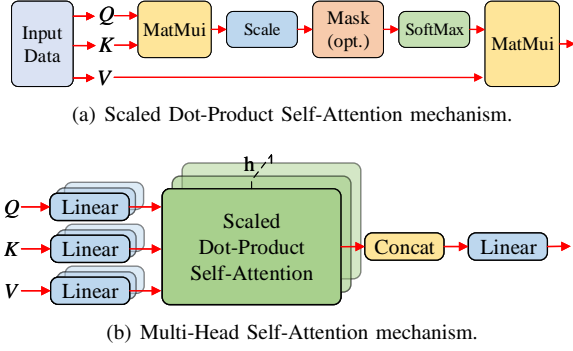


Fig. 2. The process of Multi-Head Self-Attention.

where t is the time step and k is the sensor dimension. In this way, P_t has a linear relationship with P_{t+l} , where l is any fixed time step. This allows the model to easily learn to attend according to the relative positions.

2) Sensor encoder layer

A sensor encoder layer mainly includes two sub-layers: a multi-head sensor self-attention layer and a FFN layer. As shown in Fig.1, there is a residual connection and layer normalization (Add & Norm) after each sub-layer. The purpose of residual connection is to alleviate the difficulty of training deep neural network. Layer normalization can accelerate the training process and make the model converge faster by normalizing the layer activation value. Sensor encoder layer uses the multi-head self-attention mechanism [18] to extract the importance of different sensors along the sensor dimension, therefore it can automatically learn to focus on those sensor features with higher weights without human experience intervention in the training process. Next, we will introduce the working process of the multi-head sensor self-attention.

We define the CBM data collected by the k sensors in a time window of length T as $X = \{X_1, X_2, \dots, X_T\} \in \mathbb{R}^{d_k \times T}$. We also define X'_s as the data obtained after being processed by the positional encoding layer. The working process of self-attention function is visualized in Fig. 2. First, it generates three matrices (Queries, Keys, Values) by processing the input data using the following computation:

$$Q_s = X'_s W_s^q, K_s = X'_s W_s^k, V_s = X'_s W_s^v, \quad (4)$$

where W_s^q, W_s^k, W_s^v are trainable parameters, $Q_s, K_s, V_s \in \mathbb{R}^{d_k \times D_{\text{model}}}$, D_{model} is the input dimension. Then we calculate the dot product of Q and K (scaled by $\sqrt{D_{\text{model}}}$), and apply a softmax function along the sensor dimension to obtain the weights of different sensors in X'_s . Therefore, the weight vector of different sensors at time step t is:

$$\alpha_t = \text{softmax}_{\text{sensors}} \left(\frac{Q_s K_s^T}{\sqrt{D_{\text{model}}}} \right) \quad (5)$$

where $\alpha_t = (\alpha_{t,1}, \alpha_{t,2}, \dots, \alpha_{t,k})$, $t = (1, 2, \dots, T)$. Finally, the features of different sensors weighted by the self-attention mechanism is computed as a weighted sum of V_s :

$$\text{Attention}_{\text{sensors}}(Q_s, K_s, V_s) = \alpha_t V_s. \quad (6)$$

Here we also adopt the multi-head self-attention mechanism in [18] to allow the model to jointly attend to information from

different representation subspaces at different positions, such that to improve the prediction performance. Fig. 2 visualizes multi-head self-attention, which can be expressed as:

$$\text{MultiHead}(Q_s, K_s, V_s) = \text{Concat}(\{head_i\}_{i=1}^h) W^s, \quad (7)$$

where the parameter matrices $W^s \in \mathbb{R}^{h D_{\text{model}} \times D_{\text{model}}}$, h is the number of heads, and $head_i = \text{Attention}(Q_s, K_s, V_s)_i$.

3) Time step encoder layer

As shown in Fig. 1, the time step encoder layer has the same structure as the sensor encoder layer. It mainly includes two sub-layers: a multi-head time step self-attention layer and a FFN layer. The difference is that the time step encoder layer extracts features along the time step dimension, and allows the DAST model to pay attention to the time steps that are more important to the RUL prediction. The input data of time step encoder is the transpose of X and being processed by the positional encoding layer as X'_t . The time step encoder layer first obtains the Queries, Keys, and Values matrices $Q_t, K_t, V_t \in \mathbb{R}^{T \times D_{\text{model}}}$ by Eq. (4) using trainable parameters W_t^q, W_t^k, W_t^v . Then, the weight vector of different time steps corresponding to the sensor s can be obtained by performing softmax along on the time step dimension:

$$\beta_s = \text{softmax}_{\text{time_steps}} \left(\frac{Q_t K_t^T}{\sqrt{D_{\text{model}}}} \right) \quad (8)$$

where $\beta_s = (\beta_{s,1}, \beta_{s,2}, \dots, \beta_{s,T})$, $s = (1, 2, \dots, k)$. The features of different time steps weighted by the self-attention mechanism is obtained as:

$$\text{Attention}_{\text{time_steps}}(Q_t, K_t, V_t) = \beta_s V_t. \quad (9)$$

Similar to the sensor encoder layer, we also apply multi-head self-attention to the time step encoder layer:

$$\text{MultiHead}(Q_t, K_t, V_t) = \text{Concat}(\{head_j\}_{j=1}^h) W^t, \quad (10)$$

where the parameter matrices $W^t \in \mathbb{R}^{h D_{\text{model}} \times D_{\text{model}}}$, and $head_j = \text{Attention}(Q_t, K_t, V_t)_j$.

In DAST, both the sensor encoder and time step encoder are constructed by stacking multiple identical sensor or time step encoder layers. Here for convenience, we use the same number of stacks N for both encoders, but in general they could be different. The hyperparameter N could affect the feature extraction capability of DAST. We empirically tune N in the experiments to obtain good performance. We would like to note that unlike previous works [16], [17], in DAST, features of the sensor dimension and time step dimension are extracted simultaneously, since the time step encoder and sensor encoder are arranged in parallel. This design effectively avoids the mutual influence of information from the two aspects, which helps to improve the performance of RUL prediction. We will show the advantage of our design in the experiments.

D. Feature Fusion Layer

After extracting features from the time step and sensor dimension of CBM data, DAST performs feature fusion to integrate information from the two aspects. As shown in Fig. 3, the feature fusion layer combines the time step and sensor features to form a new feature map. We denote the features

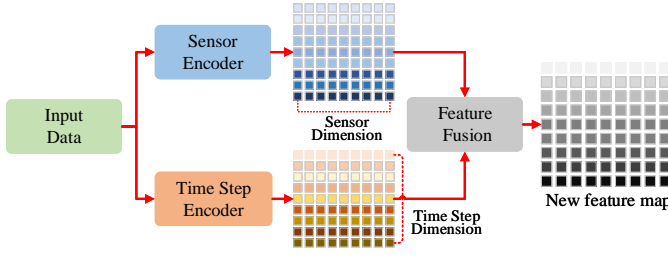


Fig. 3. The process of feature fusion.

extracted from the sensor encoder and time step encoder as $F_s \in \mathbb{R}^{d_k \times D_{\text{model}}}$ and $F_t \in \mathbb{R}^{T \times D_{\text{model}}}$, respectively. DAST performs feature fusion using the following computation:

$$F_r = \text{Concat}(F_s, F_t) W^f, \quad (11)$$

where the trainable parameter matrices $W^f \in \mathbb{R}^{(d_k+T) \times D_{\text{model}}}$, which can make the model to capture feature information from both F_s and F_t .

E. Decoder of DAST

The decoder of DAST is designed in a similar way as in the original Transformer [18]. As shown in Fig. 1, the decoder consists of an input embedding layer, multiple identical decoder layers, a flatten Layer and a FFN layer. A decoder layer mainly includes two multi-head self-attention sub-layers, including: mask multi-head self-attention and encoder-decoder multi-head self-attention sublayer. The encoder-decoder multi-head self-attention will receive Keys and Values from the output of the encoder, while Queries are from the output of the previous layer of the decoder. Therefore, the weight features of different sensors and time steps extracted in the encoder part are analyzed on the decoder, which realizes the attention of the current working cycle information and the previous different time steps and sensors information and finally outputs the predicted RUL through the FFN layer. In order to ensure that prediction of a time sequence data point will only depend on previous data points, the masked multi-head self-attention is applied in the self-attention computed by setting the corresponding dot products to $-\infty$. Such mask mechanism can ensure that model only apply attention to the data points before the target data. In other words, the self-attention mechanism will only attention on data x_{T-1} and previous data when we predict the RUL of x_T . In this paper, we follow previous works in RUL prediction and apply the rolling prediction.

IV. EXPERIMENTS

In this section, we conduct experiments to evaluate the proposed DAST model on two widely used turbofan engines datasets. All experiments are performed on a windows 10 workstation, which is equipped with 64GB RAM and an Intel 9900K CPU. For training, we use the rectified Adam optimizer and set the epoch to 100. We use the root mean square error (RMSE) as the training loss function. The learning rate is set to 0.001 and the batch size is set to 256. We apply dropout for each encoder and decoder layer, and the dropout rate is set to 0.2. Our code in Pytorch will be released soon.

TABLE I
DESCRIPTION OF THE DATASETS

Dataset	C-MAPSS				PHM-2008
	FD001	FD002	FD003	FD004	
Training engines	100	260	100	249	218
Testing engines	100	259	100	248	218
Operating conditions	1	6	1	6	6
Fault modes	1	1	2	2	2
Training set size	20631	53759	24720	61249	45918
Test set size	100	259	100	248	218

A. Datasets

In this paper, we mainly adopt the widely used C-MAPSS (Commercial Modular Aero Propulsion System Simulation) dataset [23] to evaluate our method. The C-MAPSS dataset contains four different sub-datasets. As shown in Table I, FD001, FD002, FD003, and FD004 have different numbers of operating conditions and fault modes. Among the 21 sensors in the C-MAPSS dataset (indexed from 1 to 21), sensors 1, 5, 6, 10, 16, 18 and 19 always have constant values during the run-to-failure experiments, meaning that data from these sensors cannot characterize the degradation process of the engine. Therefore, we remove these sensor data series and use the data of the remaining 14 sensors for RUL prediction. Another problem we considered is that in actual situation, the degradation process of turbofan engine in the early stage can be ignored, that is to say, the engine RUL should be kept constant in the early stage. Therefore, we follow Zheng et al. [12] and limit the engine RUL from start to degradation to RUL_{max} , and the linear degradation of turbofan engine occurs after RUL_{max} . In this work, RUL_{max} is set to 125. Besides C-MAPSS, we also test our method on the PHM 2008 dataset [24]. This dataset was widely used for RUL evaluations, which has the same data structure with C-MAPSS dataset. The details of training and test set are shown in Table I.

B. Data Preprocessing

- 1) Normalization [25]. Data from different sensors have various units and scales, which could affect the accuracy of RUL prediction. Hence, we use the min-max scaler method to normalize sensor data. Specifically, for the CBM data $X_i = \{X_1, X_2, \dots, X_T\}$, we normalize it as follows:

$$\tilde{X}_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)}, \quad (12)$$

where \tilde{X}_i is the normalized data, $\max(X_i)$ and $\min(X_i)$ denote the maximum and minimum of X_i .

- 2) Sliding window processing. A sliding window is often used for data segmentation preprocessing in order to make the model get valuable information from multivariate time series as much as possible. A simple example of sliding time window processing is shown in Fig. 4. T_w is the size of time window and the sliding stride is set to one. The RUL of the last data point in a time window serves as the RUL of that window. We will discuss the impact of time window size on the model prediction performance later.
- 3) The mean value and regression coefficient estimates [26] of the sequence data are two explicit numerical features

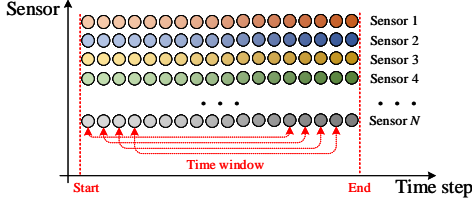


Fig. 4. Sliding window processing.

TABLE II
STRUCTURAL PARAMETERS OF DAST

Components	Layers	Parameters
Input embedding	Fully connected layer	Hidden units: 64 Activation: Linear
Encoder	Sensor encoder layer	Encoder blocks $N = 2$ Self-attention heads $H = 4$
	Time step encoder layer	Encoder blocks $N = 2$ Self-attention heads $H = 4$
Decoder	Decoder layer	Decoder blocks $N = 1$ Self-attention heads $H = 4$
Output	Fully connected layer	Hidden units: 64 Activation: (ReLU)
	Output layer	Hidden units: 1 Activation: Linear

often used in time sequence data, which can provide useful sequence statistics. In this paper, we add these two parts of feature information to the sequence.

C. Evaluation Metrics

In this paper, we use two commonly adopted performance indicators to verify the effectiveness of our method. One is the well-known RMSE metric, and the other is the Score metric [23] defined as follows:

$$\text{Score} = \begin{cases} \sum_{i=1}^n e^{-\frac{y'_i - y_i}{13}} - 1, & \text{for } y'_i - y_i < 0 \\ \sum_{i=1}^n e^{\frac{y'_i - y_i}{10}} - 1, & \text{for } y'_i - y_i \geq 0 \end{cases} \quad (13)$$

where n is the number of testing samples, y'_i is the predicted RUL and y_i is the true RUL for the i th sample. Compared to RMSE, the Score metric punishes more when the predicted RUL is larger than the true RUL. This is reasonable because in practice, such “optimistic” prediction will cause more serious impact. For both RMSE and Score, the lower the value, the better the prediction accuracy.

D. Structural Parameters of DAST

To determine the structural parameters of our DAST model, we perform grid search to find the best model configuration. The resulting structural parameters are listed in Table II.

E. Experimental Results and Analysis

To mitigate the impact of randomness, we repeat all prediction experiments 10 times, and report the mean performances throughout the paper. Below we provide evaluation of the key factors and design of our method, as well as comparison with state-of-the-art methods.

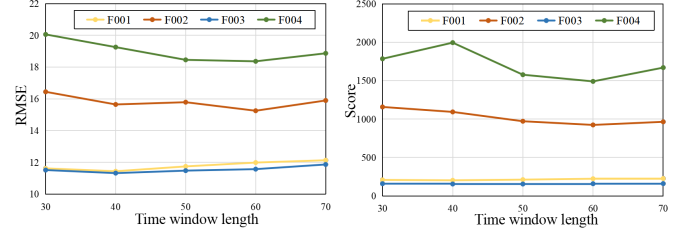


Fig. 5. Performance of DAST with different time window sizes on the C-MAPSS dataset.

TABLE III
ABLATION STUDY ON F002

Method	Metric	Mean	STD
DAST w/o sensor encoder (vanilla Transformer)	RMSE	16.47	0.39
	Score	1638.36	132.64
DAST w/o time step encoder	RMSE	16.11	0.35
	Score	1476.73	119.82
DAST with sensor and time step encoder arranged in series	RMSE	15.86	0.21
	Score	1367.61	49.67
DAST	RMSE	15.25	0.24
	Score	924.96	47.67

1) *Impact of the sliding time windows size*: The multi-variate time series data contains data with different RUL information, so it is necessary to choose a reasonable time window size before training the model. To verify the influence of this parameter, we have carried out several groups of comparative experiments using the four sub-datasets in C-MAPSS, by setting T_w to 30, 40, ..., 70. The results are plotted in Fig. 5. We can observe from Fig. 5 that the RMSE and Score values are the smallest when the time window length is 40 on the F001 and F003 datasets, while for F002 and F004 the two metrics are the best when $T_w = 60$. This is because the F002 and F004 datasets have more complex operating conditions and failure modes than the F001 and F003 datasets, so larger time window sizes could contain more degradation information. In the following experiments, we use the above-mentioned optimal time window sizes.

2) *Ablation study of DAST*: Here we evaluate the effectiveness of some internal components of our method using ablation study. More specifically, we evaluate the two key components, i.e. the sensor encoder and time step encoder, as well as the parallel feature extraction design. To this end, we perform three experiments in this part: DAST without sensor encoder, DAST without time step encoder, and DAST with sensor encoder and time step encoder arranged in series. Note that DAST without sensor encoder is essentially the vanilla Transformer. For the third experiment, the input data first pass through the sensor encoder, the output of which is treated as input to the time step encoder. We take the F002 dataset as an example, and the experimental results are shown in TABLE III. The experimental results show that the prediction performance based DAST is superior than the original Transformer. Moreover, when any of the sensor or time step encoder is removed, the performance significantly drops, showing that information from both aspects are valuable for RUL prediction. DAST also outperforms the version that

extracts features in series, which verifies the effectiveness of our parallel design in alleviating the mutual influence between information from the two aspects.

3) *Analysis of prediction results:* To visually analyze the RUL predicted by our DAST model, we randomly select an engine unit from each of four C-MAPSS datasets, and compare the predicted RUL with the actual RUL of engine. We also compare two representative RNN and CNN based methods, i.e. BiLSTM [27] and DCNN [14]. All results are plotted in Fig. 6. We can see that the prediction RUL of our method has a similar trajectory to the real RUL, showing the effectiveness of DAST in capturing degradation information. Moreover, the RUL values predicted by DAST are clearly better than those of BiLSTM and DCNN. Compared to the later stages, the prediction error tends to be larger in the early stages. This is because when the engine begins to enter the degradation stage, the CBM data contains more degradation information, which makes the later predictions more precise [16]. We can also see that compared with the other two datasets, F002 and F004 are more difficult to predict, since they contain more complicated operating conditions and fault modes.

4) *Comparisons with state-of-the-art methods:* Next, we compare the proposed method with recent published state-of-the-art RUL prediction methods. As shown in Table IV and Table V, our method consistently outperforms all comparing ones on the C-MAPSS dataset with the smallest value in both RMSE and Score, showing that DAST can produce more accurate RUL prediction. More importantly, the improvement of DAST against the best results of existing methods tends to be larger on the two harder datasets F002 and F004. Specifically, the corresponding RMSE of DAST is reduced by 21.51% and 8.60% and the Score is reduced by 38.06% and 28.94% respectively compared with state of the art. Another key observation is that, the improvement of DAST is much more prominent in terms of Score, showing its advantage in this practical metric. This is consistent with Fig. 6 where most of the RUL values predicted by DAST are close to or smaller than the actual RUL, which is desirable because overestimation of RUL could cause more serious consequences than underestimation.

5) *Visualization of the learned weights:* One advantage of DAST is that, the two encoders can automatically learn the weights of different sensors and time steps, which represents their importance for the real-time RUL prediction. The weight information is not only useful in improving the prediction performance, but also can be understood by the maintenance personnel to focus on more important sensors and time steps in real time and thus improve maintenance efficiency. To visualize this point, we choose 30 consecutive working time cycles (150-180) of engine unit 99 in the F003 dataset, and plot the average weights of each sensor and time step in the corresponding sliding window in Fig. 7. We can see that in this period, sensors T50, Nc, and phi are more important than other sensors, while for time steps, those from 15 to 24 and the last step are more informative for RUL prediction.

6) *Experimental Results on the PHM 2008 dataset:* For this experiment, we apply the same DAST model as in the C-MAPSS experiments, with sliding time window size $T_w = 60$.

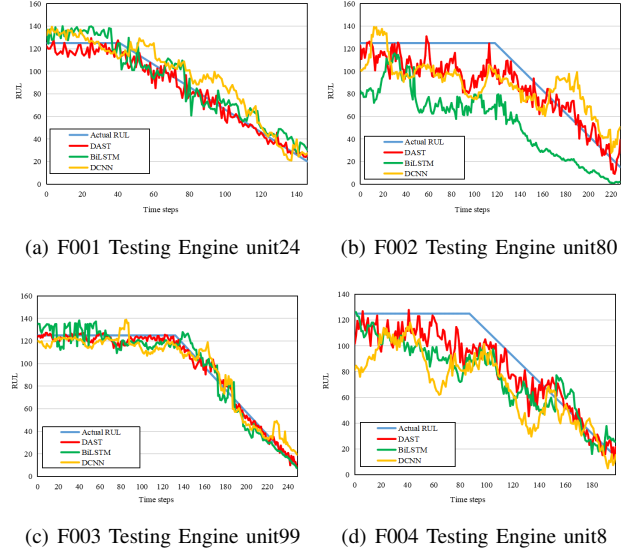


Fig. 6. RUL prediction results on four test engine units.

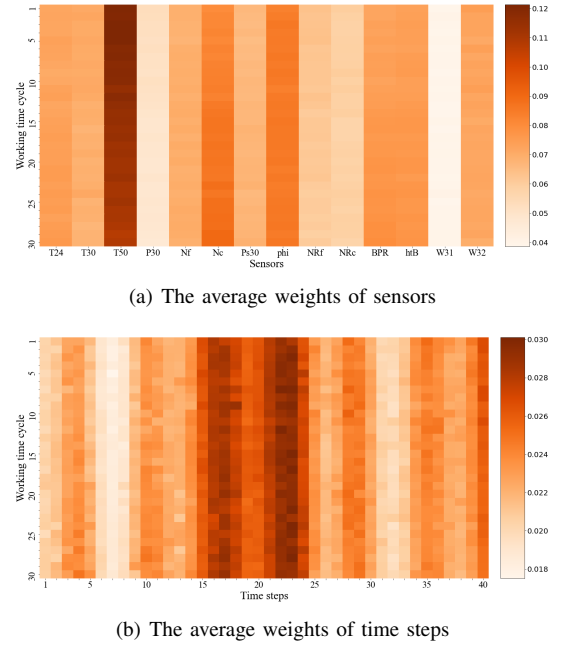


Fig. 7. The average weights of each sensor and time step.

Because the PHM 2008 dataset does not contain actual RUL labels, we need to upload the prediction results to the NASA data repository website¹ to get the Score metric for evaluation. In Table VI, we list the performance of DAST and several recent methods. As we can see, DAST significantly outperforms all the baselines. Compared to the most competitive method in [28], DAST achieves an improvement of 46.7%, showing its effectiveness in this dataset.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel deep RUL prediction method named Dual Aspect Self-attention based on Trans-

¹<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>

TABLE IV
RMSE COMPARISON WITH STATE-OF-THE-ART METHODS

Dataset	RF [1]	LSTM [12]	BiLSTM [27]	CNN [3]	DCNN [14]	MT-CNN [11]	AGCNN [16]	DAST	Imp
F001	17.91	16.14	13.65	18.44	12.61	12.58	12.42*	11.43	7.97%
F002	29.59	24.49	23.18	30.29	22.36	19.77	19.43*	15.25	21.51%
F003	20.27	16.18	13.74	19.81	12.64	12.11*	13.39	11.32	6.52%
F004	31.12	28.17	24.86	29.15	23.31	19.98*	21.50	18.36	8.10%
Average value	24.72	21.24	18.85	24.42	17.73	16.11*	16.68	14.09	12.54%

* means the best result in the baseline methods, **bold** means the best result in all methods.

TABLE V
SCORE COMPARISON WITH STATE-OF-THE-ART METHODS

Dataset	RF [1]	LSTM [12]	BiLSTM [27]	CNN [3]	DCNN [14]	MT-CNN [11]	AGCNN [16]	DAST	Imp
F001	479.73	338	295	1287	273.7	224	225.51*	203.15	9.91%
F002	7046	4450	4130	13570	10412	2023	1492*	924.96	38.06%
F003	711.13	852	317	1596	284.1	334	227.09*	154.92	31.78%
F004	4656	5550	5430	7886	12466	2097*	3392	1490.72	28.94%
Average value	3223.21	2797.50	2543	6084.75	5858.95	1169.5*	1334.15	693.43	40.71%

* means the best result in the baseline methods, **bold** means the best result in all methods.

TABLE VI
RESULTS ON THE PHM 2008 DATASET

Method	Score	Method	Score
MLP [14]	3212	Deep LSTM [12]	1862
RVR [14]	8242	Chen et al. [28]	1584*
CNN [14]	2056	DAST	845

former (DAST). Without any RNN/CNN structure, DAST uses the self-attention mechanism to process entire CBM data sequence. In particular, it is designed based on parallel feature extraction scheme, which employs a sensor encoder and a time step encoder to capture the weighted features of different sensors and time steps simultaneously. We conduct ablation studies to prove the effectiveness of our design. Experimental results on two real turbofan engine datasets show that the RUL prediction performance of our method is superior to state-of-the-art deep RUL prediction methods. In the future, we would like to explore the combination of deep learning methods and data enhancement techniques to resolve the issue of lacking training data, which is important for practical RUL prediction.

REFERENCES

- [1] 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.
- [2] 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. 1–1, 2017.
- [3] 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*, 2016.
- [4] J. Park, J. M. Ha, H. Oh, B. D. Youn, J.-H. Choi, and N. H. Kim, "Model-based fault diagnosis of a planetary gear: A novel approach using transmission error," *IEEE Transactions on Reliability*, vol. 65, no. 4, pp. 1830–1841, 2016.
- [5] 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*, vol. 17, no. 9, pp. 6438–6447, 2021.
- [6] H. Meng and Y.-F. Li, "A review on prognostics and health management (phm) methods of lithium-ion batteries," *Renewable and Sustainable Energy Reviews*, vol. 116, p. 109405, 2019.
- [7] T. H. Loutas, D. Roulas, and G. Georgoulas, "Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression," *IEEE Transactions on Reliability*, vol. 62, no. 4, pp. 821–832, 2013.
- [8] Z. Liu, Y. Cheng, P. Wang, Y. Yu, and Y. Long, "A method for remaining useful life prediction of crystal oscillators using the bayesian approach and extreme learning machine under uncertainty," *Neurocomputing*, vol. 305, pp. 27–38, 2018.
- [9] J. Guo, Z. Li, and M. Li, "A review on prognostics methods for engineering systems," *IEEE Transactions on Reliability*, vol. 69, no. 3, pp. 1110–1129, 2019.
- [10] A. Malhi, R. Yan, and R. X. Gao, "Prognosis of defect propagation based on recurrent neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 703–711, 2011.
- [11] T. San Kim and S. Y. Sohn, "Multitask learning for health condition identification and remaining useful life prediction: Deep convolutional neural network approach," *Journal of Intelligent Manufacturing*, pp. 1–11, 2020.
- [12] 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)*, 2017.
- [13] J. Chen, H. Jing, Y. Chang, and Q. Liu, "Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process," *Reliability Engineering & System Safety*, vol. 185, pp. 372–382, 2019.
- [14] 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.
- [15] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.
- [16] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1197–1207, 2020.
- [17] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, "Lstm networks based on attention ordered neurons for gear remaining life prediction," *ISA Transactions*, vol. 106, pp. 343–354, 2020.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.
- [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] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series

- forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [21] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” *arXiv preprint arXiv:2004.05150*, 2020.
 - [22] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A survey of transformers,” *arXiv preprint arXiv:2106.04554*, 2021.
 - [23] 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*, 2008.
 - [24] A. Saxena and K. Goebel, “Phm08 challenge data set,” in *NASA AMES Prognostics Data Repository*, Moffett Field, CA, Tech. Rep, 2008.
 - [25] H. Li, W. Zhao, Y. Zhang, and E. Zio, “Remaining useful life prediction using multi-scale deep convolutional neural network,” *Applied Soft Computing*, vol. 89, p. 106113, 2020.
 - [26] Y. Song, S. Gao, Y. Li, L. Jia, Q. Li, and F. Pang, “Distributed attention-based temporal convolutional network for remaining useful life prediction,” *IEEE Internet of Things Journal*, 2020.
 - [27] J. Wang, G. Wen, S. Yang, and Y. Liu, “Remaining useful life estimation in prognostics using deep bidirectional lstm neural network,” in *2018 Prognostics and System Health Management Conference (PHM)*, 2018.
 - [28] 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.