A Dual-Scale Transformer-Based Remaining Useful Life Prediction Model in Industrial Internet of Things

Junhuai Li, Kan Wang, Xiangwang Hou, Dapeng Lan, *Member, IEEE*, Yunwen Wu, Huaijun Wang, Lei Liu, *Member, IEEE*, and Shahid Mumtaz, *Senior Member, IEEE*

***Abstract*—With recent advents of industrial Internet of Things (IIoT), the connectivity and data collection capabilities of industrial equipment have be significantly enhanced, yet bringing new challenges for the remaining useful life (RUL) prediction. To fulfill the RUL predicting demand in multivariate time series, this work proposes an encoder-decoder model termed as dual-scale transformer model (DSFormer), built upon the Transformer architecture. First, in the encoder part, a dual-attention module is designed for the weight feature extraction from both dimensions of the sensor and time series, aiming to compensate for the diverse impacts of different sensors on the prediction. Next, a temporal convolutional network (TCN) module is introduced to capture sequence features and alleviate the loss of positional information incurred by stacking blocks. Then, the feature decomposition module is integrated into the decoder for trend feature extraction from sequences, providing the model with additional sequence information. Finally, compared to existing models, the proposed method can obtain the superior performance in terms of the root mean square error (RMSE) and Score metrics on the FD001, FD002 and FD003 subsets of the C-MAPSS dataset, with an average improvement of 3.2% and 2.5% respectively. In particular, the ablation experiment further validates the effectiveness of proposed modules in handling multivariate time series and extracting features.**

***Index Terms*—Industrial Internet of Things (IIoT), multi-sensor data, remaining useful life (RUL), attention mechanism, Transformer.**

I. Introduction

This work was supported in part by the National Natural Science Foundation of China under Grants 61801379 and 61971347, and in part by the 6G-SENSES project from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programm under Grant Agreement No. 101139282. (*Corresponding author: Kan Wang*.)

Junhuai Li, Kan Wang, Yunwen Wu, and Huaijun Wang are with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China, are also with Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an University of Technology, Xi'an 710048, China. (e-mail: lijunhuai@xaut.edu.cn; wangkan@xaut.edu.cn; 2201220019@stu.xaut.edu.cn; wanghuaijun@xaut.edu.cn)

Xiangwang Hou is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: hxw21@mails.tsinghua.edu.cn)

Dapeng Lan is with the Department of Informatics, University of Oslo, 0373 Oslo, Norway (email: dapengl@ifi.uio.no)

Lei Liu is with the Xidian Guangzhou Institute of Technology, Guangzhou 510555, China, and is also with Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi 710121, China (e-mail: tianjiaoliulei@163.com)

Shahid Mumtaz is with the Department of Applied Informatics, Silesian University of Technology Akademicka, 16 44-100 Gliwice, Poland, and also with the Department of Computer Sciences, Nottingham Trent University, NG1 4FQ Nottingham, U.K. (e-mail: dr.shahid.mumtaz@ieee.org)

INDUSTRIAL Internet of Things (IIoTs) and digital twins (DTs) have emerged as two significant driving forces leading the industrial revolution [1], [2]. IIoTs enable the intelligent manufacturing, intellectual management, and production optimization in the factory, by connecting sensors, devices, and production lines to the Internet, and thereby facilitating the real-time data collection and exchange [3]–[6]. Besides, the DT, as the crucial component of the IIoT, utilizes digital technologies and data models to connect physical entities with their virtual counterparts, promoting the instantaneous monitoring, simulation and prediction on physical entities. Thus, the production efficiency, cost reduction and product quality are further improved. In particular, DTs have been extensively applied in various fields [7]–[10]. In the urban planning, by establishing a digital model of the city, DTs can simulate urban traffic flow, energy consumption and environmental pollution, contributing to the optimization of urban infrastructure layout and resource allocation [11]. Further, in the agriculture, crop growth environments and requirements can be simulated, rendering wiser decisions [12]. Besides, by leveraging DTs in the industrial manufacturing sector, one can promote the monitoring and optimization of production line operations, the prediction of maintenance needs, the improvement of equipment utilization, and the reduction of failure rates [13], [14].

The remaining useful life (RUL) prediction becomes attractive in the predictive maintenance (PDM) field [7] when DTs are brought into the IIoT and manufacturing, ensuring the reliability and safety of systems during the production [15]. The RUL prediction approaches could be categorized into model-based as well as data-driven ones [16], [17]. First, in the former, the physical degradation pattern is designed built on equipment failure mechanisms; the obscure failure of complicated equipment, nevertheless, render the model-based ones challenging [18], [19]. Second, as monitoring technologies rapidly evolve, a large amount of operational data can be obtained from the electromechanical equipment, making the data-driven RUL prediction more promising [17]. Particularly in the industrial big data domain, machine learning as well deep learning-based methods are rapidly advancing. More precisely, in the data-driven RUL prediction, results are obtained through the data preprocessing and model training. Preprocessing typically involves the data normalization, noise filtering, and feature extraction, while traditional learning methods for the RUL prediction usually involve separate fea-

ture extraction and model training steps. For instance, Loutas *et al.* in [20] combined Bayesian analysis with support vector regression to predict the RUL of rolling bearings, while Liu *et al.* in [21] used extreme learning machine in the crystal oscillator, both of which heavily rely on the manual feature engineering and the expertise.

To enable the automatic feature learning and model training, deep learning has been employed using an end-to-end approach for the automatic valuable feature extraction, leading to better predictive performance in the RUL domain. Convolutional neural networks (CNNs) as well as recurrent neural networks (RNNs) have been extensively utilized to capture spatial and temporal correlations. For instance, Babu *et al.* in [22] used deep CNNs for adaptive feature learning on each sensor's time series. Various improved RNN-based models, e.g., gated recurrent unit (GRU) and long short-term memory (LSTM), are also broadly utilized. Zheng *et al.* in [23] validated the capability of LSTM on the RUL prediction with three datasets, Wang *et al.* in [24] designed a bidirectional LSTM (BiLSTM) method to analyze the health information from two temporal directions, and Behera *et al.* [25] proposed a bidirectional GRU for the RUL prediction.

Yet, for long time series, CNN requires larger convolutional kernels to get the greater receptive field [26], [27], resulting in a weaker ability to capture long-term dependencies. In addition, RNN-based methods might unavoidably drop important message due to the feature extraction through recurrent processing units [28]. Particularly in the RUL prediction, traditional methods typically assume that all input data contributes equally to the output without differentiating ones containing more degradation information [29], [30]. Thus, existing works in either CNN or RNN are typically with the low training efficiency and accuracy in practical industrial applications.

Built on the self-attention mechanism, Transformer has been broadly utilized for processing the sequence, prevailing in the natural language processing and other sequence modeling tasks [31]–[33]. Different from CNN or RNN methods in [29], [30], Transformer can focus on features with the greater impact on prediction results and capture long-term correlations between different moments in time series, thereby enhancing the RUL prediction performance [34], [35]. Nevertheless, using Transformer for the RUL prediction still poses some challenges: 1) Transformer-based time series prediction models only use self-attention to focus on time step weights, neglecting the sensor weight information present in the RUL prediction. 2) Transformer lacks a network structure that is sensitive to the positional information in the sequence. Although introducing positional encoding layers can embed the positional information, it gradually diminishes as the encoder-decoder blocks are stacked [36]. Moreover, Transformer-based approach still lags behind time serial feature extractors like RNNs [37]. Thus, while Transformer has shown promising results in various sequence modeling tasks, its direct application to the RUL prediction still requires addressing the aforementioned two issues to fully leverage its potential.

Further, with the real-time monitoring and data sharing enabled by IIoT technologies, the RUL prediction can enhance the production efficiency, production quality, and equipment reliability. Motivated by the possible advantage of Transformer, we propose a dual-scale Transformer model, named as DSFormer, to address the limitations of existing methods in capturing time series information and long-term dependencies between moments in the RUL prediction in IIoTs. The main contributions are list as follows:

- A dual-scale attention module that simultaneously extracts weight features from both dimensions of sensor and time series is designed, which replaces the multi-head self-attention module in the Transformer encoder to better capture the importance of different sensors.
- In the encoder, a temporal convolutional network (TCN) module is introduced to capture positional relationships in time series, enabling the learning of more local information and slowing down the loss of positional information between multiple encoder blocks.
- In the decoder, a feature decomposition module is designed to extract more abstract representations, by integrating the decomposed features of time series with those automatically learned by the model.

By integrating these improvements, DSFormer aims to enhance the RUL prediction by effectively handling sensor weights, capturing spatial relationships and utilizing feature decomposition, and thereby achieving the smart resilience in the IIoT.

The rest is organized as follows. Section II reviews the principles of Transformer model. Section III provides the structure of proposed model and principles behind each module. Section IV presents the comparison and ablation experiments on public datasets with analyses. Lastly, the work is concluded in Section V.

II. Preliminaries

As one sequence modeling another one, Transformer does not rely on traditional RNN/CNN structures for the feature extraction, but fully utilizes the attention mechanism and feed-forward neural networks. Different from the RNN weak in processing long sequences, Transformer can efficiently capture long-term correlations between entries in sequences, demonstrating its superior performance in the natural language processing and time series prediction.

A. Positional Encoding

Like most models analyzing the sequence data, Transformer first uses the input embedding layer to map the dimensions of input vectors to the predefined model dimension d_{model}, enabling the subsequent feature extraction. Besides, since Transformer does not include sequential structures like RNNs and relies solely on attention mechanisms to analyze sequence data, it is unable to capture the temporal information of sequence. Therefore, apart from the input embedding layer, the positional encoder is typically appended to offer the relative positional information of sequences, using both sine and cosine functions at different frequencies as follows:

$$p_t\left(2i\right) = \sin\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right), \tag{1}$$

and

$$p_t\left(2i+1\right)=\cos\left(\frac{t}{10000^{2i/d_{\text{model}}}}\right), \tag{2}$$

where t is the position of entry in the sequence, and $2i$ and $2i+1$ are the odd and even numbered dimension, respectively. For any distance l, there exists a linear dependency between $\mathbf{p}_t$ and $\mathbf{p}_{t+l}$, which enables the model to learn the positional information of sequence.

B. Attention Mechanism

After obtaining outputs from input and positional embedding layers, Transformer feeds them into stacked encoder and decoder blocks for feature extraction. Each block respectively consists of a multi-head attention mechanism followed by one feed-forward network.

First, given an initial vector $\mathbf{X} \in \mathbb{R}^{N\times d_{\text{model}}}$ with the sequence length as N, the following operations are performed through multiplying $\mathbf{X}$ by different weight matrices $\mathbf{W}^Q \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$, $\mathbf{W}^K \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ and $\mathbf{W}^V \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ to obtain three matrices, i.e., query, key and value, as follows:

$$\mathbf{Q}=\mathbf{X}\mathbf{W}^Q, \tag{3}$$

$$\mathbf{K}=\mathbf{X}\mathbf{W}^K, \tag{4}$$

and

$$\mathbf{V}=\mathbf{X}\mathbf{W}^V, \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{N\times d_{\text{model}}}$, $\mathbf{K} \in \mathbb{R}^{N\times d_{\text{model}}}$ and $\mathbf{V} \in \mathbb{R}^{N\times d_{\text{model}}}$ hold. Therefore, we have the following as

$$\text{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{\text{model}}}}\right)\mathbf{V}, \tag{6}$$

in which the Softmax function processes the dot product of $\mathbf{Q}$ and $\mathbf{K}$ to obtain the weight matrix by rows, and d_{model} is used to scale the dot product results to prevent the gradient vanishing.

Then, the multi-head attention is achieved by employing multiple self-attention mechanisms to learn multiple submatrices, enabling the model to attend various aspects of information, formulated as

$$\begin{aligned}&\text{MultiheadAtt}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)\\ =\;&\text{Concat}\left(\text{head}_1,\cdots,\text{head}_h,\cdots\text{head}_H\right)\mathbf{W}^{\text{o}},\end{aligned} \tag{7}$$

and

$$\text{head}_h=\text{Attention}_h\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right). \tag{8}$$

That is, the input data $\mathbf{X}$ concurrently enters all attention heads to obtain overall H weighted feature matrices $\text{head}_h, 1 \le h \le H$. Next, these H attention heads are concatenated and multiplied by matrix $\mathbf{W}^{\text{o}}$ for a linear transformation to yield the final attention mechanism output.

III. Methodology

First, we introduce the RUL predicting model DSFormer built on the Transformer architecture, as illustrated in Fig. 1. The upper and lower parts of Fig. 1 represent the encoder and decoder blocks, respectively, composed of attention modules and feed-forward neural networks. By stacking multiple blocks, the encoder and decoder structures are realized, built on the Transformer.

In particular, the dual-scale attention module is used to replace the multi-head self-attention module in the traditional Transformer encoder, thereby capturing weighted feature matrices from both dimensions of sensor and time series. Additionally, the model is equipped with a TCN module and a feature decomposition module to provide additional sequential features, such as trend terms, enabling better capturing of temporal patterns in the sequence. The model's training procedure can be summarized as below:

1) Input encoder: Feature extraction is performed separately through the dual-scale attention module and the TCN module.
2) Fusion of different features: The weighted features from both sensor and time step dimensions, along with the positional features of the sequence, are fused to obtain a new feature matrix containing features from different dimensions.
3) Input decoder: The new feature matrix is fed into the decoder, allowing both attentions to the current feature information and that from the encoder. Lastly, the RUL prediction is carried out through a feed-forward neural network, which incorporates the decomposed sequential features.

A. Designing of Dual-Scale Attention Module

Device RUL values primarily hinge on the signal from multiple sensors scattering among different time series. To fully leverage both sensor and temporal information, and analyze the different device degradation patterns contained in various data, we present the dual-scale attention module integrating both sensor and time series information, as illustrated in Fig. 2.

Specifically, the module performs weighted attention from both dimensions of the sensor and time series to obtain the output matrices $\mathbf{F}_t$ and $\mathbf{F}_s$. Subsequently, the information from both aspects is fused to obtain a new feature matrix, defined as

$$\mathbf{F}_d=\text{Concat}\left(\mathbf{F}_s,\mathbf{F}_t\right)\mathbf{W} \tag{9}$$

where $\text{Concat}\left(\cdot\right)$ represents the concatenation operation, and $\mathbf{W}$ is the trainable parameterized matrix, respectively. More precisely, the dual-scale module is completely based on attention mechanisms to handle the long-term dependency information, and employs a parallel strategy to prevent mutual interference between the two weighted modules. Note that, the dual-scale attention over both dimensions can help the model to better understand and utilize the relationship between sensors and time series, and improves the ability to model the evolution of device states.

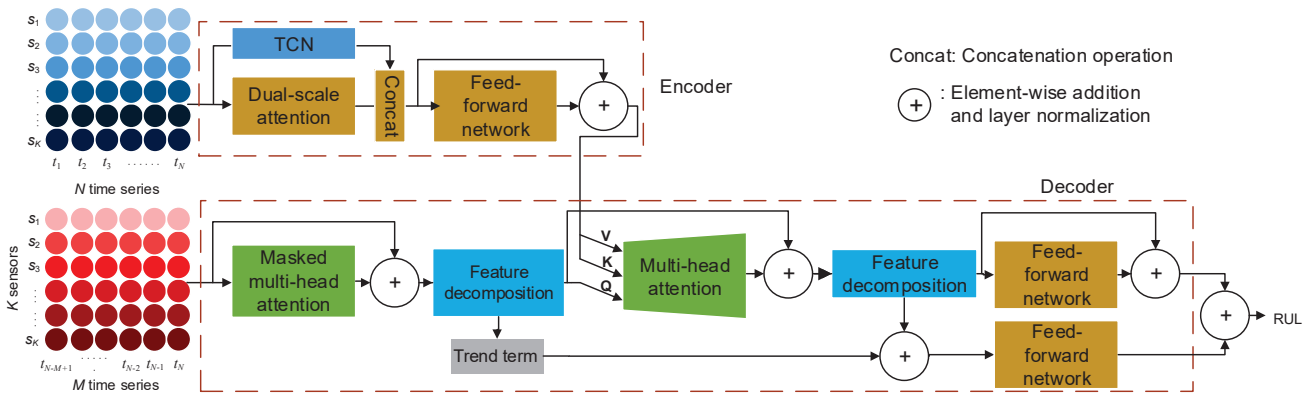


Fig. 1. The architecture of proposed dual-scale Transformer for RUL prediction. For clarity, only one encoder and one decoder, rather than blocks, are presented herein.

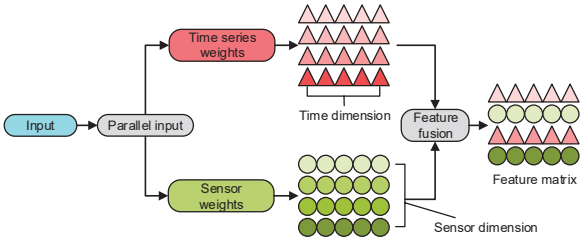


Fig. 2. Dual-scale attention module. Both the temporal and sensor information are fused to build the feature matrix.

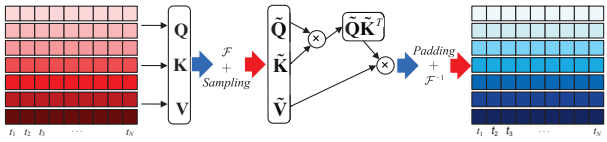


Fig. 3. Frequency attention mechanism. $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are Fourier transformed into and then sampled as $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{V}}$. Then, the frequency-domain attention mechanism is used on $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{V}}$, by incorporating the padding operation.

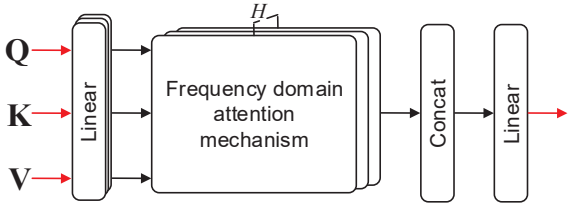


Fig. 4. Multi-head attention mechanism in the frequency domain. By concatenating the outputs from multiple heads and multiplying them by a weight matrix for linear transformation, the final output is obtained.

Afterwards, regarding the time series attention module and compared to time-domain analysis, running various neural network structures in the frequency domain not only reduces the impact of noise on results, but also provides stronger representational capabilities. Particularly in the Transformer model, its further development is constrained by high computational complexity. However, most signals exhibit sparsity in the frequency domain. By sampling fewer frequency domain information, it is possible to minimize the length of analyzed sequence, thus reducing the model's computational complexity. Herein, the Fourier transform works to map the time series as frequency spectrum for analysis. Both Fig. 3 and Fig. 4 illustrate the attention mechanism of time series attention module.

In Fig. 3, $\mathcal{F}$ and $\mathcal{F}^{-1}$ separately denote the Fourier and its inverse transform. When an input sequence $\{s_t\}_{t=1}^{N}$ is predefined, the discrete Fourier transform becomes as $S_m = \sum_{t=0}^{N-1} s_t e^{-jwmt}$, in which $j^2 = -1$ and $\{S_m\}_{m=1}^{M}$ is the complex frequency domain series. Likewise, its inverse transform can be defined as $s_t = \sum_{m=0}^{M-1} S_m e^{jwmt}$. Since the model performs random sampling in the frequency domain, subsequent calculations require padding to ensure that the dimensions of output sequence remain consistent with the input sequence. For the input matrix $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{model}}}$ of the module, we first apply the attention mechanism to obtain $\mathbf{Q} \in \mathbb{R}^{N \times \mathbf{D}_{\text{model}}}$, $\mathbf{K} \in \mathbb{R}^{N \times \mathbf{D}_{\text{model}}}$ and $\mathbf{V} \in \mathbb{R}^{N \times \mathbf{D}_{\text{model}}}$. Next, the Fourier is used to transform $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ into $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{V}}$. Finally, M frequency domain points are selected ($M \ll N$), to reduce the computational complexity. Zhou *et al.* in [38] have also shown that discarding some high-frequency information in the frequency domain has minimal impact on the model's performance when extracting time series features. Therefore,

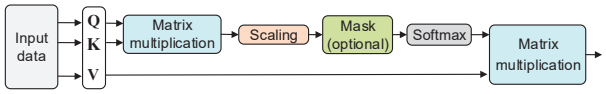


Fig. 5. Scaled dot-product self-attention mechanism in the sensor dimension, which emphasizes more on sensor features with higher weights.

the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are transformed into $\tilde{\mathbf{Q}} \in \mathbb{C}^{M\times \mathbf{D}_{\text{model}}}$, $\tilde{\mathbf{K}} \in \mathbb{C}^{M\times \mathbf{D}_{\text{model}}}$ and $\tilde{\mathbf{V}} \in \mathbb{C}^{M\times \mathbf{D}_{\text{model}}}$, defined as follows:

$$\tilde{\mathbf{Q}} = \text{Select}\left(\bar{\mathbf{K}}\right) = \text{Select}\left(\mathcal{F}(\mathbf{Q})\right), \tag{10}$$

$$\tilde{\mathbf{K}} = \text{Select}\left(\bar{\mathbf{V}}\right) = \text{Select}\left(\mathcal{F}(\mathbf{K})\right), \tag{11}$$

and

$$\tilde{\mathbf{V}} = \text{Select}\left(\bar{\mathbf{Q}}\right) = \text{Select}\left(\mathcal{F}(\mathbf{V})\right), \tag{12}$$

where $\text{Select}(\cdot)$ represents the random sampling in the frequency domain. The frequency-domain attention mechanism is written using $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{V}}$ as follows:

$$\text{Attention}_{\text{freq}}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathcal{F}^{-1}\left(\text{Padding}(\text{Softmax}(\tilde{\mathbf{Q}}\tilde{\mathbf{K}}^{\text{T}})\tilde{\mathbf{V}})\right), \tag{13}$$

where $\text{Padding}(\cdot)$ represents the padding operation. To ensure the consistency between dimensions of the input and output, it is obligatory to perform $\text{Padding}(\cdot)$ before the inverse Fourier transform, extending it to $\mathbb{C}^{N\times d_{\text{model}}}$.

Besides, as shown in Fig. 4 to learn multiple feature submatrices, similar to the Transformer in Section II, we use the multi-head attention mechanism to integrate H frequency domain attentions. By concatenating the outputs from these H heads and multiplying them by a weight matrix for linear transformation, we obtain the final output of the time series attention module. Next, for the sensor one, the multi-head self-attention mechanism [26] is employed to analyze the significance among diverse sensors in the sensor dimension. Therefore, during the training, there is no need for manual intervention, as the model automatically focuses on sensor features with higher weights. The self-attention mechanism is depicted in Fig. 5.

Finally, from (3), (4) and (5), $\mathbf{Q}_s$, $\mathbf{K}_s$ and $\mathbf{V}_s$ are obtained based on the sensor dimension for the input data. More precisely, the self-attention mechanism is used to weight them, i.e.,

$$\text{Attention}_{\text{sensor}}(\mathbf{Q}_s,\mathbf{K}_s,\mathbf{V}_s) = \text{Softmax}\left(\frac{\mathbf{Q}_s\mathbf{K}_s^T}{\sqrt{\mathbf{d}_{\text{model}}}}\right)\mathbf{V}_s. \tag{14}$$

Likewise, the multi-head attention mechanism is proposed in the sensor attention module, allowing the model to learn information from different positions and representation spaces. As such, the final output of the sensor attention module can be expressed as

$$\text{MultiheadAtt}_{\text{sensor}}(\mathbf{Q}_s,\mathbf{K}_s,\mathbf{V}_s) = \text{Concat}\left(\{\text{head}_h\}_{h=1}^{H}\right)\mathbf{W}^s, \tag{15}$$

where $\mathbf{W}^s$ is the parameterized matrix.

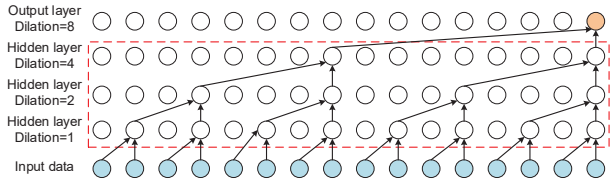


Fig. 6. An illustration of dilated convolution with three hidden layers. The dilation rates of hidden layers are 1, 2, and 4, respectively.

B. Designing of Encoder-Decoder

To improve the performance of Transformer in extracting positional features and sequence patterns, feature extraction structures are designed in both the encoder and decoder. On one hand, the positional features are extracted using the TCN module in the encoder, operating in parallel with the dual-scale attention module and mitigating the vanishing of positional information between encoder blocks. Note that, during stacking of blocks, the original positional encoding maybe insufficient to capture the long-range dependencies of the entire sequence, as each block only focuses on information within a specific range. To overcome the positional information loss, the TCN is introduced to capture long-term dependencies in sequences. On the other hand, the sequence patterns are extracted using the feature decomposition module in the decoder, in which the trending term indicates the changes in the sequence, while the residual term serves as the input for next module.

First, the encoder primarily includes one input embedding layer, one positional encoding layer, one TCN module, one dual-scale attention module, and one feed-forward neural network layer. The first two layers, i.e., input embedding and positional encoding layers remain consistent with the original Transformer. However, to address the limitation of positional encoding layer in capturing positional information, the encoder in parallel utilizes the TCN module and the dual-scale attention module as a new feature extraction structure. On one hand, the TCN module provides additional sequence positional information to the model, thereby mitigating the vanishing of positional information between multiple encoder blocks. On the other hand, the parallel usage of both feature extraction modules also enriches the sequence features available to the model.

Compared to the widely used RNN models, Bai *et al.* in [39] introduced the TCN model in 2018 to address issues like gradient vanishing and the inability to perform parallel computation in RNNs. Particularly for time series data, TCN considers the causal relationship in sequences, adapting it to capture positional dependencies over time. Besides, due to the presence of convolutional operations, TCN can learn more local information from the data. As shown in Fig. 6, the causal convolution and dilated convolution structures are illustrated. In the causal convolution, the data collected at moment t in the focused layer hinges only on the data from previous layer at time t and before, represented as

$$y_t^{n+1} = f\left(x_1^n, x_2^n, \cdots, x_t^n\right), t = 1, 2, \cdots, T, \tag{16}$$

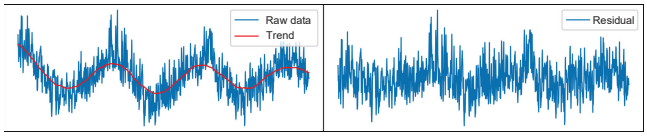


Fig. 7. An illustration of trend feature decomposition. The left sub-figure shows the raw data and its associated trend term, and the right sub-figure represents the residual term.

in which n, T and $f(\cdot)$ are the layer index, time window length and convolution operation, separately. Unlike traditional convolutions, dilated convolutions increase their receptive field through the dilation factor. The model's receptive field becomes wider, as the network depth increases. Note that, the input data consists of multiple sensor time series data. Thus, the TCN module uses one-dimensional convolution kernels to process the data from each sensor separately. After the outputs from the TCN module and the dual-scale attention module are fused, they are passed through the feed-forward network for the ultimate outcome.

Then, it is designed that the decoder is composed of four layers, i.e., an input embedding layer, a multi-head self-attention module, a feature decomposition module, and a feed-forward neural network. Unlike the original Transformer, an additional feature decomposition module is introduced herein. By combining the dis-aggregated features from time series with the ones learned by the model, a more abstract representation of features can be obtained. It has been demonstrated in [40] that incorporating these additional features in various time series prediction models can better reflect the original data and effectively improve the model's performance. Fig. 7 illustrates the feature decomposition for the original data.

In particular, to extract the trend term, it is common to use a fixed window for average pooling. However, in the presence of complicated periodic patterns in real data, this method faces difficulties. It is intuitive that a smaller window length can be more sensitive to details and abrupt features in time series. Thus, we next adopt a set of filters with varying sizes to extract multiple trend terms and then combine them into the final trend using data correlation weights, i.e.,

$$F(\mathbf{X}) = \text{Concat}\left(\{\text{AvgPool}_d(\text{Padding}_d(\mathbf{X}))\}_{d=1}^{d_{\text{model}}}\right), \quad (17)$$

and

$$\mathbf{X}_{\text{trend}} = \text{Softmax}\left(L(\mathbf{X})\right) F(\mathbf{X}), \quad (18)$$

in which $\text{AvgPool}_d(\cdot)$ denotes the average pooling operation along the d-th sensor dimension, and $F(\cdot)$ refers to an internal block containing d_{model} average pooling filters. Padding is used to make the sequence's length stay the same during the process, $\mathbf{X}_{\text{trend}}$ represents the extracted trend components, $L(\cdot)$ stands for the fully connected layer, and $\text{Softmax}\left(L(\mathbf{X})\right)$ represents the weights used to blend the outputs of multiple filters. Meanwhile, the residual term is obtained as

$$\mathbf{X}_{\text{residual}} = \mathbf{X} - \mathbf{X}_{\text{trend}}, \quad (19)$$

where $\mathbf{X}_{\text{residual}}$ continues to serve as the input for subsequent modules.

TABLE I
STATISTICS FOR DATASETS

Dataset	FD001	FD002	FD003	FD004
Training engines	100	260	100	249
Testing engines	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	6	2	2
Training set size	20631	53759	24720	61249
Test set size	100	259	100	248

Finally, the workflow of decoder is partitioned into the following four procedures:

1) Masked multi-head self-attention module. It ensures that the model's predicted results at a given moment only depend on the data before it. In other words, the RUL prediction at moment t is conditioned only on the data from moments $1, 2, \cdots, t-1$. This is because that in the sequence-to-sequence task, the model needs to generate outputs sequentially and gradually, and therefore should not rely on the future information. Thus, masking ensures that the model follows the causality in both training and prediction, thereby improving the model's ability to generalize sequences.
2) Feature decomposition module. Inspired by [41], the original sequence is divided into the trend term and the residual term. After the division, the residual term continues to be fed into the next module for learning, while the trend term participates in the subsequent fusion process.
3) Encoder-decoder multi-head attention module. It receives the output from the encoder as $\mathbf{V}$ and $\mathbf{K}$, and the residual term from the feature decomposition module as $\mathbf{Q}$, to perform multi-head attention computation.
4) Feature fusion. After passing through the feature decomposition module and the feed-forward neural network again, the output is fused with the trend term feature. The final RUL prediction is then obtained through a flatten layer followed by a fully connected layer.

Note that, when training, we optimize the predicted RUL value $\{\hat{r}_t\}_{t=1}^N$ with the ground truth $\{r_t\}_{t=1}^N$ by minimizing the mean square error (MSE) loss function as

$$\text{Loss} = \frac{1}{N}\sum_{t=1}^{N}\left(\hat{r}_t - r_t\right)^2. \quad (20)$$

IV. Experiment Comparison and Analysis

A. Experimental Environment and Parameters

1) Dataset: Experiments are carried out on the extensively utilized RUL prediction dataset C-MAPSS [42], which is actually derived from the system-level engine simulation to simulate normal and fault events over time series of flights. In particular, the data set consists of four subsets, as exhibited in Table I. To evaluate the engines' performance worsening and damage spreading under certain flight conditions and failure modes, FD001-FD004 subsets take effect. Due to the more complicated operational model in FD002 and FD004, it is more tough to evaluate them than FD001 and FD003.

In FD001, there are 21 time series data collected from 21 sensors. However, it is observed that sensors 1, 5, 6, 10, 16, 18 and 19 have constant data throughout the monitoring process and hardly encapsulate any valuable information to predict the RUL. Therefore, the residual 14 sensors (namely, sensors excluding 1, 5, 6, 10, 16, 18 and 19) are selected for prediction. The visualization of sensor data is shown in Fig. 8.

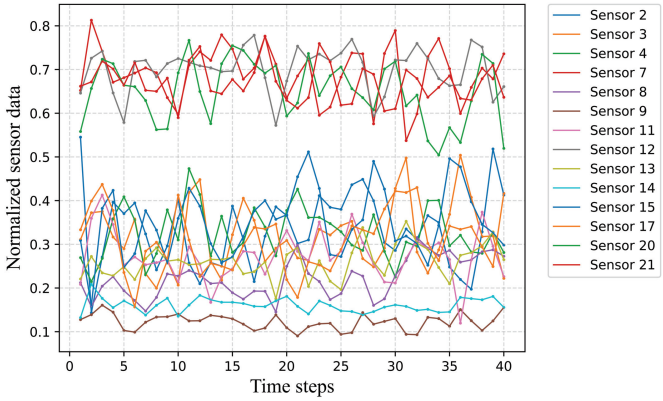


Fig. 8. Multivariate time series data of sensors from FD001. 21 time series, each of which is collected by one sensor, are plotted to shown the fluctuation.

Considering the practical application, the initial degradation process of engine can be neglected. Therefore, the initial RUL of the engine will remain constant until a certain moment when a failure occurs, and then the engine's performance starts to gradually decline. Following [43], the initial RUL is constrained between 120 and 130, thus setting as 125 herein. After the fault happens, the linear diminishing occurs for the RUL value.

2) Evaluation Metrics: Score and root mean square error (RMSE) are taken as verification metrics. Lower RMSE and Score values indicate higher prediction accuracy of neural networks. First, the root mean square of differences between label and evaluated values is defined as RMSE as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(\hat{r}_t - r_t\right)^2} \tag{21}$$

where r_t and $\hat{r}_t$ respectively represent the label and evaluated values of samples, the number of which is N. RMSE can quantify the accuracy of the model in numerical prediction and reflect the average error of the model over the entire time series. Second, Score is a scoring function used by Heimes *et al.* in [44], defined as

$$\text{Score} = \begin{cases} \sum_{t=1}^{N}\left[\exp(-(\hat{r}_t - t_t)/13) - 1\right], r_t > \hat{r}_t, \\ \sum_{t=1}^{N}\left[\exp((\hat{r}_t - r_t)/10) - 1\right], r_t \leq \hat{r}_t. \end{cases} \tag{22}$$

Score focuses more on the consistency of model's prediction, rather than simply on the overall fit of values. Predictions that exceed the true RUL will result in higher score values, mainly due to the optimistic estimation of RUL.

3) Parameter Settings: The dual-scale attention module consists of 8 frequency domain attention heads for time-step weighting and 4 self-attention heads for sensor weighting. The encoder comprises 2 identical sub-layers, each containing a parallel dual-scale attention module and a TCN module, followed by a fully connected ReLU activating layer. Further, the decoder includes 1 sub-layer with 2 multi-head self-attention modules, 2 feature decomposition modules and 1 fully connected ReLU activating layer. The final output is generated through 1 fully connected layer, 1 flatten layer and 1 output layer. Other hyperparameters used include the batch size as 32, the step size as 0.0001, the dropout rate as 0.05, and d_{model} as 128, respectively. Grid search is conducted for hyperparameter tuning, but no significant differences are observed among different parameter combinations.

4) Experimental Environment: Experiments are conducted on a system running a 64-bit Windows 10 22H2 operating system with Python 3.9 and PyTorch 1.11.0, and the hardware setup contains an AMD Ryzen 5 3600 6-Core Processor, an NVIDIA GeForce GTX 1650 graphics card, and a RAM of 16GB.

B. Analysis of Impact of Sliding Window Length on Prediction Results

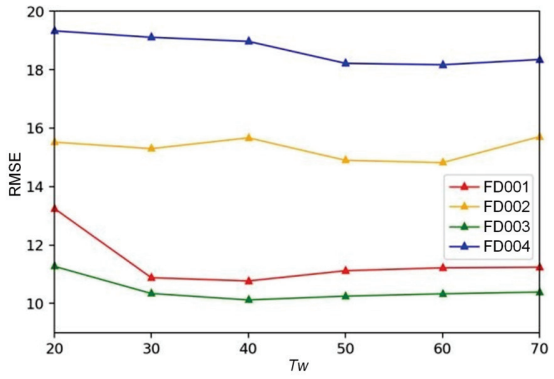
The varying operational situations and modes in four different datasets would affect the correlation between time series length and its associated RUL values. Therefore, for each operational situation of the equipment, the sliding window length is set as T_w to get the best prediction results. Models are tested under conditions with different values of $T_w \in \{20, 30, 40, 50, 60, 70\}$ on all four datasets, with evaluation metrics presented in Fig. 9. The analysis shows that different values of window length would influence the prediction considerably.

More precisely, the best scores are achieved with $T_w = 40$ in the FD003 and FD001 datasets, while the best scores are reached with $T_w = 60$ in the FD004 and FD002 datasets. This is because that the latter two have more complicated operational situations, and longer input sequences can provide the model with more device degradation information, thereby improving the prediction performance.

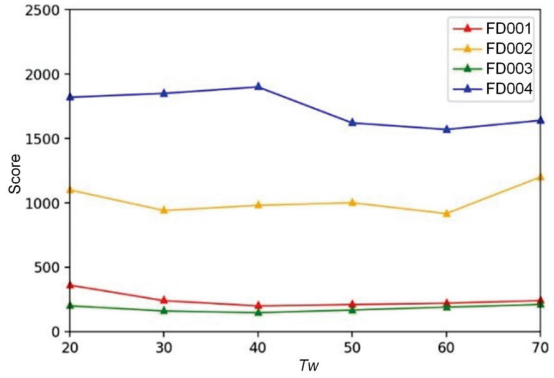
C. Comparison and Analysis of Evaluation Metrics Results

As described in Subsection IV-A, our proposed DSFormer model is compared with other benchmark models on all four datasets of C-MAPSS. Both Tables II and III display the experimental effects of various approaches on all four datasets.

The benchmark models are categorized into three groups: 1) models based on RNN/CNN: LSTM [23], BiLSTM [24], DCNN [43], ELSTMNN [45]; 2) models that integrate RNN/CNN with attention mechanisms: Chen *et al.* [40], AGCNN [50], DATCN [48], DACNN [48], DARNN [49], BiGRU-AS [46], Deep&Attention [47]; 3) models based on the Transformer: DAST [51], DSFormer. In both tables, bold numbers represent the best results achieved on each associated dataset, underlined numbers indicate the second best results, "/" denotes missing results, and "*" signifies the best results



(a) RMSE versus the window length



(b) Score versus window length

Fig. 9. Evaluation metrics versus the window length. Different values of window length affect the prediction accuracy.

TABLE II
RMSE RESULTS

Metrics	RMSE			
Dataset	FD001	FD002	FD003	FD004
LSTM+Attention (2020) [40]	14.53	/	/	27.08
ELSTMNN (2021) [45]	18.22	/	23.21	/
BiGRU-AS (2021) [46]	13.68	20.81	15.53	27.31
Deep&Attention (2021) [47]	12.98	17.04	11.88	19.54
DCNN (2018) [43]	12.61	22.36	12.64	23.31
LSTM (2017) [23]	16.14	24.49	16.18	28.17
BiLSTM (2018) [24]	13.65	23.18	13.74	24.86
DACNN (2020) [48]	12.01	17.21	11.95	18.24
DATCN (2020) [48]	11.78*	16.95*	11.56	18.23
DARNN (2021) [49]	12.04	19.24	10.18*	**18.02***
AGCNN (2020) [50]	12.42	19.43	13.39	21.50
DAST (2022) [51]	11.43	15.25	11.32	18.36
DSFormer (proposed)	**10.77**	**14.82**	**10.12**	18.17
Vanilla	17.92	17.86	27.48	26.56

among the second category of models. Besides, the vanilla Transformer version is also listed, by incorporating only the encoder blocks and waiving decoder ones. More precisely, for the vanilla Transformer, the embedding size of feed-forward network is 128, with two encoder blocks and batchsize of 64. Due to the different values of dataset dimension, the head number for FD001 to FD004 is set as 16, 23, 18 and 23, respectively. Following conclusion are drawn as

1) Proposed DSFormer model gets the best performance in both metrics on the FD001, FD002 and FD003 datasets, but not on FD004, in which it performs second best, trailing only behind DARNN and DAST. Compared to existing models, DSFormer shows an average performance improvement of 3.2% and 2.5% in RMSE and Score metrics, respectively, on the first three datasets. This demonstrates that DSFormer effectively adapts to RUL prediction tasks under different conditions, by attending information weights and extracting more sequence features.

TABLE III
SCORE RESULTS

Metrics	Score↓			
Dataset	FD001	FD002	FD003	FD004
LSTM+Attention (2020) [40]	322.44	/	/	5649.14
ELSTMNN (2021) [45]	571	/	839	/
BiGRU-AS (2021) [46]	284	2454	428	4708
Deep&Attention (2021) [47]	282	1386	222*	2472
DCNN (2018) [43]	273.7	10412	284.1	12466
LSTM (2017) [23]	338	4450	852	5550
BiLSTM (2018) [24]	295	4130	317	5430
DACNN (2020) [48]	238.51	1621.50	316.65	2253.51*
DATCN (2020) [48]	229.48	1842.38	257.11	2317.32
DARNN (2021) [49]	261.95	933.58*	247.85	2587.44
AGCNN (2020) [50]	225.51*	1492	227.09	3392
DAST (2022) [51]	203.15	924.96	154.92	**1490.72**
DSFormer (proposed)	**199.82**	**916.72**	**147.38**	1570.20
Vanilla	406.71	112.84	4595.14	7316.10

2) Among various models, those incorporating attention mechanisms perform better in both metrics compared to the models based solely on RNN/CNN. It suggests that by applying attention mechanisms to focus on key features in multivariate time series, additional weighted information is provided, thus improving the prediction performance of baseline models. Furthermore, the models involving both RNN/CNN and attention mechanisms can be categorized into two groups: (1) those considering both sensor and time series weights, including DATCN, AGCNN, DACNN and DARNN, and (2) those only considering time series weights, such as BiGRU-AS, and Deep&Attention. It is evident that except for the Score metric on FD003, models marked with "*" in the tables belong to the former group, which indicates that considering the contribution levels of variables across multiple dimensions can improve the model's overall performance. It is worth noting that DARNN achieves its weighted attention by employing the widely used channel attention mechanism in convolutional networks, focusing on different channels in the feature maps rather than directly weighting sensors. Although it outperforms other models in some metrics, its overall performance is less stable, which might be attributed to the information loss caused by global average pooling operations.
3) Transformer-based models outperform most models combining RNN/CNN with attention mechanisms, attributed to the fact that in traditional models involving attention mechanisms, the time series are fed sequentially into attention blocks and RNN/CNN blocks, incurring interference between the extracted feature information and hindering further improvement in model accuracy. Meanwhile, the Transformer-based models performs

best on the Score, compared to traditional models. It indicates that when dealing with long time series, the Transformer model, which fully utilizes attention modules to seize the long-range correlation in time series, can achieve superior prediction performance compared to RNN/CNN structures.

4) In the FD004 dataset, DSFormer does not perform as well as DAST (in Score) and DARNN (in RMSE). We must note that there are 6 operating conditions and 2 failure modes in FD002 and FD004, which are more complicated and thus more difficult to predict than FD001 and FD003. From Tables II and III, in FD001, FD002 and FD003, DSFormer always performs best in both metrics. Meanwhile, in the Score, DSFormer is close to DAST and far better than DARNN (and other baselines); and in the RMSE, DSFormer is close to DARNN and far better than DAST (and other baselines). It concludes that our proposed method still achieves competitive results in the complicated operating conditions.
5) DSFormer is far better than the vanilla version in both metrices on all four datasets, since only the encoder but without decoder blocks is built in the vanilla version.

D. Attention Weight Analysis

The dual-scale attention module is a key component, which effectively combines the weights from both sensor and time series dimensions through parallel learning. The experiments above have demonstrated the performance improvement, and weight analysis can help maintainer identify the most important sensors and time moments, thereby improving the fault detection. To demonstrate the model's capability to train sensor and time series weights, the attention weights are visualized for a selected sliding window in the FD001 dataset, shown in Fig. 10.

The sliding window contains 14 sensors and 40 time series on monitoring data. Since the model's training dimension is set to be 560, Fig. 10(b), Fig. 10(c), and Fig. 10(d) identically contain 560 data samples on the horizontal axis. Specifically, both Fig. 10(b) and Fig. 10(c) display the weighted outputs for different time series/sensors, showing significant differences in distribution compared to the original inputs after weightings. In particular, in Fig. 10(b), the initial stages of time series includes key information and trends that would affect the future data, and thus the model tends to focus more on these initial stages to capture the essential features. By integrating the weight information from both dimensions, Fig. 10(d) illustrates the final output of the dual-scale attention module. More especially, the proposed dual-scale module is dedicated to assessing the significance of varying sensors and time series, actively focusing on more crucial information, thereby enhancing the prediction performance.

E. Prediction Results Comparison and Analysis

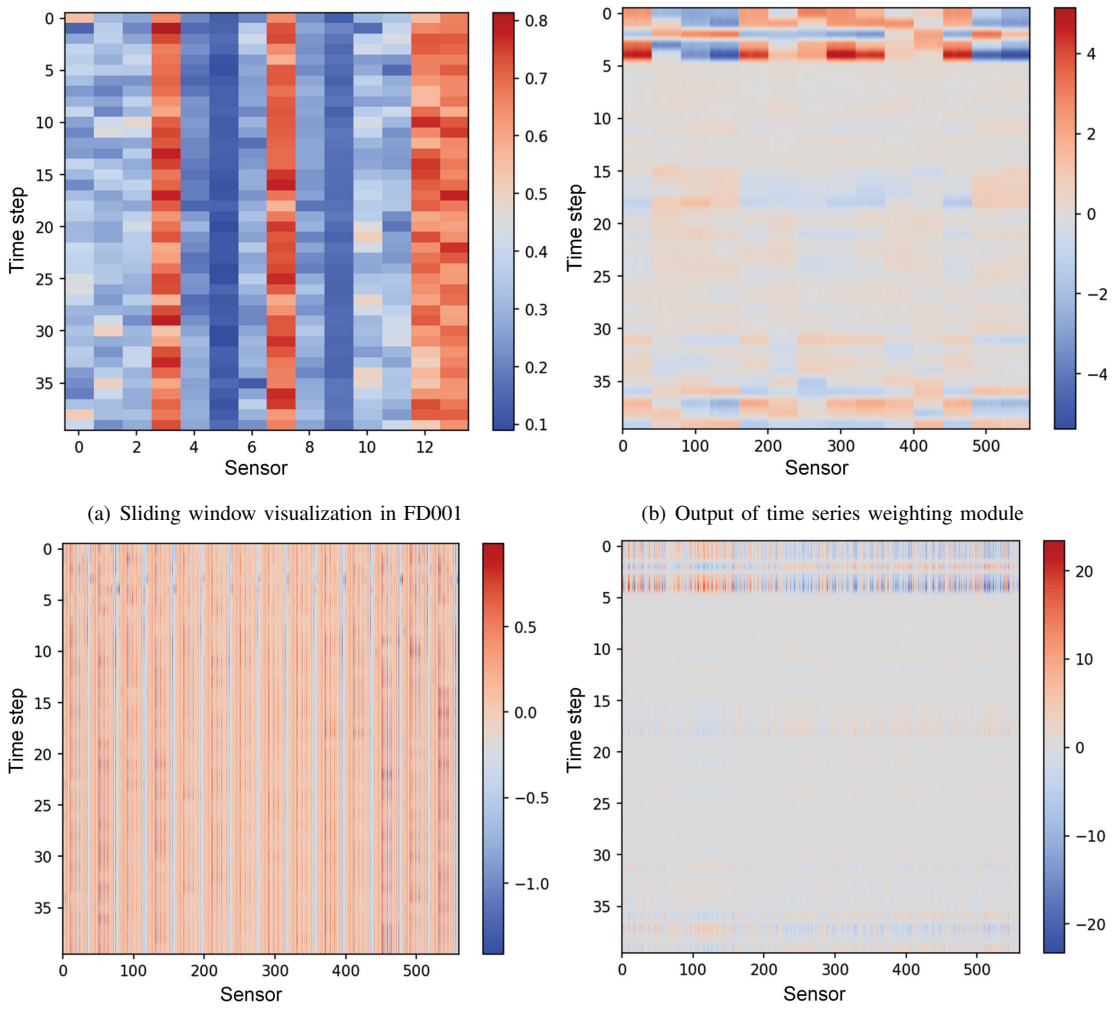
To provide a more intuitive comparison of prediction results among different models, we further run BiLSTM, DAST and DSFormer on the same engine data from four datasets, generating RUL prediction curves. BiLSTM is an RNN-based model, while DAST and our proposed model are both Transformer-based ones. The predicted results from three models are exhibited in Fig. 11. Among both figures, the true RUL values along the time dimension are selected as the prediction targets. From the results, we can observe the following:

1) Compared to the BiLSTM model, both DAST and DSFormer achieve results closer to the true RUL. In the latter part of the plot, the Transformer-based models show smaller prediction errors. This is because that after the engine enters the degrading phase, the data encapsulates more fault and degrading information; the Transformer model, which integrates sensor with time series weights, can better capture these fault features. Besides, in the FD002 dataset with more complicated operating conditions and longer input sequences, Transformer-based models also obtain better prediction results, further demonstrating its advantage in capturing long-term dependencies between moments.
2) From both Transformer-based models, we can see that DSFormer outperforms DAST in terms of prediction results. It is possible that although DAST also considers both sensor and time series weights, stacking the encoder-decoder blocks means that the deepened network depth would drop the positional information. Especially for the FD002 dataset with longer input sequences, the larger prediction errors might occur. On the other hand, DSFormer utilizes the TCN module to learn implicit positional information and enables the transmission of positional information between encoder and decoder modules. Besides, the trend decomposition can preferably seize the long-term correlation in the sequence, thereby improving the prediction accuracy.

F. Ablation Experiments

To analyze the effectiveness of various modules in the proposed DSFormer, ablation experiments are conducted. The key modules of model include the dual-scale attention, the TCN and the feature decomposition modules. For the dual-scale attention module, separate investigations are done using three different configurations: (1) replacing it with a self-attention module, (2) using only the time series attention module, and (3) using a serial combination of time series attention and sensor attention modules. As such, 5 model variants are generated as

1) without the TCN module (abbreviated as "w/o TCN").
2) without the feature decomposition module (abbreviated as "w/o Decomp.").
3) using the self-attention module (referred to as "with Self-Atten.").
4) using only the time series attention module (abbreviated as "with Only-Time").
5) using a serial combination of time series attention and sensor attention modules (abbreviated as "with Serial-Attention").



(a) Sliding window visualization in FD001

(b) Output of time series weighting module

(c) Output of sensor weighting module

(d) Output of dual-scale attention weighting module

Fig. 10. Attention weights visualization. The sliding window contains 14 sensors and 40 time series on monitoring data, and the model's training dimension is set to be 560.

TABLE IV
COMPARISON OF ABLATION EXPERIMENTS

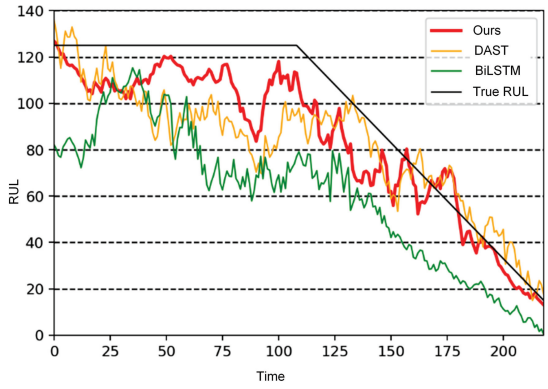
Metrics	RMSE↓	Score↓
w/o TCN	16.02	1645.25
w/o Decomp.	15.43	1276.22
with Self-Atten.	16.11	1697.90
with Only-Time	15.81	1523.76
with Serial-Attention	14.96	962.57
DSFormer (Proposed)	**14.82**	**916.72**

These model variants are evaluated on the FD002 dataset using the aforementioned methods, and their respective evaluation metrics are compared with the original DSFormer model. Results are presented in Table IV, where bold numbers indicate the best-performing variant.

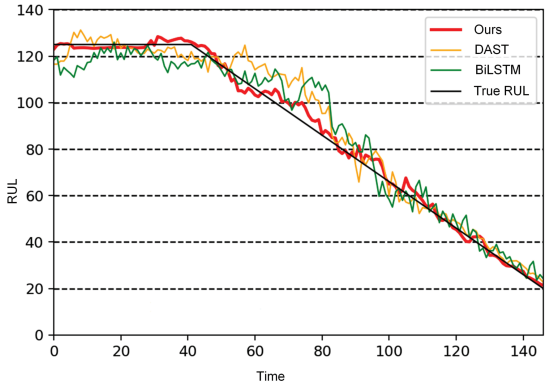
Comparison on the FD002 dataset demonstrates that the original DSFormer model performs best in both RMSE and Score metrics. Particularly, for the dual-scale attention module, using either a serial combination or individual weighted modules leads to a reduction in model performance, verifying the effectiveness of three key modules (i.e., dual-scale attention, TCN and feature decomposition) in the multivariate time series prediction.

V. CONCLUSION AND FUTURE WORKS

The RUL enables IIoT enterprises to implement highly effective maintenance by accurately predicting industrial equipment lifespan, thereby optimizing production and operational efficiency, establishing a more robust interconnected system, and enhancing the reliability and availability of IIoT systems. To meet the demand for the RUL prediction on multivariate time series, this work designs a novel encoder-decoder model built on the Transformer architecture. First, the joint attention between sensor and time steps was implemented to obtain weights from both aspects. In particular, the TCN module, parallel to the dual attention module and together forms the encoder part, was introduced to capture position features, avoiding the loss of location information incurred by the block stacking. Second, a feature decomposition module was added to the original Transformer structure to extract trend features from the sequence, providing additional sequence information for the model and forming the decoder part. Finally, by



(a) Engine 24 prediction plot in FD001



(b) Engine 80 prediction plot in FD002

Fig. 11. Prediction plots in engine 24 and 80 on FD002. In the latter part of plots, the Transformer-based models show smaller prediction errors.

comparing proposed model with other benchmarks and variant models, experiments on the C-MAPSS dataset could verify its validity. In the future, we will further consider the model compression and model partitioning to lower the computing resource demand by the Transformer, adapting it to deployed in the edge.

REFERENCES

[1] J. Li, R. Wang, and K. Wang, "Service function chaining in industrial Internet of Things with edge intelligence: A natural actor-critic approach," *IEEE Trans. Ind. Inform.*, vol. 19, no. 1, pp. 491–502, 2023.

[2] X. Hou, J. Wang, Z. Fang, Y. Ren, K.-C. Chen, and L. Hanzo, "Edge intelligence for mission-critical 6G services in space-air-ground integrated networks," *IEEE Netw.*, vol. 36, no. 2, pp. 181–189, 2022.

[3] Y. Lu and X. Xu, "Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services," *Robot. Comput. Integr. Manuf.*, vol. 57, pp. 92–102, 2019.

[4] L. Bu, Y. Zhang, H. Liu, X. Yuan, G. Jia, and S. Han, "An IIoT-driven and AI-enabled framework for smart manufacturing system based on three-terminal collaborative platform," *Adv. Eng. Inform.*, vol. 50, p. 101370, 2021.

[5] C. Meshram, R. W. Ibrahim, A. J. Obaid, S. G. Meshram, A. Meshram, and A. M. A. El-Latif, "Fractional chaotic maps based short signature scheme under human-centered IoT environments," *J. Adv. Res.*, vol. 32, pp. 139–148, 2021.

[6] X. Hou, J. Wang, C. Jiang, X. Zhang, Y. Ren, and M. Debbah, "UAV-enabled covert federated learning," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 10, pp. 6793–6809, 2023.

[7] D. Zhong, Z. Xia, Y. Zhu, and J. Duan, "Overview of predictive maintenance based on digital twin technology," *Heliyon.*, vol. 9, no. 4, p. e14534, 2023.

[8] F. Tao, W. Liu, M. Zhang, T.-l. Hu, Q. Qi, H. Zhang, F. Sui, T. Wang, H. Xu, Z. Huang *et al.*, "Five-dimension digital twin model and its ten applications," *CIMS.*, vol. 25, no. 1, pp. 1–18, 2019.

[9] T. Brockhoff, M. Heithoff, I. Koren, J. Michael, J. Pfeiffer, B. Rumpe, M. S. Uysal, W. M. Van Der Aalst, and A. Wortmann, "Process prediction with digital twins," in *Proc. ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Fukuoka, Japan, 2021, pp. 182–187.

[10] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15 513–15 526, 2023.

[11] T. Ruohomaki, E. Airaksinen, P. Huuska, O. Kesaniemi, M. Martikka, and J. Suomisto, "Smart city platform enabling digital twin," in *Proc. International Conference on Intelligent Systems (IS)*, Funchal, Portugal, 2018, pp. 155–161.

[12] J. Reitz, M. Schluse, and J. Rossmann, "Industry 4.0 beyond the factory: An application to forestry," in *Proc. Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*, Heidelberg, Germany, 2019, pp. 107–116.

[13] H. Wang, M. Zhou, and B. Liu, "Tolerance allocation with simulation-based digital twin for CFRP-metal countersunk bolt joint," in *Proc. ASME International Mechanical Engineering Congress and Exposition (IMECE)*, Pittsburgh, USA, 2018, pp. 9–15.

[14] J. Du, C. Jiang, A. Benslimane, S. Guo, and Y. Ren, "SDN-based resource allocation in edge and cloud computing systems: An evolutionary Stackelberg differential game approach," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1613–1628, 2022.

[15] Z. Huang, Y. Yang, Y. Hu, X. Ding, X. Li, and Y. Liu, "Attention-augmented recalibrated and compensatory network for machine remaining useful life prediction," *Reliab. Eng. Syst. Saf.*, vol. 235, p. 109247, 2023.

[16] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal. Process.*, vol. 104, pp. 799–834, 2018.

[17] E. Zio, "Prognostics and health management (PHM): where are we and where do we (need to) go in theory and practice," *Reliab. Eng. Syst. Saf.*, vol. 218, p. 108119, 2022.

[18] Y. Bi, Y. Yin, and S.-Y. Choe, "Online state of health and aging parameter estimation using a physics-based life model with a particle filter," *J. Power. Sources.*, vol. 476, p. 228655, 2020.

[19] Y. Zhao and Y. Wang, "Remaining useful life prediction for multi-sensor systems using a novel end-to-end deep-learning method," *Measurement*, vol. 182, p. 109685, 2021.

[20] T. H. Loutas, D. Roulias, and G. Georgoulas, "Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic E-support vectors regression," *IEEE Trans. Reliab.*, vol. 62, no. 4, pp. 821–832, 2013.

[21] 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.

[22] G. S. Babu, P. Zhao, and X. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Proc. Database Systems for Advanced Applications (DASFAA)*, Dallas, USA, 2016, pp. 214–228.

[23] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE International Conference on Prognostics and Health Management (ICPHM)*, Dallas, USA, 2017, pp. 88–95.

[24] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional LSTM neural network," in *Proc. Prognostics and System Health Management Conference (PHM)*, Chongqing, China, 2018, pp. 1037–1042.

[25] S. Behera, R. Misra, and A. Sillitti, "Multiscale deep bidirectional gated recurrent neural networks based prognostic method for complex non-linear degradation systems," *Inf. Sci.*, vol. 554, pp. 120–144, 2021.

[26] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. International Conference on Neural Information Processing Systems*, Red Hook, USA, 2016, pp. 4905–4913.

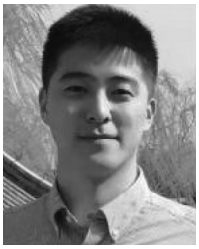
[27] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF conference on computer vision and pattern recognition*, New Orleans, USA, 2022, pp. 11 963–11 975.

[28] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classi-

fication," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2098–2117, 2022.
[29] P. M. Kebria, A. Khosravi, S. Nahavandi, P. Shi, and R. Alizadehsani, "Robust adaptive control scheme for teleoperation systems with delay and uncertainties," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3243–3253, 2020.
[30] W. Peng, Z.-S. Ye, and N. Chen, "Bayesian deep-learning-based health prognostics toward prognostics uncertainty," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, 2020.
[31] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers effective for time series forecasting?" in *Proc. AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.
[32] B. Li, W. Cui, L. Zhang, C. Zhu, W. Wang, I. W. Tsang, and J. T. Zhou, "DifFormer: Multi-resolutional differencing transformer with dynamic ranging for time series analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13 586–13 598, 2023.
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Long Beach, USA, 2017, pp. 5998–6008.
[34] H. M. Usman, R. ElShatshat, and A. H. El-Hag, "Distribution Transformer remaining useful life estimation considering electric vehicle penetration," *IEEE Trans. Power Deliv.*, vol. 38, no. 5, pp. 3130–3141, 2023.
[35] H. Peng, B. Jiang, Z. Mao, and S. Liu, "Local enhancing Transformer with temporal convolutional attention mechanism for bearings remaining useful life prediction," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.
[36] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of Transformers," *AI Open*, vol. 3, pp. 111–132, 2022.
[37] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. Kiran *et al.*, "RWKV: Reinventing RNNs for the Transformer era," *arXiv preprint arXiv:2305.13048*, 2023, https://doi.org/10.48550/arXiv.2305.13048.
[38] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. International Conference on Machine Learning (ICML)*, Baltimore, USA, 2022, pp. 27 268–27 286.
[39] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018, https://doi.org/10.48550/arXiv.1803.01271.
[40] 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 Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2521–2531, 2020.
[41] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A seasonal-trend decomposition," *J. Off. Stat.*, vol. 6, no. 1, pp. 3–73, 1990.
[42] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. International Conference on Prognostics and Health Management (ICPHM)*, Denver, USA, 2008, pp. 1–9.
[43] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, 2018.
[44] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. International Conference on Prognostics and Health Management (ICPHM)*, Denver, USA, 2008, pp. 1–6.
[45] Y. Cheng, J. Wu, H. Zhu, S. W. Or, and X. Shao, "Remaining useful life prognosis based on ensemble long short-term memory neural network," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
[46] Y. Duan, H. Li, M. He, and D. Zhao, "A BiGRU autoencoder remaining useful life prediction scheme with attention mechanism and skip connection," *IEEE Sens. J.*, vol. 21, no. 9, pp. 10 905–10 914, 2021.
[47] Y. Liu and X. Wang, "Deep & attention: A self-attention based neural network for remaining useful lifetime predictions," in *Proc. International Conference on Mechatronics and Robotics Engineering (ICMRE)*, Budapest, Hungary (Virtual), 2021, pp. 98–105.
[48] 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. Things. J.*, vol. 8, no. 12, pp. 9594–9602, 2020.
[49] F. Zeng, Y. Li, Y. Jiang, and G. Song, "A deep attention residual neural network-based remaining useful life prediction of machinery," *Measurement*, vol. 181, p. 109642, 2021.
[50] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Trans. Industr. Inform.*, vol. 17, no. 2, pp. 1197–1207, 2020.
[51] Z. Zhang, W. Song, and Q. Li, "Dual-aspect self-attention based on transformer for remaining useful life prediction," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

Junhuai Li received the B.S. degree in electrical automation from Shaanxi Institute of Mechanical Engineering, Xi'an, China, in 1992, the M.S. degree in computer application technology from Xi'an University of Technology, Xi'an, in 1999, and the Ph.D. degree in computer software and theory from Northwest University, Xi'an, in 2002. He is currently a Professor with the School of Computer Science and Engineering, Xi'an University of Technology. His research interests include the Internet of Things technology and network computing.

Kan Wang received the B.S. degree in broadcasting and television engineering from the Zhejiang University of Media and Communications, Hangzhou, China, in 2009, and the Ph.D. degree in military communications from the State Key Laboratory of ISN, Xidian University, Xi'an, China, in 2016. From 2014 to 2015, he was with Carleton University, Ottawa, ON, Canada, as a Visiting Scholar funded by the China Scholarship Council. Since 2017, he has been with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an. His current research interests include 5G cellular networks, resource management, and massive IoT.



Xiangwang Hou (Graduate Student Member, IEEE) received the B.E. degree in electronic information engineering from the Shandong University of Technology, Shandong, China, in 2017, and the M.E. degree in information and communication engineering from Xidian University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree in electronics and communication engineering with Tsinghua University, Beijing, China. His research interests include UAV/AUV networks, federated learning, and wireless AI.

Dapeng Lan (Member, IEEE) received the Ph.D. degree from the Informatics Department, University of Oslo, Oslo, Norway, in 2022, the M.Sc. degree in ICT innovation from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2016, the second M.Sc. degree in innovation in information and communication technology from the Technical University of Berlin, Berlin, Germany, in 2017. He is the CEO of the Techforgood AS, Norway, and the guest Researcher with the University of Oslo. He was a Postdoc Research Fellow with the Department of Informatics, University of Oslo. His research interests include edge/fog computing, Internet of Things, and distributed systems.

Yunwen Wu received the B.S. degree in network engineering from Xi'an University of Technology, Xi'an, China, in 2020, and the master degree in computer science and technology from the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. He is now an engineer in Bank of China Software Center.

Huaijun Wang received the B.S. and M.S. degrees in computer science from the Xi'an University Technology, Xi'an, China, in 2005 and 2010, respectively, and the Ph.D. degree from Northwest University, Xi'an, in 2014. He is currently an associate professor with Xi'an University of Technology. His research interests include application and security of CPS, and modeling of effectiveness evaluation of security.

Lei Liu (Member, IEEE) received the B.Eng. degree from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 2013 and 2019, respectively, all in communication engineering. He is currently a Lecturer with the Department of Electrical Engineering and Computer Science, Xidian University. From 2013 to 2015, he was with a technology company. From 2018 to 2019, he was a visiting Ph.D. student with the University of Oslo, Oslo, Norway. His research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things.

Shahid Mumtaz (Senior Member, IEEE) is an IET Fellow, IEEE ComSoc and ACM Distinguished Speaker, recipient of IEEE ComSoc Young Researcher Award (2020), founder and EiC of the IET Journal of Quantum Communication, Vice-Chair of the Europe/Africa RegionIEEE Com Soc Green Communications Computing Society, and Vice-Chair of IEEE Standard P1932.1: Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks. He is the author of 4 technical books, 12 book chapters, 300+ technical papers (200+ IEEE journals/transactions, 100+ conference proceedings), and received 2 IEEE best paper awards in the area of mobile communications. Most of his publication is in the field of wireless communication. He is serving as Scientific Expert and Evaluator for various research funding agencies. He was awarded an Alain Bensoussan Fellowship in 2012. He was the recipient of the NSFC Researcher Fund for Young Scientist in 2017 from China.