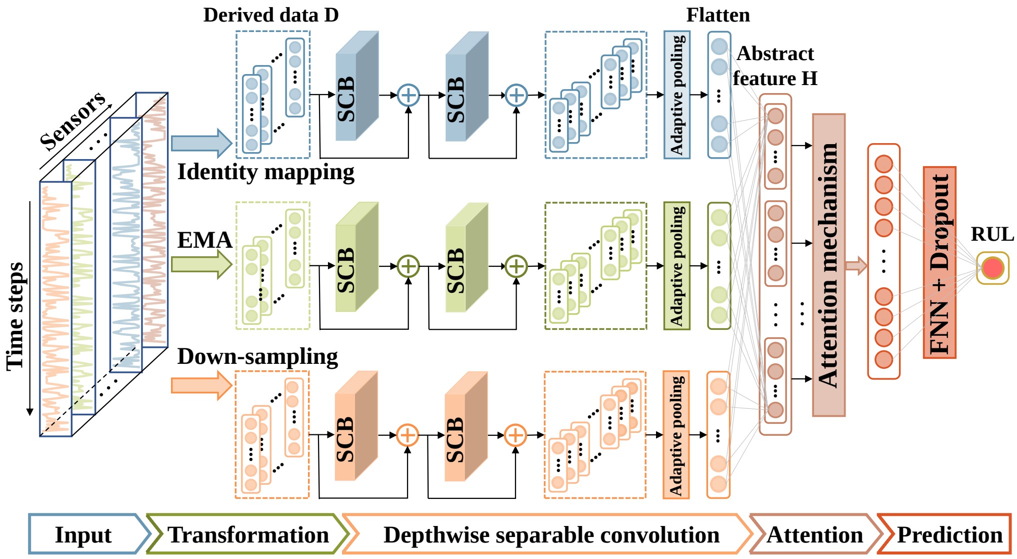
Graphical Abstract

Machine Remaining Useful Life Prediction Using Temporal Deep Degradation Network with Attention-based Dynamic Degradation Patterns



Highlights

Machine Remaining Useful Life Prediction Using Temporal Deep Degradation Network with Attention-based Dynamic Degradation Patterns

- A novel attention-based TDDN model is proposed for the RUL prediction of machines.
- The hyperparameters of the TDDN model are optimized by the hyperband algorithm automatically.
- Deep separable convolutional networks are adopted to learn temporal and spatial features effectively from degradation-related derived data in time and frequency domains.
- The attention mechanism is employed to encode the dynamic patterns of machine degradation.
- The comparative experiments and ablation studies on the C-MAPSS dataset show the superior performance of the proposed method.

Machine Remaining Useful Life Prediction Using Temporal Deep Degradation Network with Attention-based Dynamic Degradation Patterns

Abstract

The accurate prediction of remaining useful life (RUL) is vital for prognostic analysis and predictive maintenance. Since machines' RUL varies for different operating conditions and degradation patterns, this paper proposes a temporal deep degradation network (TDDN) model to effectively capture the dynamic patterns of degradation development to predict RUL. In the TDDN model, multiple data transformation techniques are applied in parallel to extract degradation-related derived data in time and frequency domains. The deep separable convolutional network is then employed to extract abstract features for modeling intra-sensor temporal correlations and inter-sensor spatial correlations in the degradation-related derived data. The attention mechanism is introduced to evaluate the attention weights of abstract feature components at different degradation stages. Finally, the feed-forward neural network with the dropout technique is constructed to predict the RUL of machines. The hyperband algorithm is also adopted to automatically optimize the TDDN model's hyperparameters. The performance of the TDDN model is evaluated on the commercial modular aero-propulsion system simulation dataset. The comparative experiments and ablation studies results show the effectiveness and superiority of the TDDN model for RUL prediction.

Keywords: Prognostics and health management, Remaining useful life, Deep separable convolutional network, Attention mechanism, Hyperband algorithm, Dynamic degradation patterns

1. Introduction

Prognostics maintenance is challenging due to machines' increasingly nonlinear complexity and uncertainty. Prognostics and health management (PHM) technology is designed to make maintenance decisions to improve the reliability, stability, and efficiency of machines[1, 2]. As a technical term that describes the development of faults in PHM, the remaining useful life (RUL) is defined as the period from the current time to the failure within a component[3, 4].The accurate RUL prediction plays a vital role in PHM that monitors and detects the health states of machines to provide precise failure warnings and avoid safety accidents.

Generally, the existing RUL prediction algorithms can be divided into two main categories: model-based approaches and data-driven approaches. Physical model-based approaches describe the mechanical degradation development through a mathematical model[5]. However, the subcomponents of machines are cross-coupled, making some model parameters hard to obtain. It is extremely challenging to build a physical model that accurately simulates the degradation development of machines. In contrast, data-driven approaches do not rely on physical models but on extracting degradation-related features from the historical data[6, 7]. They can effectively reveal the underlying mapping relationships between streaming sensor data and RUL. Hence, data-driven approaches attract huge research interest. They can be further divided into conventional machine learning approaches and deep learning approaches. Conventional machine learning approaches, *e.g.* support vector machine (SVM)[8], hidden Markov model (HMM)[9], extreme learning machine (ELM)[10], are usually combined with expert knowledge and signal processing algorithms to extract features from streaming sensor data to predict RUL. However, they are weak in learning essential degradation-related features when streaming sensor data have very high dimensions. Given the higher dimensions and more complex relationships of streaming sensor data, deep learning approaches are more capable of capturing degradation patterns and achieving accurate RUL prediction.

Deep learning neural networks with the ability of automatic feature extraction and great nonlinear fitting have been widely used in the field of machine prognostics. So far, deep learning approaches, such as long short-term memory (LSTM), convolutional neural network (CNN), and attention mechanism, have provided a promising solution to improve RUL prediction accuracy. LSTM is able to mine degradation patterns from streaming sensor data based on inherent time series processing ability. Wang *et al.*[11] applied LSTM to learn the nonlinear mapping between degradation patterns and RUL. Ahmed Elsheikh *et al.* [12] proposed the bidirectional handshaking LSTM (BHSLSTM) to effectively process the streaming sensor data in both directions. To further meet the challenges of highly nonlinear and multi-dimension streaming sensor data, some researchers combined LSTM with feature extraction methods to predict RUL. Fan *et al.*[13] integrated the autoregressive integrated moving average (ARIMA) model and LSTM to improve RUL prediction accuracy. Ellefsen *et al.*[14] designed a semi-supervised framework combining restricted Boltzmann machines (RBM) and LSTM to predict RUL. Wang *et al.*[15] applied kernel principle component analysis (KPCA) to extract crucial features and LSTM to capture degradation trends from learned features. Besides, considering the distribution discrepancy of training and test data, Fu *et al.*[16] integrated deep residual LSTM (DRLSTM) and two domain modules to improve the domain-invariance of degradation features. Shi *et al.*[17] proposed a dual-LSTM to detect change points that aligned the engine degradation process among the entire datasets. Bae *et al.*[18] utilized LSTM to learn physical health timestep from streaming sensor data, which enhanced the consistency of timestep information to improve the accuracy of RUL prediction. Although LSTM-based methods have achieved competitive performance, there are still several limitations in RUL prediction. LSTM is sequential processing rather than parallel processing over time. The error may propagate to future predictions if the input of previous steps is erroneous. Additionally, LSTM takes more resources to store previous states and time to train networks. It is confronted with memory-bandwidth limitations and hardware acceleration problems when LSTM is deployed in practical applications.

CNN is also a popular deep learning approach with excellent feature extraction ability for RUL prediction. Ding *et al.*[19] proposed a deep convolutional neural network (DCNN) to capture the representative degradation patterns from streaming sensor data. Yang *et al.*[20] developed a double-CNN model architecture in which the first CNN model detected incipient fault point and the second CNN predicted RUL. Compared with traditional CNN, some recent work has shown that specific CNN was more suitable for RUL prediction. DU *et al.*[21] utilized CNN blocks with different kernel sizes to extract multi-scale temporal features from streaming sensor data automatically. Fan *et al.*[22] adopted a fully convolutional network (FCN) to extract high-level degradation features for RUL prediction, where convolutional layers replaced the fully connected layers in standard CNN. Zheng *et al.*[23] improved the convolutional operation of standard CNN by injecting holes into the convolution kernel to gain a larger receptive field, which reduced the network parameters. Cao *et al.*[24] proposed a temporal convolutional network with residual self-attention mechanism (TCN-RSA) framework. TCN was applied to capture long-term degradation information from time-frequency domain, and RSA was introduced to focus on essential features to improve the performance of RUL prediction. All aforementioned approaches pay equal attention to degradation features that are extracted from multi-sensor data. Actually, the data in each sensor measures a different physical property and contains different degrees of degradation information. Hence, it is necessary to consider how to effectively model streaming sensor data's intra-correlations and inter-correlations to extract more accurate degradation features.

Recently, attention-based deep learning architectures have shown their advantages in RUL prediction and attracted the widespread attention of researchers. This is because the attention mechanism dynamically highlights relevant features to obtain degradation patterns corresponding to the health states[25]. Zhao *et al.*[26] applied the attention mechanism on streaming sensor data to weigh different sensors. Zhang *et al.*[27] utilized the attention mechanism in the temporal dimension to assign different weights to corresponding time stances. Zhao *et al.*[28] proposed a two-stage attention-based deep learning framework that combined a feature attention and a temporal attention to focus on critical degradation features. Xu *et al.*[29] established a dual-stream self-attention neural network (DS-SANN) which improved the operation times of self-attention on input data to learn more degradation information. Xia *et al.* [30] developed a distance self-attention network (DSAN) in which the match function of the attention mechanism was redesigned by considering reasonable weights for each time step. Compared with the dot-product and scaled dot-product function in the traditional attention mechanism, the new distance function achieved better accuracy in RUL prediction. From the literature review, it can be found that the attention mechanism is an effective approach to focus on key degradation patterns from streaming sensor data. Accordingly, we will employ the attention mechanism to capture dynamic degradation patterns to improve RUL prediction performance.

Based on the above analysis, we propose a novel data-driven method named temporal deep degradation network (TDDN) for RUL prediction. The TDDN model consists of data transformation, deep separable convolutional network (DSCN), and attention mechanism modules. Specially, the data transformation module and DSCN module are designed to extract abstract features for modeling temporal and spatial correlations in streaming sensor data. The attention mechanism module is adopted to calculate the attention weights of abstract feature components to capture dynamic degradation patterns for RUL prediction. Finally, the TDDN model outperforms the existing deep learning neural network models on four subsets in the commercial modular aero-propulsion system simulation (C-MAPSS) dataset. The main contributions of this paper are as follows:

- The TDDN model is proposed for the RUL prediction of machines. The hyperparameters of TDDN model are optimized by hyperband algorithm automatically.

- Deep separable convolutional networks are adopted to learn temporal and spatial features effectively from degradation-related derived data in time and frequency domains, which extracts richer degradation features from high-dimension sensor data.

- The dynamic patterns of machine degradation are encoded by the attention mechanism into the attention map which can serve as a fingerprint of the degradation under different operating conditions. The degradation fingerprint can significantly improve the accuracy of RUL prediction.

- The comparative experiments and ablation studies are conducted on the benchmark C-MAPSS dataset. The TDDN model achieves better performance on four subsets than other models.

The remaining paper is organized as follows: The proposed TDDN model is introduced and discussed in Section 2. Section 3 illustrates the experimental dataset, training, and results. Section 4 conduct ablation study on the TDDN model and analyze how abstract features and attention maps make the TDDN model predict RUL effectively and accurately. We summarize the paper with conclusions and future work in Section 5.

2. Proposed Model: Temporal Deep Degradation Network

2.1. Problem definition of RUL prediction with TDDN

In the field of mechanical component RUL prediction, the streaming sensor data are $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_i; \cdots; \mathbf{x}_n] \in \mathbb{R}^{n \times m}$ and the corresponding RUL labels are $\mathbf{y} = [y_1; y_2; \cdots; y_i; \cdots; y_n] \in \mathbb{R}^n$, where $\mathbf{x}_i = [x_{i,1}\ x_{i,2}\ \cdots\ x_{i,m}]$ is sensor data at time i, n is the length of the streaming sensor data, and m is the number of sensors. The streaming sensor data $\mathbf{X}$ are usually collected by different types of sensors, so $\mathbf{X}$ contains intra-sensor temporal features and inter-sensor spatial features that represent the two structures within the sensor data and between the sensor data. For multivariate RUL prediction problem, the streaming sensor data are segmented into the sequence of moving windows with padding strategy, and each moving window contains different dynamic degradation patterns. In the sequence of moving windows $[\mathbf{X}_1 \cdots \mathbf{X}_j \cdots \mathbf{X}_n]$, the j-th is defined as $\mathbf{X}_j = \left[\mathbf{x}_{j-w+1}; \mathbf{x}_{j-w+2}; \cdots; \mathbf{x}_j\right] \in \mathbb{R}^{w \times m}$ and labeled as y_j, where w is the window size. The training samples are then expressed as $\left\{(\mathbf{X}_j, y_j)\right\}_{j=1}^{n}$. The TDDN model is trained to capture dynamic degradation patterns and map the moving window $\mathbf{X}_j$ to the RUL label y_j, that is $y_j = g(\mathbf{X}_j)$, where g is the TDDN model.

The overall structure of the TDDN model is shown in Fig. 1. First, the data transformation module has three components and is used to generate degradation-related derived data. The identity mapping, down-sampling, and exponential moving averages are arranged in parallel to help extract degradation patterns from the streaming sensor data individually. Next, the DSCN modules are separately utilized for each degradation-related derived data to extract abstract features via multiple separable convolutional blocks (SCB) and adaptive pooling. Finally, the attention mechanism module is employed to assign higher weights to more important abstract feature components. The output of the attention mechanism is fed into feed-forward neural network (FNN) layers to predict RUL. In addition, a hyperband algorithm is introduced to automatically search the optimal hyperparameters of the TDDN model to improve RUL prediction performance. Below, the three modules in the TDDN model are elaborated.

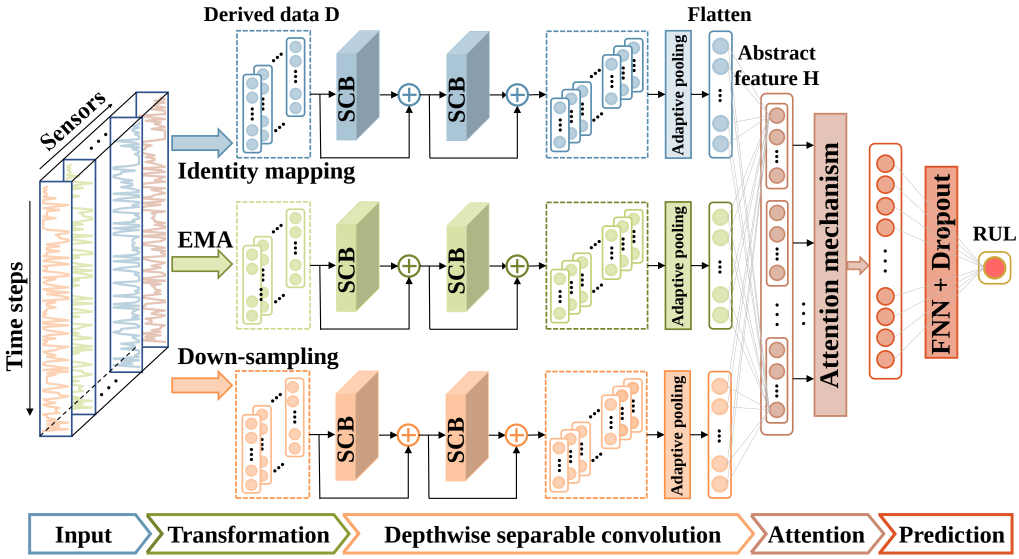


Figure 1: Architecture of the proposed TDDN for RUL prediction.

2.2. Data Transformation

The data transformation module, including the identity mapping and down-sampling in time domain and the exponential moving average (EMA) in frequency domain, is applied to discover more valuable degradation patterns from the moving window[31]. The identity mapping is a function whose output is identical to the input, which is utilized to make the model more compact. The down-sampling helps to identify degradation patterns in the temporal dimension. Given that the input moving window $\mathbf{X}_j$, we can reshape it as $\left[\mathbf{x}_1^j, \cdots, \mathbf{x}_i^j, \cdots, \mathbf{x}_m^j\right]$ by fixing the sensor dimension, where $\mathbf{x}_i^j = \left[x_{1,i} \; x_{2,i} \; \cdots \; x_{w,i}\right]^T$ is the i-th sensor data. The down-sampling is an operation on the moving window X by taking down every k-th point for each sensor $\mathbf{x}_i^j$ as $\left[x_{1+k*r,i}^j, \; r = 0, 1, \ldots, \left\lfloor \frac{n-1}{k} \right\rfloor\right]$.

The EMA is a low-pass filter that removes high-frequency noise from the streaming sensor data. It gives higher weights to recent data with exponentially decreasing weights over time. The formula of EMA is an operation for each sensor $\mathbf{x}_i^j$ expressed as $\left[\alpha \sum_{r=0}^{l-1} (1-\alpha)^r x_{k-r,i}^j,\ k = l, l+1, \ldots, n\right]$ corresponding to $\mathbf{X}_j$, where l is the window size of EMA. α is a constant between 0 and 1 which denotes the importance of recent data and is set to 0.95 in this module.

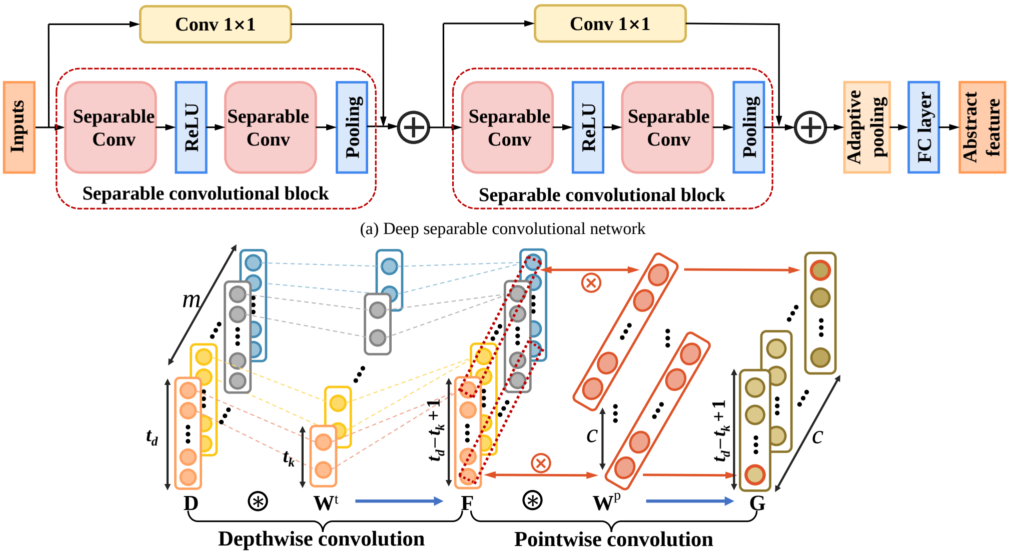
2.3. Deep Separable Convolutional Network and Abstract Features

A standard CNN generally comprises convolutional layers, pooling layers, and a fully connected layer[32]. In convolutional layers, each kernel attaches equal importance to intra-sensor temporal features and inter-sensor spatial features, which ignores the discrepancies of features in different physical sensor data. In addition, the standard convolution always leads to high computational complexity and memory budget with the increasing number of network layers[23]. To deal with those problems, we adopt the DSCN[33] to enhance the feature capability of the TDDN model, in which the SCB is introduced to replace the standard convolutional layer. As shown in Fig. 2 (a), the DSCN contains three main structures: SCB, residual connection, and adaptive pooling.

The separable convolution is the core of SCB, which factorizes standard convolution into two parts, depthwise convolution (Dconv) for temporal features within sensor data, pointwise convolution (Pconv) for spatial features between sensor data. Fig. 2 (b) demonstrates the general process of separable convolution. The Dconv applies one convolutional kernel on each channel to generate the corresponding feature map. Let $\mathbf{D} \in \mathbb{R}^{t_d \times m}$ denotes the one of the outputs of data transformation module and $\mathbf{W}^t \in \mathbb{R}^{t_k \times m}$ represents the weight matrix of depthwise convolutional kernel, where t_d and t_k are the length of degradation-related derived data and kernel, respectively. The feature maps $\mathbf{F}$ learned from $\mathbf{D}$ is defined as,

$$\mathbf{F}_{i,j} = \sum_{k=1}^{t_k} \mathbf{W}_{k,j}^t \cdot \mathbf{D}_{i+k-1,j}\,, \tag{1}$$

where $i = 1, 2, \ldots, t_d - t_k + 1$ and $j = 1, 2, \ldots, m$ .The feature maps $\mathbf{F} \in \mathbb{R}^{(t_d - t_k + 1) \times m}$ are then fed to Pconv to capture spatial features. The Pconv is a weighted combination of these feature maps



(a) Deep separable convolutional network

(b) Separable convolution

Figure 2: The illustration of DSCN for abstract feature extraction.

and can be expressed as,

$$\mathbf{G}_{i,j} = \sum_{k=1}^{m} \mathbf{F}_{i,k} \cdot \mathbf{W}^{p}_{j,k}, \tag{2}$$

where $i = 1, 2, \ldots, t_d - t_k + 1$, $j = 1, 2, \ldots, c$, $\mathbf{W}^p \in \mathbb{R}^{c \times m}$ is the weight matrix of Dconv, and c is the channel size of G. Finally, the SCB effectively models the temporal and spatial features of sensor data by stacking separable convolutions, rectified linear unit (ReLU) activation function, and pooling operation.

Residual connection is used to speed up the training of deep neural networks[24]. As Fig. 2 (a) shows, the residual connection is added between the input and output of SCB, where the 1×1 convolution is adopted to match up the dimension of input and output. Adaptive pooling is introduced to transform the output of DSCN into a fixed length l_f since all degradation-related derived data are in different lengths. Supposing that the length of the output of DSCN is l_o, the pooling size of the pooling layer can be obtained according to the integer ratio of l_o and l_f. After adaptive pooling, as shown in Fig. 1, the outputs of DSCN from different degradation-related derived data are concatenated together and fed to a fully connected layer to obtain abstract feature $\mathbf{H}_j$. The abstract feature $\mathbf{H}_j$ is the pseudo temporal order corresponding to sensor data in the input moving window $\mathbf{X}_j$. $\mathbf{H}_j$ with the same dimension of the input moving window $\mathbf{X}_j$ is expressed as, $\mathbf{H}_j = \left[\mathbf{h}_1^j \ \cdots \ \mathbf{h}_i^j \ \cdots \ \mathbf{h}_m^j\right]$, where $\mathbf{h}_i^j$ represents the abstract feature component with the dimension of $w \times 1$.

2.4. Attention Mechanism and Degradation Attention Weights

The attention mechanism[34] is adopted to calculate the degradation attention weights to encode the dynamic degradation patterns. The attention mechanism generates degradation attention weights according to the similarities between abstract feature components and the degradation stages. The overall structure of the attention mechanism is shown in Fig. 3. The augmentation of abstract features $\mathbf{H}_j$ allows the attention mechanism to learn crucial degradation features. Without loss of generality, the first abstract feature component $\mathbf{h}_1^j$ of the moving window $\mathbf{X}_j$ is used to generate the reference feature $\mathbf{H}_{rj}$. $\mathbf{H}_{rj}$ is defined as $[\mathbf{h}_1^j \ \cdots \ \mathbf{h}_1^j \ \cdots \ \mathbf{h}_1^j]$ whose dimension is the same with $\mathbf{H}_j$. According to the difference and product of abstract feature $\mathbf{H}_j$ and reference feature $\mathbf{H}_{rj}$, two new alternative features $\mathbf{H}_{dj}$ and $\mathbf{H}_{pj}$ are defined as, $\mathbf{H}_{dj} = [0 \ \cdots \ \mathbf{h}_i^j - \mathbf{h}_1^j \ \cdots \ \mathbf{h}_m^j - \mathbf{h}_1^j]$ and $\mathbf{H}_{pj} = [\mathbf{h}_1^j \cdot \mathbf{h}_1^j \ \cdots \ \mathbf{h}_i^j \cdot \mathbf{h}_1^j \ \cdots \ \mathbf{h}_m^j \cdot \mathbf{h}_1^j]$. Then, the four features $\mathbf{H}_j$, $\mathbf{H}_{rj}$, $\mathbf{H}_{dj}$ and $\mathbf{H}_{pj}$, are stacked together as the input $\mathbf{O}_j = [\mathbf{H}_j; \mathbf{H}_{rj}; \mathbf{H}_{dj}; \mathbf{H}_{pj}]$ for the attention layer.

The attention layer is composed of a one-layer multilayer perceptron (MLP) and a softmax function in Fig. 3. Given input matrix $\mathbf{O}_j = \left[\mathbf{o}_1^j \ \cdots \mathbf{o}_i^j \ \cdots \mathbf{o}_m^j\right]$, where $\mathbf{o}_i^j$ is the i-th column vector in $\mathbf{O}_j$, $\mathbf{o}_i^j$ is fed into the MLP to generate the hidden state $\mathbf{u}_i^j$ as,

$$\mathbf{u}_i^j = \tanh\left(\mathbf{W}_a \mathbf{o}_i^j + \mathbf{b}_a\right) , \tag{3}$$

where $\mathbf{W}_a$ and $\mathbf{b}_a$ are the weight and the bias of MLP, respectively. Then, a softmax function is employed to calculate the attention weight according to the correlation score between $\mathbf{u}_i^j$ and a healthy characteristic vector $\mathbf{u}_s$. The higher the score is, the stronger the correlation is, and the abstract feature component will be assigned with a higher degradation attention weight. The healthy characteristic vector $\mathbf{u}_s$ can be interpreted as the mapping of physical states (temperature, pressure, speed, and so on) of the machines. $\mathbf{u}_s$ is learned in training process after being randomly

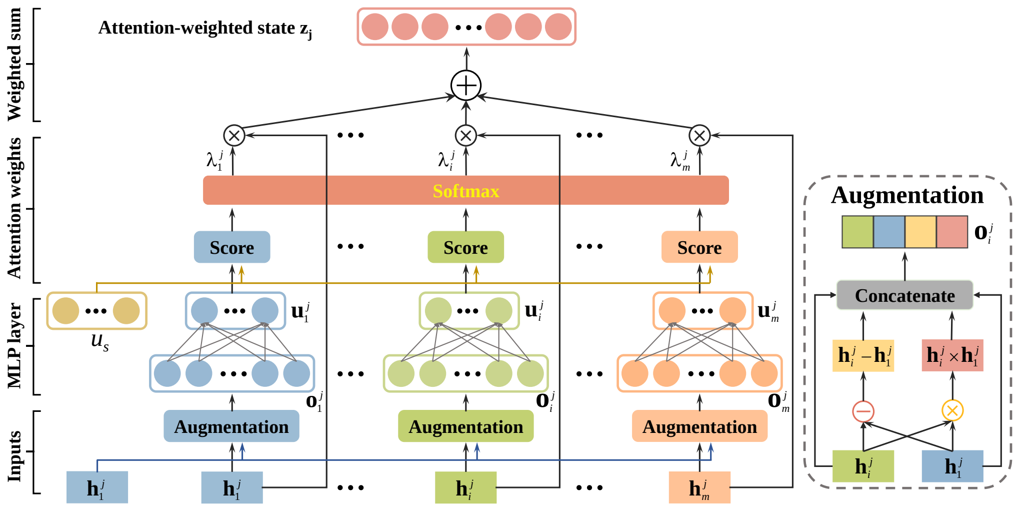


Figure 3: The illustration of the overall structure of attention mechanism

initialized. After that, the degradation attention weight λ_i^j for abstract feature component $\mathbf{h}_i^j$ is computed with softmax function as,

$$\lambda_i^j = \frac{\exp\left((\mathbf{u}_i^j)^T \mathbf{u}_s\right)}{\sum_i \exp\left((\mathbf{u}_i^j)^T \mathbf{u}_s\right)} \,. \tag{4}$$

The attention-weighted state $\mathbf{z}_j$ is a weighted sum of all extracted abstract feature components,

$$\mathbf{z}_j = \sum_{i=1}^{m} \lambda_i^j \mathbf{h}_i^j \,. \tag{5}$$

Finally, the attention-weighted state $\mathbf{z}_j$ is fed into a FNN to predict the RUL y_j. The FNN contains two fully connected layers and a ReLU activation function. The formula of this module is expressed as,

$$y_j = \mathrm{ReLU}\left(\mathbf{z}_j \mathbf{W}_1 + \mathbf{b}_1\right) \mathbf{W}_2 + \mathbf{b}_2 \,, \tag{6}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the learnable weights, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bias. In addition, dropout technique [35] is added to the first fully connected layer to randomly drop units with a fixed probability p when training a network. It is a regularization method that trains multiple networks with different architectures to reduce overfitting and improve the robustness of the TDDN model.

2.5. Hyperparameter Optimization with Hyperband Algorithm

Hyperparameters are set before the model training process and significantly influence the prediction performance of the TDDN model. With the growth of model complexity, the number of hyperparameters can increase exponentially. It is impossible for researchers to manually tune hyperparameters to select an optimal configuration[36]. Therefore, the hyperband algorithm is employed to discover optimal hyperparameters from a high-dimension and non-convex search space.

The hyperband algorithm is an improvement of random search, which adaptively allocates resources to a set of candidate configurations and utilizes the successive halving strategy to speed up convergence. The successive halving strategy selects the best configuration from candidate configurations within multiple rounds of iterating, where every round has the same epochs[37]. Half of the configurations with the worst performance are discarded in each round and repeated until one configuration remains. However, the best configuration may not always perform among the top ones, so it would be discarded in one round when its performance happens to be poor. To avoid this situation, the hyperband algorithm sets different epochs of the round to explore the performance of candidate configurations more widely. Eventually, the best configuration is selected from candidate configurations by combining successive halving strategies and different epochs of the round. Therefore, the hyperband is robust to search for an optimal hyperparameter configuration, and a more detailed description of the algorithm can be found in [38].

3. Experimental Study: A Small Fleet of Turbofan Engines

This section describes the process of sensor selection, data normalization, sample construction,label setting, and the TDDN model training in the C-MAPSS dataset. Then, the proposed method is evaluated on a benchmark dataset and compared with the latest prediction results.

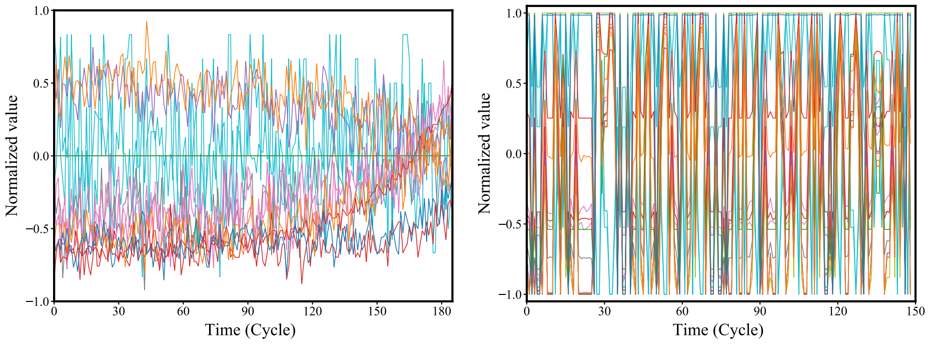
3.1. Data Preprocessing

The C-MAPSS dataset is widely used to evaluate the performance of models. It is generated by a thermo-dynamical simulation model which simulates damage propagation and performance degradation in MATLAB and Simulink environment. The dataset is divided into four subsets according to engine operating conditions and fault modes. Four subsets are denoted as FD001, FD002, FD003, and FD004, respectively. In each subset, engine number, operating cycle number, three operating settings, and 21 sensor measurements are recorded to reflect turbofan engine degradation. The detailed description of the training and test dataset for each subset can be found in the previous work[39]. The operating conditions, fault modes, and other statistical data of each subset are listed in Table 1. The sensor selection, data normalization, moving windows, and padding strategy are implemented for the data preprocessing to improve the computational efficiency of RUL prediction.

Table 1: Information of the C-MAPSS dataset

Dataset	FD001	FD002	FD003	FD004
Engines in training data	100	260	100	259
Engines in test data	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2
Training minimum life cycle	128	128	145	128
Test minimum life cycle	31	21	38	19

The sensor selection of FD001 and FD003 subsets are different from that of the FD002 and FD004 subsets. In FD001 and FD003 subsets, there are twenty-one sensors labeled with the indices of $(1, \cdots, 21)$ and three operating settings of s1,s2, and s3. The twenty-one sensors and operating settings in FD001 subset are visualized in Fig. 4(a). According to the ascending, descending, and unchanged trends, we sort the sensors and operating settings into three trend categories listed in Table 2. Most sensors have ascending or descending trends in the degradation trajectories, while s3 and sensors 1, 5, 6, 10, 16, 18, and 19 remain unchanged and provide no useful degradation information for RUL prediction. Hence, the streaming data in sensors and settings with the ascending and descending trends in Table 2 are used as inputs for the TDDN model. Due to the same operating conditions and degradation patterns, the same trend categories also exist in FD003 subset. However, selecting sensors in FD002 and FD004 subsets are more challenging because they have six operating conditions. Since sensors and settings in FD004 subset have fluctuating trajectories in Fig. 4(b), it is impossible to select relevant sensors in visualization. Hence, all operating settings and sensor measurements are used for the RUL prediction of FD002 and FD004 subsets.



(a) Engine 38 in FD001 training subset

(b) Engine 135 in FD004 training subset

Figure 4: The visualization of the streaming sensor data and settings in FD001 and FD004 subsets

Table 2: The trend categories of the sensors and operating settings in FD001.

Trend	Sensors
Ascending	2, 3, 4, 8, 9, 11, 13, 15, 17, s1, s2
Descending	7, 12, 20, 21, s1, s2
Unchanged	1, 5, 6, 10, 14, 16, 18, 19, s3

Data normalization is implemented to eliminate the difference of magnitudes and units in the streaming sensor data. Each sensor data is normalized to be within the range of $[-1, 1]$ by a min-max scalar. Supposing that $\mathbf{x}_i = [x_{i,1} \ \cdots \ x_{i,n}]$ represents the streaming data of i-th sensor, the i-th normalized sensor value is calculated as,

$$\mathbf{x}_i = \frac{2\left(\mathbf{x}_i - \min\left(\mathbf{x}_i\right)\right)}{\max\left(\mathbf{x}_i\right) - \min\left(\mathbf{x}_i\right)} - 1\,, \tag{7}$$

where $\min\left(\mathbf{x}_i\right)$ and $\max\left(\mathbf{x}_i\right)$ denote the maximum and minimum values in the vector $\mathbf{x}_i$, respectively. Moreover, the normalization of test dataset is based on the maximum and minimum values of training dataset.

In addition, as shown in Table 1, the streaming sensor data in different engines have different lengths. Cycle length discrepancy limits the maximum size of the moving window and affects the accuracy of RUL prediction. The smaller size of the moving window may result in larger prediction errors. In order to eliminate the size limitation of the moving window, the first cycle value $\mathbf{x}_1$ is padded $w-1$ times before original streaming sensor $\mathbf{X}$, which is expressed as,

$$\mathbf{X}_{padding} = [\underbrace{\mathbf{x}_1 \ \cdots \ \mathbf{x}_1}_{w-1} \ \mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_{n-1} \ \mathbf{x}_n] \,. \tag{8}$$

Moving windows and padding strategy are illustrated in Fig. 5. With the padding strategy, the engine RUL can be predicted even if only a small number of cycles are given. After data segmentation, a piecewise linear function is applied to set the RUL labels of the sequence of moving windows $[\mathbf{X}_1 \cdots \mathbf{X}_j \cdots \mathbf{X}_n]$. It is observed that the degradation usually occurred at the last 120-130 cycles[40]. The maximum value of RUL is set to be 125 based on numerous experiments in this paper.

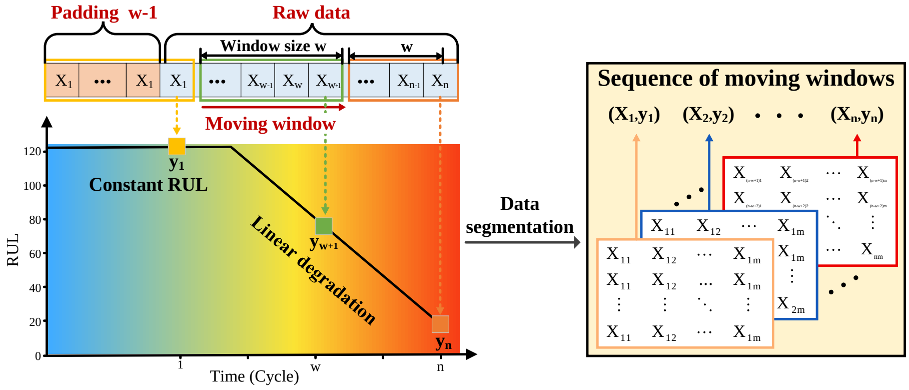


Figure 5: The illustration of data segmentation using moving windows to generate input samples

3.2. The Training of TDDN model

The TDDN model is written in Python 3.8 with Pytorch 1.9.0 deep learning framework and carried out on a high-performance computing platform with Intel(R) Xeon(R) E5-2620 v3 CPU (main frequency = 2.4 GHz). In the training procedure, 20% engines in each training dataset are randomly selected as the validation set, and the remaining engines are used as the training samples. The validation dataset is applied to evaluate the performance of the TDDN to select optimal hyperparameters. The Adam optimizer is adopted to update the weights in the network to minimize the mean square error (MSE) loss function. The maximum training epoch is preset at 200, and other hyperparameters are optimized through the hyperband algorithm.

The hyperparameters in the proposed TDDN model include: the window size of the EMA and down-sampling rate in the data transformation module, the size of convolution kernels and output length in the DSCN module, the number of neurons in the attention mechanism module, the number of neurons and the probability of dropout in regression module, and the rest are the window size of input data, the learning rate, and the batch size. The type and range of each hyperparameter are set based on our previous knowledge and listed in Table 3. 100 possible configurations are randomly sampled from search space on every subset, and then input to the hyperband algorithm to obtain the optimal hyperparameter configuration. The final chosen hyperparameter configuration of the TDDN model for four datasets is listed in Table 3. The detailed layer of the TDDN model with optimal hyperparameters for FD001 is shown in Table 4. It can be observed from Table 3 that the hyperparameter configuration of the four models shows a large difference due to the operating conditions of the machines. Hence, the hyperband algorithm is effective in optimizing hyperparameters and improving RUL prediction accuracy.

Table 3: Hyperparameters range and tuning results of four subsets for TDDN model with hyperband optimization

Hyperparameter	Value range	FD001	FD002	FD003	FD004
Dropout	$(0.2, 0.6)$	0.5	0.3	0.3	0.2
Batch size	$[16, 32, 64, 128]$	64	16	32	16
Learning rate	$\left(10^{-5}, 10^{-4}\right)$	2.05×10^{-5}	3.45×10^{-4}	4.63×10^{-5}	8.11×10^{-5}
Down-sampling rate	$[2, 3, 4]$	3	2	2	2
Window size of EMA	$(2, 10)$	8	6	9	8
Length of DSCN output	$[2, 3, 4]$	3	4	3	2
Size of convolution kernels	$[2, 3, 5]$	2	3	2	5
Number of neurons in MLP	$[32, 64, 128, 256]$	64	128	64	256
Window size of input data	$[32, 48, 64, 80]$	64	80	64	80
Number of neurons in FNN	$[32, 64, 128, 256]$	128	32	128	128

Table 4: Layer details of the proposed TDDN model for FD001 subset

Layers	Output shape	Connected to
Input layer	(64, 64, 16)	None
Identity mapping	(64, 64, 16)	Input layer
EMA	(64, 57, 16)	Input layer
Down-sampling	(64, 21, 16)	Input layer
DSCN-1	(64, 3, 64)	Identity mapping
DSCN-2	(64, 3, 64)	EMA
DSCN-3	(64, 3, 64)	Down-sampling
Concatenate layer	(64, 576)	DSCN-1, DSCN-2, DSCN-3
FC layer	(64, 1024)	Concatenate layer
Reshape layer	(64, 16, 64)	FC layer
Attention mechanism	(64, 64)	Reshape layer
FC layer	(64, 128)	Attention mechanism
Dropout	(64, 128)	FC layer
FC layer	(64, 1)	Dropout

3.3. Performance Benchmark

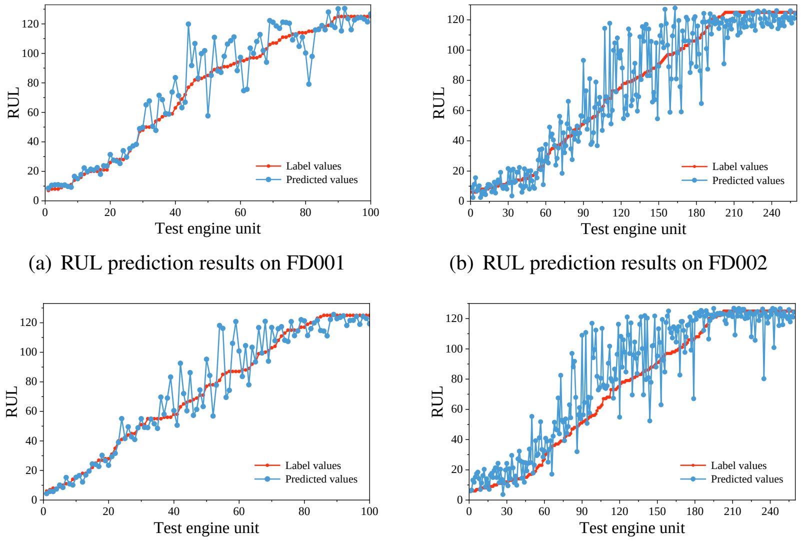
The predicted RUL results of test engines in four subsets are shown in Fig. 6. The test engine units are arranged in ascending order of the real RUL value. The mean absolute error (MAE) and standard deviation (STD) are adopted to evaluate the performance of RUL estimations with different remaining cycles to failure[29]. The MAE and STD are formulated as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |d_i| \, , \tag{9}$$

$$\text{STD} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (|d_i| - MAE)^2}, \tag{10}$$

where $d_i = \text{RUL}_i^{'} - \text{RUL}_i$ is the difference between the i-th sample predicted RUL value and true RUL value, and n is the total number of samples in the test dataset. The results are summarized in Table 5.

As illustrated in Fig. 6, the predicted RUL is almost able to closely track the real RUL, indicating the validity of the TDDN model. Besides, it is observed from Table 5 that the TDDN model performs fewer errors at the health (initial 20% RUL) and failure stage (final 20% RUL) than at the degradation stage (middle 60% RUL). The phenomenon has also been reported in previous literature[29, 30, 41]. The reason is that the degradation patterns are distinct at the health and failure stage while fuzzy at the degradation stage. The TDDN model detects the subtle difference in patterns at the degradation stage relatively poorly. A detailed analysis of degradation patterns at different stages is given in section 4.3.



(a) RUL prediction results on FD001

(b) RUL prediction results on FD002

(c) RUL prediction results on FD003

(d) RUL prediction results on FD004

Figure 6: RUL prediction results of four subsets with the TDDN model

Table 5: Comparison on MAE and STD of test engine units in different RUL on the C-MAPSS dataset

	Whole RUL		Initial 20% RUL		Middle 60% RUL		Final 20% RUL	
	MAE	STD	MAE	STD	MAE	STD	MAE	STD
FD001	7.28	8.05	6.93	6.98	9.69	9.21	1.92	1.07
FD002	10.15	9.83	8.16	8.29	14.92	11.13	4.46	3.15
FD003	7.03	7.81	5.31	4.50	9.79	9.35	1.75	1.57
FD004	10.23	11.04	6.68	9.62	13.65	13.06	5.94	5.20

Note: The whole RUL includes all test samples in the subset. The initial 20% RUL represents the true RUL of the samples between 100 and 125, the middle 60% RUL for 25 to 100, and the final 20% RUL for 0 to 25 in the subset.

Two performance metrics, namely root mean square error (RMSE) and score, are used to compare the performance of the proposed TDDN model with other methods. RMSE is defined as,

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}\,, \tag{11}$$

and score s as a function of the error is defined as,

$$s = \begin{cases} \sum_{i=1}^{n}\left(e^{-\frac{d_i}{13}} - 1\right), & d_i < 0\,, \\ \sum_{i=1}^{n}\left(e^{\frac{d_i}{10}} - 1\right), & d_i \geq 0\,. \end{cases} \tag{12}$$

Late prediction errors usually lead to severe consequences for a mechanical degradation scenario, while early prediction errors can allow maintenance in advance. Thus, the score metric in Eq. 12 penalizes late prediction errors ($d_i \geq 0$) more than early prediction errors ($d_i < 0$). Since a single outlier significantly affects the score value, it cannot thoroughly evaluate the performance of a model. Therefore, performance metrics are a combined aggregate of RMSE and score to ensure consistently accurate prediction. The TDDN model is compared with several state-of-the-art models, LSTM[42], CNN-LSTM[43], RBM-LSTM[14], HDNN[44], DIDRLSTM[16], DABA[20], TaFCN[22], GAM-CapsNet[26], and DSAN[30] on all four subsets in the C-MAPSS dataset according to the performance metrics of RMSE and score. As listed in Table 6, the TDDN model achieves the best performance in terms of RMSE and score among all the methods (except

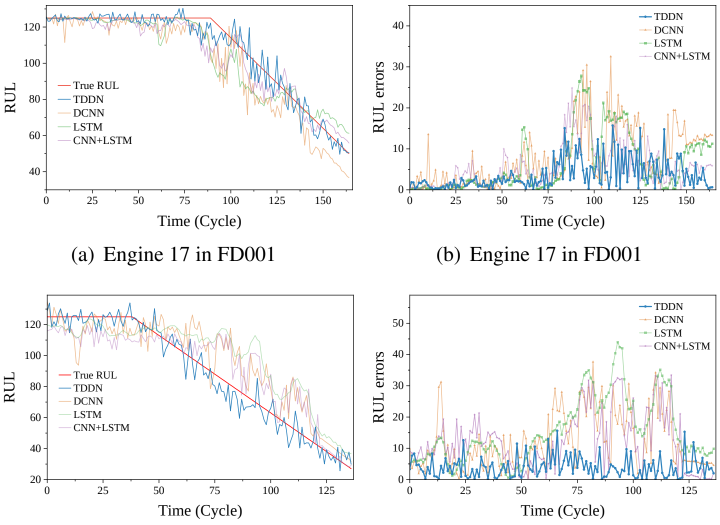
Table 6: The benchmarking of the TDDN model on the C-MAPSS dataset

Method	RMSE				Score			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
LSTM	16.74	29.43	18.07	28.40	388.7	10654	822.19	6370.6
CNN-LSTM	14.40	27.23	14.32	26.69	290	9869	316	6594
DA-TCN	11.78	16.95	11.56	18.23	229.48	1842.38	257.11	2317.32
HDNN	13.02	15.24	12.22	18.16	245	1282.42	287.72	1527.42
DIDRLSTM	16.10	28.90	16.90	32.30	213	3483	249	4123
DABA	12.25	17.08	13.39	19.86	**198**	1575	290	1741
TaFCN	13.99	17.06	12.01	19.79	336	1946	251	3671
GAM-CapsNet	12.42	17.38	11.73	19.83	273	1872	243	2490
DSAN	13.4	22.06	15.12	21.03	242	2869	479	2677
TDDN	**10.81**	**13.17**	**10.33**	**13.95**	213.28	**917.14**	**220.44**	**1156.74**

Note: The bold entities are the best results of the existing approaches.

score on FD001 subset). The results indicate that the proposed method has a superior performance in different scenarios. Especially, the TDDN model achieves noticeable improvement in complex operating conditions in comparison with the previous art-of-the-state model, with more than 13.58% reduction in RMSE and 22.48% reduction in score on FD002 subset, and 23.18% reduction in RMSE and 24.27% reduction in score on FD004 subset.

One representative engine is randomly selected to visualize the prediction performance in four subsets of FD001, FD002, FD003, and FD004. Fig. 7 shows the RUL prediction results and the corresponding absolute errors of four subsets with TDDN, DCNN, LSTM and CNN-LSTM models. It is observed that the prediction results of four models evolve around the actual RUL curve, among which the TDDN model has the smallest deviation range. Furthermore, there are some abnormal fluctuations in the degradation stage in the other three models' prediction results, while TDDN model prediction results always fluctuate within small errors. These results suggest that the TDDN model is effective to capture the trend of the degradation development and has outstanding prediction performance under various operating conditions.

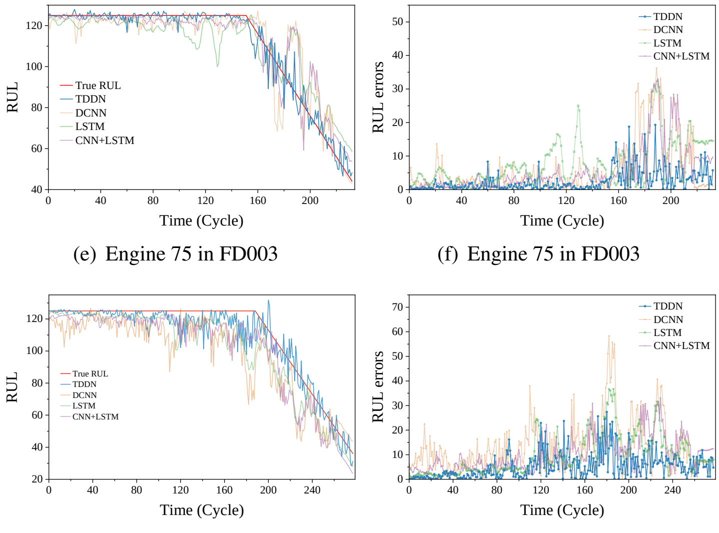


(a) Engine 17 in FD001

(b) Engine 17 in FD001

(c) Engine 253 in FD002

(d) Engine 253 in FD002



(e) Engine 75 in FD003

(f) Engine 75 in FD003

(g) Engine 35 in FD004

(h) Engine 35 in FD004

Figure 7: The predicted RUL and prediction errors on different test datasets

4. TDDN Performance Analysis

In this section, we discuss the validity of the TDDN model with an ablation study. The abstract features extracted by DSCN and the dynamic degradation patterns identified by the attention mechanism are crucial to the prediction results. We analyze the mechanism of the abstract features characterizing degradation, and illustrate how the attention map can effectively describe the degradation.

4.1. Ablation Study of TDDN Model

The TDDN model, consisting of data transformation, DSCN, and attention mechanism, has achieved outstanding RUL prediction performance. To evaluate the contribution of each module, ablation experiments are carried out to remove them from the proposed TDDN model separately. The investigation includes five variants of the TDDN model: Model-1 has no identity mapping; Model-2 has no down-sampling; Model-3 has no EMA; Model-4 has no DSCN; Model-5 has no attention mechanism. All models have the same hyperparameters as the TDDN model and run 10 trials on every subset to improve the reliability of experimental results. The average values of all models are shown in Table 7.

Table 7: Results of ablation study on four subsets

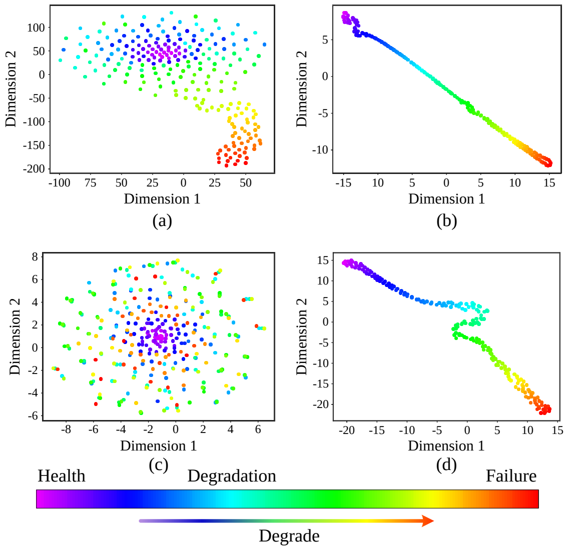
Method	RMSE				Score			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Model-1	11.59	14.92	10.75	15.92	263.42	1192.46	**217.32**	1624.09
Model-2	11.21	15.16	10.84	15.42	249.12	1412.87	238.47	1792.59
Model-3	**10.73**	14.43	11.52	14.95	231.28	1236.21	251.82	1423.58
Model-4	12.39	18.79	12.01	19.73	298.27	2104.85	284.25	2493.28
Model-5	11.92	16.54	12.37	18.89	283.56	1835.24	304.85	2073.55
TDDN	10.81	**13.17**	**10.33**	**13.95**	**213.28**	**917.14**	220.44	**1156.74**

It can be seen from Table 7 that the DSCN and attention mechanism are the two vital modules to improve model performance. The model-4 has the worst performance among all models, which lacks the ability to learn abstract features from the streaming sensor data. Similarly, the model-5 cannot capture dynamic degradation patterns and has relatively large prediction errors. Particularly, when the engines in FD002 and FD004 subsets have complex operating conditions, their degradation patterns are more difficult to be extracted from the streaming sensor data. Therefore, the DSCN or attention mechanism solely is not well adapted for complex operating conditions and performs poorly on FD002 and FD004 subsets compared to the TDDN model. Moreover, the model-1, model-2, and model-3 have smaller RMSE errors and lower score values than model-4 and model-5, whose performance improvements are attributed to the combination of DSCN and attention mechanism.

Furthermore, the TDDN model achieves small improvements compared to model-1, model-2, and model-3, indicating that the data transformation module help extract degradation patterns to enhance prediction performance. Hence, through systematically integrating data transformation, DSCN, and attention mechanism, the TDDN model achieves superior performance in all four subsets, clearly showing each module's validity.

4.2. Abstract Feature and Degradation Trajectory

The abstract feature corresponding with dynamic degradation patterns is automatically extracted from the moving window $\mathbf{X}_j$ by DSCN. After DSCN, the moving window $\mathbf{X}_j$ is mapped into abstract feature $\mathbf{H}_j$. To understand the feature learning capability of DSCN, t-stochastic neighbor embedding(t-SNE)[45], a nonlinear technique for unsupervised dimensionality reduction, is applied to visualize abstract features to gain more insights while preserving much of local structure in the high-dimension space. Fig. 8 illustrates the t-SNE projection of sequence of moving windows $[\mathbf{X}_1 \cdots \mathbf{X}_j \cdots \mathbf{X}_n]$ and sequence of abstract features $[\mathbf{H}_1 \cdots \mathbf{H}_j \cdots \mathbf{H}_n]$ extracted by DSCN in an engine degradation. Figs. 8 (a) and 8(c) show that the t-SNE dots of sequence of moving windows are highly mixed and do not form any degradation trajectory, because the original streaming sensor data are highly fluctuating signals in Fig. 4. In contrast, it is observed in Figs. 8 (b) and 8(d) that the t-SNE dots of sequence of abstract features form a clear degradation trajectory, which is critical to improving the prediction performance of the TDDN model.



(a)

(b)

(c)

(d)

Figure 8: The t-SNE visualization of (a) the sequence of moving windows for engine 38 in FD001 training subset, (b) the sequence of abstract features for engine 38 in FD001 training subset, (c) the sequence of moving windows for engine 38 in FD004 training subset, (d) the sequence of abstract features for the engine 38 in FD004 training subset. Each point represents a moving window or abstract feature extracted from the corresponding moving window. The color of points are corresponding to RUL. The hotter the color is, the closer the engine is to the final failure.

The abstract feature is a nonlinear mapping of the streaming sensor data. To understand the degradation trajectories in detail, we select eight abstract feature components in $\mathbf{H}_j$ whose development trajectories are shown in Fig 9. Compared to the plots of the streaming sensor data in Fig. 4, the trajectories of eight abstract feature components have fewer fluctuations and are more consistent with the trend of the RUL curve. DSCN works very well as a specific filter similar to a band-pass filter in frequency domain and effectively removes fluctuations in the streaming sensor data. Therefore, the abstract features can be accurately extracted from the streaming sensor data for the description of dynamic degradation patterns and RUL prediction.

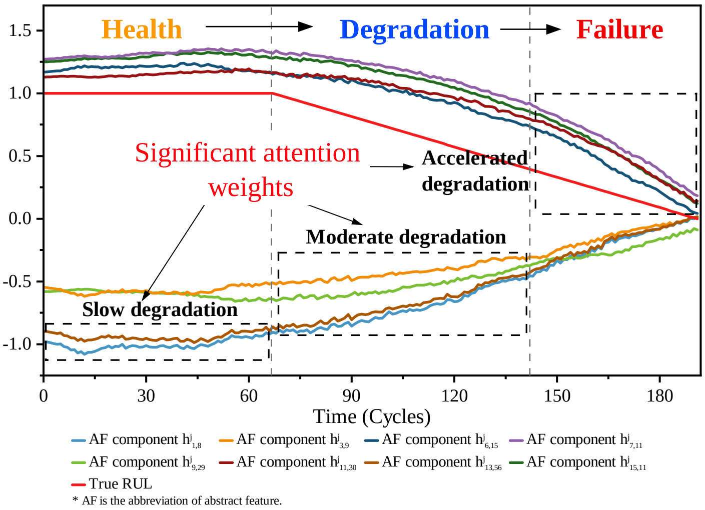


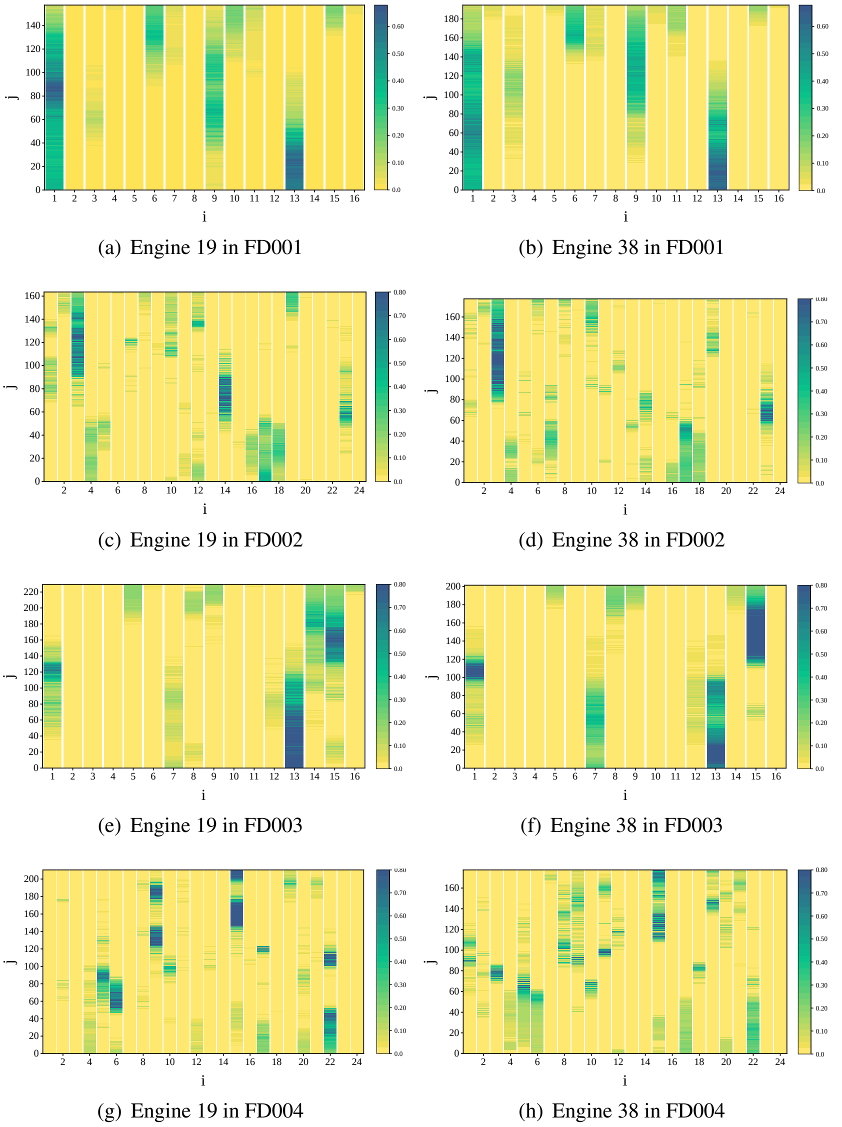
Figure 9: The trajectory of the k-th element in i-th abstract feature component $\mathbf{h}^j_{i,k}$ in the j-th moving window $\mathbf{X}_j$ for engine 38 in FD001 training subset.

4.3. Attention Maps and Dynamic Degradation Patterns

The attention mechanism generates attention weights $\left[\lambda^j_1, \cdots, \lambda^j_i, \cdots, \lambda^j_m\right]$ for the abstract features $\mathbf{H}_j$ in each moving window $\mathbf{X}_j$. The evolution of attention weights of abstract features can describe the degradation development effectively and compactly. The attention map, the temporal concatenation of attention weights corresponding with each moving window, is used to visualize the evolution of attention weights. The attention maps essentially reveal dynamic degradation patterns. In other words, the attention map is an encoding of dynamic patterns embedded in the streaming sensor data. It can serve as a fingerprint that characterizes the degradation of an engine under different operating conditions. To illustrate the encoding, the attention maps of engine 19 and engine 38 in FD001, FD002, FD003, and FD004 four subsets are plotted in Fig. 10. As shown in Table 1, the four subsets have two kinds of operating conditions, so it can be found from Fig. 10 that the engines demonstrate two main dynamic degradation patterns. Specifically, Figs. 10(a), 10(b), 10(e) and 10(f) show the attention maps of engine 19 and engine 38 in FD001 and FD003 subsets that share the similar fingerprint of degradation. On the other hand, the attention maps of engine 19 and engine 38 in FD002 and FD004 subsets also share the similar fingerprint of degradation in Figs. 10(c), 10(d), 10(g) and 10(h) show.

The dynamic degradation patterns are represented in terms of the attention maps at the three degradation stages of health, degradation, and failure. Engine 38 in FD001 in Fig. 10(b) and FD002 in Fig. 10(d) are used to elucidate how attention maps are served as fingerprints to identify the degradation stages. For example, according to the attention map in Fig. 10(b), the degradation development of engine 38 in FD001 subset is roughly divided into health stage roughly during the first 69 cycles, degradation stage roughly from cycle 70 to cycle 154, and failure stage roughly during the last 40 cycles. Abstract feature components 1 and 13 have the largest attention weights at the health stage. The attention weights of the remaining abstract feature components are close to zero. Therefore, the attention weights for the two abstract feature components 1 and 13 form the unique dynamic patterns at the health stage. The attention weights for abstract feature components 1 and 13 persist at the degradation stage while those for abstract feature components 3 and 9 increase. The attention weights for four abstract feature components 1, 3, 9, and 13 form the unique dynamic patterns at the degradation stage. The attention weights for abstract feature components 3 and 13 disappear at the failure stage. In contrast, the attention weights for abstract feature components 1 and 9 persist, and the attention weights for abstract feature components 6, 11, and 15 appear to be more significant. The attention weights for the five abstract feature components 1, 6, 9, 11, and 15 form the unique dynamic patterns for the failure stage. Hence, the set of abstract feature components with significant attention weights at the three degradation stages can serve as a fingerprint to describe the degradation development. (1, 13) is the set of abstract feature components with significant attention weights for the health stage, (1, 3, 9, 13) for the degradation stage, and (1, 6, 9, 11, 15) for the failure stage. Therefore, the attention map effectively encodes the dynamic degradation patterns at the three degradation stages. It is also found that the abstract feature components that have significant attention weights at the three stages have shown different degradation trends in Fig. 9, which indicates that the TDDN model can capture the crucial degradation features and dynamic degradation patterns.

Compared to Fig. 10(b), Fig. 10(d) shows the degradation development of engine 38 in FD002 is more complicated and can still be roughly divided into health stage roughly during the



(a) Engine 19 in FD001

(b) Engine 38 in FD001

(c) Engine 19 in FD002

(d) Engine 38 in FD002

(e) Engine 19 in FD003

(f) Engine 38 in FD003

(g) Engine 19 in FD004

(h) Engine 38 in FD004

Figure 10: The plots of the eight attention maps show the evolution of the attention weight λ_i^j of abstract feature $\mathbf{h}_i^j$ in moving window $\mathbf{X}_j$ of Engine 19 and Engine 38 in FD001, FD002, FD003 and FD004 four subsets. In each plot, x-axis and y-axis denote the indices of abstract feature components and moving window accordingly. Attention weights are visualized according to the colorbars on the right side.

first 52 cycles, degradation stage roughly from cycle 53 to cycle 142, and failure stage roughly during the last 35 cycles. Since the operating conditions are much more complicated in FD002, the attention map at the degradation stage demonstrates two distinct dynamic patterns for the first part from cycle 53 to cycle 92 and the second part from cycle 92 to cycle 142, respectively. Therefore, it is observed from Fig. 10(d) that (4, 7, 16, 17, 18) is the set of abstract feature components with significant attention weights for the health stage, (3, 14, 23) and (3, 12, 19) for the degradation stage, and (2, 3, 6, 8, 10) for the failure stage.

It is not easy to extract the dynamic degradation patterns from the fluctuating streaming sensor data in different operating conditions. However, by combining DSCN and attention mechanism, the TDDN model can capture dynamic degradation patterns in terms of the attention maps. Thus, the TDDN model with attention-based dynamic degradation pattern significantly improves the accuracy of RUL predictions.

5. Conclusions

In this paper, a novel data-driven method called TDDN is proposed for the RUL prediction of machines. In the TDDN, multiple data transformation techniques and DSCNs are parallel adopted to extract abstract features from streaming sensor data. The abstract features are further fed to the attention mechanism to capture dynamic degradation patterns. In addition, the hyperband algorithm is introduced to tune the hyperparameters of the TDDN model. The comparative experiments are performed on a benchmark C-MAPSS dataset to evaluate the performance of the TDDN model. Compared with the performance metrics of existing deep learning approaches on the C-MAPSS dataset, the TDDN model achieves the best RUL prediction results. Furthermore, the ablation studies and discussions are conducted to analyze the validity of each module. The results indicate that the abstract features extracted by DSCN are capable of eliminating noise-related information and characterizing degradation trajectory. The attention mechanism encodes dynamic degradation patterns into the attention map which can serve as a degradation fingerprint to accurately describe the degradation development of machines. Based on those, the TDDN model achieves a superior RUL prediction performance.

It should be pointed out that the TDDN model outputs point predictions without uncertainty quantification of RUL prediction results. The performance of model is easily affected by uncertain factors in operating conditions. In future work, we will extend the TDDN model by replacing dropout with Monte Carlo dropout inference in the FNN for quantifying the output uncertainties or introducing other uncertainty quantification methods, thereby promoting the reliability and robustness of RUL prediction.

References

[1] B. Rezaeianjouybari, Y. Shang, Deep learning for prognostics and health management: State of the art, challenges, and opportunities, Measurement 163 (2020) 107929.

[2] Y. Hu, X. Miao, Y. Si, E. Pan, E. Zio, Prognostics and health management: A review from the perspectives of design, development and decision, Reliability Engineering & System Safety 217 (2022) 108063.

[3] Y. Wen, M. F. Rahman, H. Xu, T.-L. B. Tseng, Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective, Measurement 187 (2022) 110276.

[4] R. Gong, J. Li, C. Wang, Remaining useful life prediction based on multi-sensor fusion and attention tcn-bigru model, IEEE Sensors Journal (2022).

[5] J. Wang, Z. Li, G. Bai, M. J. Zuo, An improved model for dependent competing risks considering continuous degradation and random shocks, Reliability Engineering System Safety 193 (2020) 106641.

[6] G. Jiang, W. Zhou, Q. Chen, Q. He, P. Xie, Dual residual attention network for remaining useful life prediction of bearings, Measurement (2022) 111424.

[7] L. Liu, X. Song, K. Chen, B. Hou, X. Chai, H. Ning, An enhanced encoder–decoder framework for bearing remaining useful life prediction, Measurement 170 (2021) 108753.

[8] C. Ordóñez, F. S. Lasheras, J. Roca-Pardinas, F. J. de Cos Juez, A hybrid arima–svm model for the study of the remaining useful life of aircraft engines, Journal of Computational and Applied Mathematics 346 (2019) 184–191.

[9] Z. Chen, Y. Li, T. Xia, E. Pan, Hidden markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy, Reliability Engineering & System Safety 184 (2019) 123–136.

[10] Z. Liu, Y. Cheng, P. Wang, Y. Yu, Y. Long, A method for remaining useful life prediction of crystal oscillators using the bayesian approach and extreme learning machine under uncertainty, Neurocomputing 305 (2018) 27–38.

[11] F. Wang, X. Liu, G. Deng, X. Yu, H. Li, Q. Han, Remaining life prediction method for rolling bearing based on the long short-term memory network, Neural Processing Letters 50 (3) (2019) 2437–2454.

[12] A. Elsheikh, S. Yacout, M.-S. Ouali, Bidirectional handshaking lstm for remaining useful life prediction, Neurocomputing 323 (2019) 148–156.

[13] D. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, Z. Sun, Well production forecasting based on arima-lstm model considering manual operations, Energy 220 (2021) 119708.

[14] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, H. Zhang, Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture, Reliability Engineering & System Safety 183 (2019) 240–251.

[15] Y. Wang, J. Zhao, C. Yang, D. Xu, J. Ge, Remaining useful life prediction of rolling bearings based on pearson correlation-kpca multi-feature fusion, Measurement 201 (2022) 111572.

[16] S. Fu, Y. Zhang, L. Lin, M. Zhao, S.-s. Zhong, Deep residual lstm with domain-invariance for remaining useful life prediction across domains, Reliability Engineering & System Safety 216 (2021) 108012.

[17] Z. Shi, A. Chehade, A dual-lstm framework combining change point detection and remaining useful life prediction, Reliability Engineering & System Safety 205 (2021) 107257.

[18] J. Bae, Z. Xi, Learning of physical health timestep using the lstm network for remaining useful life estimation, Reliability Engineering & System Safety 226 (2022) 108717.

[19] H. Ding, L. Yang, Z. Cheng, Z. Yang, A remaining useful life prediction method for bearing based on deep neural networks, Measurement 172 (2021) 108878.

[20] B. Yang, R. Liu, E. Zio, Remaining useful life prediction based on a double-convolutional neural network architecture, IEEE Transactions on Industrial Electronics 66 (12) (2019) 9521–9530.

[21] X. Du, W. Jia, P. Yu, Y. Shi, S. Cheng, A remaining useful life prediction method based on time–frequency images of the mechanical vibration signals, Measurement 202 (2022) 111782.

[22] L. Fan, Y. Chai, X. Chen, Trend attention fully convolutional network for remaining useful life estimation, Reliability Engineering & System Safety (2022) 108590.

[23] L. Zheng, Y. He, X. Chen, X. Pu, Optimization of dilated convolution networks with application in remaining useful life prediction of induction motors, Measurement 200 (2022) 111588.

[24] Y. Cao, Y. Ding, M. Jia, R. Tian, A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings, Reliability Engineering & System Safety 215 (2021) 107813.

[25] X. Li, V. Krivtsov, K. Arora, Attention-based deep survival model for time series data, Reliability Engineering & System Safety 217 (2022) 108033.

[26] C. Zhao, X. Huang, Y. Li, S. Li, A novel remaining useful life prediction method based on gated attention mechanism capsule neural network, Measurement 189 (2022) 110637.

[27] Z. Zhang, W. Zhang, K. Yang, S. Zhang, Remaining useful life prediction of lithium-ion batteries based on attention mechanism and bidirectional long short-term memory network, Measurement 204 (2022) 112093.

[28] Y. Zhao, Y. Wang, Remaining useful life prediction for multi-sensor systems using a novel end-to-end deep-learning method, Measurement 182 (2021) 109685.

[29] D. Xu, H. Qiu, L. Gao, Z. Yang, D. Wang, A novel dual-stream self-attention neural network for remaining useful life estimation of mechanical systems, Reliability Engineering & System Safety 222 (2022) 108444.

[30] J. Xia, Y. Feng, D. Teng, J. Chen, Z. Song, Distance self-attention network method for remaining useful life estimation of aeroengine with parallel computing, Reliability Engineering & System Safety (2022) 108636.

[31] Z. Cui, W. Chen, Y. Chen, Multi-scale convolutional neural networks for time series classification, arXiv preprint arXiv:1603.06995 (2016).

[32] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436–444.

[33] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[34] W. Wang, Z. Chen, H. Hu, Hierarchical attention network for image captioning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8957–8964.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[36] H. Mo, L. L. Custode, G. Iacca, Evolutionary neural architecture search for remaining useful life prediction, Applied Soft Computing 108 (2021) 107474.

[37] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-Tzur, M. Hardt, B. Recht, A. Talwalkar, A system for massively parallel hyperparameter tuning, Proceedings of Machine Learning and Systems 2 (2020) 230–246.

[38] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, The Journal of Machine Learning Research 18 (1) (2017) 6765–6816.

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

[40] F. O. Heimes, Recurrent neural networks for remaining useful life estimation, in: 2008 international conference on prognostics and health management, IEEE, 2008, pp. 1–6.

[41] J. Xia, Y. Feng, C. Lu, C. Fei, X. Xue, Lstm-based multi-layer self-attention method for remaining useful life estimation of mechanical systems, Engineering Failure Analysis 125 (2021) 105385.

[42] C.-S. Hsu, J.-R. Jiang, Remaining useful life estimation using long short-term memory deep learning, in: 2018 ieee international conference on applied system invention (icasi), IEEE, 2018, pp. 58–61.

[43] Z. Wu, S. Yu, X. Zhu, Y. Ji, M. Pecht, A weighted deep domain adaptation method for industrial fault prognostics according to prior distribution of complex working conditions, IEEE Access 7 (2019) 139802–139814.

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

[45] I. Remadna, L. S. Terrissa, Z. Al Masry, N. Zerhouni, Rul prediction using a fusion of attention-based convolutional variational autoencoder and ensemble learning classifier, IEEE Transactions on Reliability (2022).