Deep & Attention : A Self-Attention based Neural Network for Remaining Useful Lifetime Predictions

Yuanjun Liu
Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence,
University of Chinese Academy of Sciences
Beijing, China
e-mail: liuyuanjun2018@ia.ac.cn

Xingang Wang*
Institute of Automation, Chinese Academy of Sciences
Beijing, China
e-mail: xingang.wang@ia.ac.cn

2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE) | 978-1-6654-1489-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMRE51691.2021.9384841

***Abstract*—The remaining useful lifetime (RUL) of assets plays a critical role in machine prognostics and health management (PHM). Accurate RUL predictions can reduce losses caused by equipment faults. Most existing data-driven PHM methods rely on long short-term memory (LSTM) networks to model the relationship of time series data and RUL. However, because of the sequential nature of LSTM, it is not conducive to parallel computing. Herein, we propose the Deep & Attention Network, which uses a combination of convolutional neural networks and Attention methodologies instead of LSTM. In the proposed Deep & Attention Network, the Attention component models the temporal property, while the Deep component learns the effect of noise data. Experiments on NASA's Commercial Modular Aero-Propulsion System Simulation datasets demonstrate that the proposed network achieves a level of performance similar to that of other state-of-the-art RUL prediction models. Moreover, compared with LSTM-based methods, our Self-Attention-based method is conducive to parallel computing.**

***Index Terms*—C-MAPSS; Deep learning; Remaining useful life; Attention mechanism**

I. Introduction

Traditional planned machine maintenance is performed regardless of whether or not a system fails. However, this maintenance strategy has inevitable shortcomings. Maintenance intervals that are too short can result in unnecessary maintenance procedures. Conversely, if the intervals are too long, failures may occur between two maintenance procedures and disrupt the normal operation of the system. In response to the problems of under-maintenance or over-maintenance in traditional maintenance strategies, researchers have proposed a predictive maintenance strategy called prognosis and health management (PHM) [1]. Engineering practice indicates that PHM technology can effectively decrease the probability of equipment failure and decrease maintenance costs, particularly in areas with high safety and reliability requirements; thus, PHM can significantly reduce the amount of system downtime and increase the success rate of tasks [2]. The main problem encountered with PHM is that of accurate assessment of the system health and prediction of the remaining useful life (RUL) of equipment based on massive state monitoring data [3]. In PHM, remaining useful life (RUL) relates to the amount of time left before a piece of equipment cannot perform its intended function. Accurate RUL prognostics enable the interested parties to assess an equipment's health status and to plan future maintenance actions, e.g. logistics of personnel, spare parts and services [31]. In past years, data-driven methods have been developed to model the nonlinear, complex and multi-dimensional systems without prior hnowledge. In this paper, we propose a data-driven method for remaining useful life predictions.

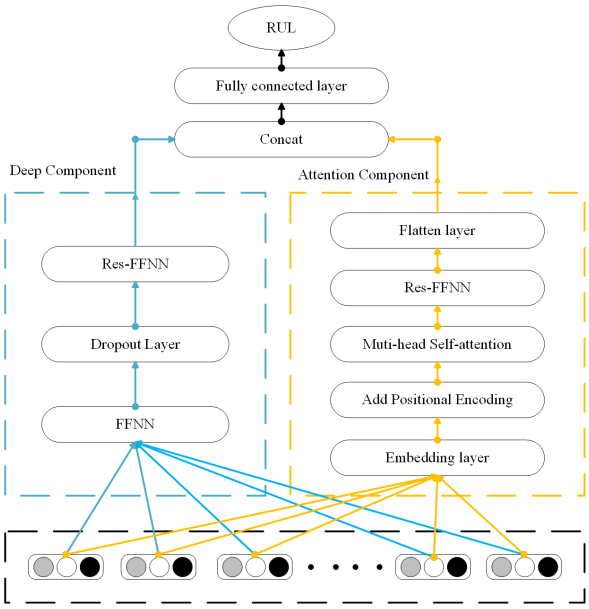

Fig. 1. Sketch view of the Deep & Attention model: The Deep and Attention components share the same input data. The Attention component learns the long-term relationship, while the Deep component removes the timing relationship, which learns the effect of noise data.

Deep learning is a branch of machine learning in which deep abstract feature extraction and complex nonlinear relationship expression are achieved by stacking deep neural networks. In many fields, deep learning has achieved far better results than traditional machine learning methods. In recent years, numerous deep learning studies have yielded state-of-the-art results in manufacturing tasks [4]. For example, an auto-encoder-based method was able to identify the starting time of

978-0-7381-3205-1/21/$31.00 ©2021 IEEE

a bearing failure accurately [5]. In research on the deep belief network (DBN) class methods, a DBN-based method was used to predict the material removal rate during polishing [6]. In particular, in RUL predictions for turbofan engine degradation data, long short-term memory (LSTM) networks significantly outperformed other methods [7]–[9]. Among them, variants of the LSTM model achieved the best performance in tests with the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) datasets. However, the LSTM model cannot parallelize calculations very well and consumes substantial computing resources in actual deployment. To solve these problems, we adopted an Attention mechanism based method to model the time series; this solution is more conducive to parallel computing and does not cause a loss of model accuracy. To the best of our knowledge, this is the first time that a LSTM model has been replaced with an Attention mechanism based method for RUL estimation.

An analysis of data from the C-MAPSS datasets revealed that data obtained from the sensor were more stable over the long term. Owing to the influence of noise, the trend of change is not obvious in the short term. In this situation, if the data in one time period are exchanged with data from an adjacent time period, the impact on the prediction results of the model will be minimal. There is a strict timing relationship between time series data; however, in some time periods, this timing relationship is overwhelmed by noise. To address this problem, we propose the Deep & Attention Network. This model is divided into a Deep component and an Attention component. The Attention component models the timing relationship of the data, while the Deep component accounts for the influence of noise and removes the time relationship of the input data. In tests with the C-MAPSS datasets, we demonstrate that the proposed method can achieve state-of-the-art results without using the LSTM model.

The main contributions of this paper are summarized as follows.

We propose a new model, the Deep & Attention Network, that can consider the time relationship of the input data and the influence of noise simultaneously. The Attention component models the timing relationship of the data, while the Deep component accounts for the influence of noise and removes the time relationship of the input data.

Without using LSTM, our proposed model achieved state-of-the-art performance in tests with the C-MAPSS datasets. Moreover, in practical applications, our model is more conducive to parallel computing.

II. Related Work

In machine Prognostics and Health Management (PHM), machine learning methods have shown better performance without redundant hard work of domain expert. Due to a large number of data with labels supplied for the training stage, the machine learning methods are also regarded as data-driven methods. In this section, we review recent studies for the data-driven methods of PHM. In the prognostics literature, several traditional machine learning methods have been proposed to estimate the RUL of industry machines. Linear regression is introduced to diagnose the fault of a rolling bearing [10]. The support vector regression (SVR) is employed to predict the RUL of assets [11]. Further, Zio et al. presented a similarity-based approach for prognostics of the remaining useful life of a system [12].

In recent years, deep neural networks have been widely used in various fields, e.g. object recognition, image classification [17] and speech recognition [18]. Neural network methods with back propagation have been used in the prediction of a ball bearing's RUL [13]. An artificial neural network approach is proposed for the accurate equipment RUL prediction [14], [15]. Although neural networks-based approaches have outperformed the traditional machine learning methods, these methods are especially affected by handcrafted features and various specified parameters. Zhang et al. proposed a multi-objective deep belief networks ensemble method to estimate the remaining useful life of the system [16].

Deep convolutional neural networks have made continuous breakthroughs in image classification [19]–[21]. Especially, deep residual learning methods have been proposed to lighten the problem of vanishing gradients and the method could also perform better than flatten structure [22]. Then, CNN is employed to process multivariate time series signals from sensor, due to ability to model spatial information of structed data. Li et al. proposed convolutional neural network based regression approach to estimate the RUL for the first time [23]. Different from the existing CNN structure for computer vision, their convolution and pooling filters are applied along the temporal dimension over the multi-channel sensor data. Time window approach is employed for sample preparation in order for better feature extraction by DCNN and no accurate physical model is required [9]. Li et al. presents 2-D CNN to extract time-frequency domain information for estimating the RUL of machine. These results show that DCNN could compress time series data to high dimensional presentation for RUL prediction [24].

More recently, Recurrent Neural Networks (RNN) have shown better performance than other artificial neural networks. Although time series could be modeled by RNN, it has difficulty to learn long-term dependencies [25]. To address the problem of vanishing gradients, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been proposed [26], [27]. Yuan et al. have compared LSTM to different variants of RNN e.g. traditional RNN, Gated Recurrent Unit LSTM (GRU-LSTM) and AdaBoost-LSTM, and LSTM could outperform others in the RUL estimation task [28]. Zheng et al. presents a method of LSTMs follow by Feed Forward Neural Networks(FFNNS), which improve the accuracy for the prediction of RUL [8]. Wu et al. have also prove that LSTM is a natural fit for the RUL predictions [29]. What's more, they presents a similar result that LSTM could outperform other variants of RNNs. In real-life industry application, high-quality training data with labels might be challenging to acquire. Therefore, the combination of supervised and unsupervised (semi-supervised) learning has been

introduced in RUL prediction [30]. Additionally, a Genetic Algorithm (GA) approach, due to the diverse amount of hyper-parameters, is proposed in the training procedure. Costa et al. have combined the LSTM network with global attention layers to learn RUL relationships, and the method show a state-of-the-art performance [31].

Although LSTM could ease the problem of vanishing gradients, it might be invalid for modeling long-term dependencies. In machine translation, attention mechanism has made great breakthrough [32]. Then, attention mechanism often placed LSTM among RNN realizations in NLP. In clinical prediction tasks, an architecture solely based on attention mechanisms has made remarkable sucess [33]. Compared to LSTMS, attention-based architecture could be implemented in parallel more handiy. Besides performance gain, Galassi et al. have shown that attention mechanisms could be used to analyse sequence input by the attention weights [34]. In this work, we propose a novel architecture without RNNS to learning the engine degradation. Similar to most data-driven method, the attention based model can be trained on the dataset without any feature engineering and expertise.

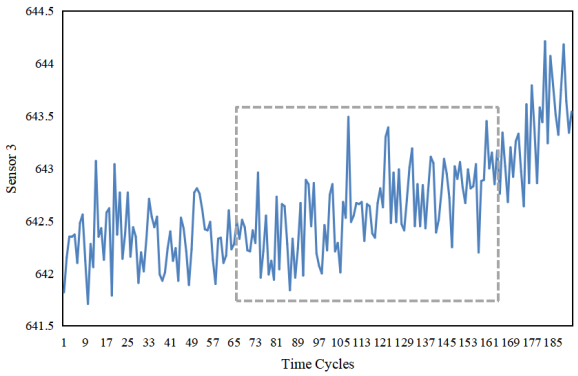

Fig. 2. Time series collected from sensor 3 indicate that data increase from start to failure. However, in the dashed box, the data trend is hidden locally in noise.

III. Method

In this section, we provide a detailed introduction to the Deep & Attention Network. First, we introduce the task of RUL prediction and define its symbols. Second, we introduce the structure of the entire model. Finally, we explain the core part of the model in detail.

A. Problem and symbol definition

The dataset is defined as $\{x_i, y_i\}_{i=1}^{N}$, where N represents N samples, $x_i \in R^{T_w \times f_{input}}$ is the input of sample i, and $y_i \in R$ is the RUL value of sample i. In general, the RUL prediction task needs to predict the RUL value at a certain time point t_i. Therefore, x_i is composed of the sampling data of the nearest T_w samples at time t_i, denoted by $\{x_i = (x_t)_{t=t_i-T_w+1}^{t_i}\} \in R^{T_w \times f_{input}}$, where f_{input} is the feature number. The RUL estimation task is defined as

$$\hat{y_i} = F_{RUL}(x_i, \{W_{RUL}\}), \tag{1}$$

where x_i is the input data, $\hat{y_i}$ is the estimated value of RUL, and the function $F_{RUL}(x_i, \{W_{RUL}\})$ is expressed as a mapping function.

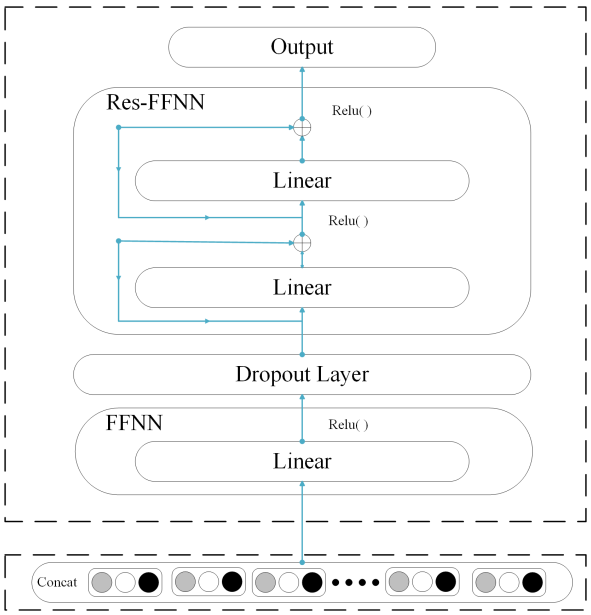

Fig. 3. Structure of the Deep component is similar to that of ResNet. The first layer resizes the input data into a fixed dimension, and the subsequent layers add the output to the input. To prevent overfitting, a dropout layer is used after the first layer.

B. Model structure

The input data, x_i, are multivariate time series data, and each feature is a time series collected from different parts of the machine. As shown in Fig. 2, each feature has a stable trend of change from a long-term perspective; however, owing to the influence of noise, the change in features over the short term is not obvious.

As shown in Fig. 1, the Attention component is used to model the long-term trend of the input data. It maps the input x_i of length T_w to the Embedding vector e_i of length T_w through the convolutional neural network (CNN) methodology, where the Embedding vector is denoted as $\{(e_t)_{t=t_i-T_w+1}^{t_i}\} \in R^{T_w \times f_{emb}}$, and f_{emb} is the dimension of the Embedding space. In the task of RUL estimation, the current state-of-the-art methods are based on LSTM models. Because of the sequential nature of LSTM, it is not conducive to parallel computing. In this paper, we propose a method based on Self-Attention to model the timing relationship of the input sequence. The final output of the Self-Attention component is a d_{atten} dimensional vector, which is represented as

$$h_i^{atten} = F_{atten}(e_i, W_{atten}) \tag{2}$$

where $h_i^{atten} \in R^{d_{atten}}$ represents the output of the Attention component.

The Deep component is used to model the short-term noise effects; thus, a feed-forward neural network is used. Because the trend of changes between adjacent time series is covered by

noise during a relatively short period of time, the output of the model will not change significantly if the adjacent time series are exchanged. As shown in Fig. 1, to model the weak timing relationship between the input time series, we concatenate each time t of the input multivariate time series x_i together. The variable, i.e., the eliminated time series relationship, is denoted as

$$x_i^{flatten} = concat(\{x_t\}_{t=t_i-T_w+1}^{t_i}) \tag{3}$$

Then, we place $x_i^{flatten}$ into the feed-forward neural network to obtain the final output vector, h_i^{dnn}, where $h_i^{dnn} \in R^{d_{dnn}}$.

The output vectors, h_i^{atten} and h_i^{dnn}, are spliced together, and the estimated RUL value is obtained after linear weighting as follows:

$$\hat{y} = W_o concat(flat\_concat(h_i^{atten}), h_i^{dnn}) + b_o \ \hat{y} \in R \tag{4}$$

where the flat_concat function expands the h_i^{atten} along with the time dimension and performs concatenation to obtain the output vector. The loss function is

$$Loss_i = \| \hat{y}_i - y_i \|^2 \tag{5}$$

C. Deep component

As shown in Fig. 3, the Deep component is a feed-forward neural network. First, the input multivariate time series are flattened to obtain $x_i^{flatten}$, which concatenates x_i together. This operation eliminates the time dimension of the input data, thereby weakening the influence of noise on the short-term changing trends. Then, the flattened data are fed into the multi-layer stacked neural network to obtain the high-dimensional vector representation of the Deep component. Each layer of the neural network is represented as follows:

$$h_{i+1} = RELU(W_l h_l + b_l + h_l) \tag{6}$$

To accelerate model convergence, the residual feed forward neural network (Res-FFNN), which is similar to ResNet [22], is introduced in the output of the model.

D. Embedding component

As shown in Fig. 4, the Embedding component maps a multivariate time series into a higher-dimensional Embedding space $\{(e_t)_{t=t_i-T_w+1}^{t_i}\} \in R^{T_w \times f_{emb}}$, thereby obtaining high-level semantic representations between different features. We use multi-stacked 1D convolutional layers to slide in the time (t) dimension, thereby obtaining the high-order semantic representation, e_t, of each time t. The Embedding component includes two types of CNN structures. The first type is a single-layer CNN structure with a filter size of three. This structure changes the dimensionality of the output Embedding of each layer. The second type is the CNN Res Block, which is similar to ResNet; it includes two CNN layers with a filter size of one. This structure ensures that the model converges faster while obtaining higher-level semantics. As the first-layer structure of the Attention component, the Embedding component preserves the time series relationship of the data on the premise of obtaining high-level semantic representation.

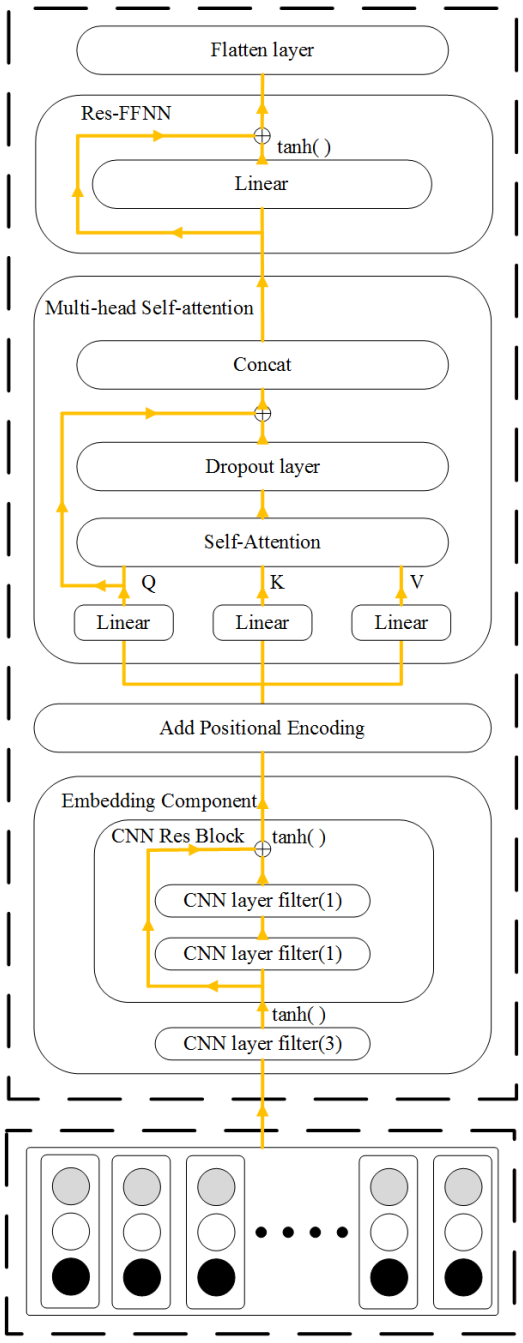

Fig. 4. Attention component mainly contains the Embedding and Self-Attention components. The Embedding component learns the timing relationship of the input data; thus, it consists of several stacked CNN layers. Then, positional encoding is added to the output of the Embedding component. Finally, the vectors are fed into the Self-Attention component to model the long-term dependence.

E. Multi-Head Self-Attention component

As shown in Fig. 4, the Self-Attention component models timing relationships and can capture the long-term trends of features. Compared with LSTM, the Self-Attention component can model longer time series and can parallelize calculations better. Before inputting the Embedding vector into the Self-Attention component, we added positional encoding to the Embedding vector [32], thus adding the time information between different Embedding vectors.

In general, the Self-Attention component can be defined as the mapping of a query q and a set of key-value pairs into an output o. For Embedding at each time t, we calculate the Attention weight using the inner product between q_t and the keys. We then use the Attention weight to perform a weighted summation of the values to obtain the output vector, o. The Self-Attention component is defined as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

where Q, K, and V are the formats of the linear mapping of e_i and are defined as $Q = W^q e_i$, $K = W^k e_i$, and $V = W^v e_i$, respectively.

Multi-Head Self-Attention calculates the Attention vectors of multiple subspaces and then concatenates these Attention vectors together. This approach can model the correlation of different embeddings from multiple angles and enhance the performance of the model. The Multi-Head Self-Attention function is defined as follows:

$$Multi\text{-}Head(Q, K, V) = Concat(head_1, \ldots, head_h) \tag{8}$$

where $head_i$ is defined as follows:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{9}$$

The Multi-Head function maps the Embedding to the Attention vector. Then, the Attention vector, a_i, is fed into the single-layer Res-FFNN to obtain h_i^{atten}. The function is defined as follows:

$$h_i^{atten} = tanh(W_{ffn}a_i + b_{ffn} + a_i) \tag{10}$$

Finally, we expand and concatenate the h_i^{atten} along the time dimension to obtain the final output of the Attention component. It is worth noting that the Deep & Attention Network uses an Attention mechanism to model time series, which is more conducive to parallel computing than the RNN model.

IV. Experiment

In this section, we compare Deep & Attention Network with the state-of-the- art on C-MAPSS Datasets. The results show that Deep & Attention Network can achieve the state-of-the-art performance while being more conducive to parallel computing.

A. C-MAPSS Datasets

Deep & Attention Network is evaluated in the benchmark Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) datasets. C-MAPSS datasets include four different datasets. The data in datasets includes 21 sensors and 3 operational settings. Each of the four datasets splits several degradation engines into training and testing data. In addition, each dataset contains run-to-failure information collected under various operating conditions and fault modes.

The Engine is considered to start in a random initial state, and every data recorded is in a healthy state. The training sets collects data from the beginning to the failure. The test sets collects data endding at a certain time point before the failure.

TABLE I
THE C-MAPSS DATASETS

Data	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	1	2	2

The four sub datasets details will be provided in Table I. For the convenience of the presentation, we will represent the four datasets as FD001, FD002, FD003, and FD004. The four datasets collect data under various operating conditions and fault modes. According to Table. I, FD001 and FD003 have one operating condition value, and FD002 and FD004 have 6 types of operating conditions value. Therefore, the hyper-parameters of the model will be different for different data during the experiment.

B. Data Processing

Although the dataset includes 21 sensor values and 3 operational settings, some features remain constant over time. During the experiment, constant features are removed. Since the range of variation of different features is totally different, each input feature is regularized using the z-zero method:

$$x_i' = \frac{x_i - \mu_i}{\sigma_i} \tag{11}$$

In the traditional RUL target value prediction, the RUL target value reduces linearly with time and this definition assumes that a system decays linearly over time. In practical application, the decline of a system changes only slightly in the beginning, and suddenly declines rapidly at the end. In order to better model the change of RUL value over time, a piece-wise linear RUL target function is proposed. The maximum value of RUL is limited to a constant value R_e and starts to decay linearly when the RUl value is less than this maximum value. For C-MAPSS datasets, we set R_e to 125 time cycles.

C. Evaluation metrics

Similar to other methods evaluated on C-MAPSS datasets, we use two metrics to evaluate the performance of the proposed method. We use Root Mean Squared Error (RMSE) as a direct evaluation indicator. In addition, we also use a scoring

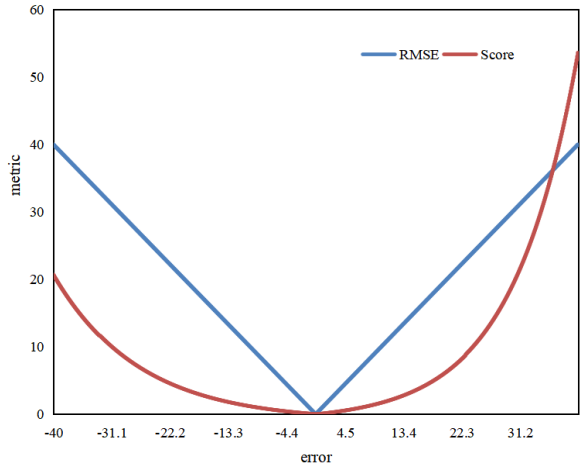

Fig. 5. RMSE and Scoring metrics plot.

function as an evaluation indicator, which is definitions in the following:

$$s = \begin{cases} \sum_{i=1}^{n} e^{-\frac{c_i}{a_1}} - 1, \ if \ c_i < 0 \\ \sum_{i=1}^{n} e^{\frac{c_i}{a_2}} - 1, \ if \ c_i \geq 0 \end{cases} \tag{12}$$

where $a_1 = 13$, $a_2 = 10$, and $c_i = \hat{RUL}_i - RUL_i$. Because the predicted RUL value is greater than or less than the true value will cause different effects. According to Fig. 5, this scoring metrics penalty for positive loss is greater than negative loss.

D. Methods of comparison

We compared the following methods for the task of RUL estimation:

- LSTM+FNN [8] It combines multiple layers of LSTM cells with normal feed forward layers.
- CNN+FNN [9] It contains convolution neural networks following with feed forward layers and dropout components.
- GA+LSTM [30] It proposes a unsupervised deep learning techniques and uses Genetic Algorithm approach to tune the diverse amount of hyper-parameters in the training procedure.
- LSTM+ATTENTION [31] It combines LSTM and attention mechanism.
- DEEP & ATTENTION The DEEP & ATTENTION network contains ATTENION part and DNN part and models timing relationship and noise effect separately.

E. Parameter setting

Because of the limited amount of data in the 4 sub datasets, as shown in the Fig. 3 and Fig. 4, we have done dropout operations on the output of the attention component and the first layer of the dnn component, where the droprate is set to 0.5. We set up the batch size to 256, the learning rate to 0.001, and the maximum epochs to 500. For the multi-head self-attetion layer, we set up the number of the heads to 4. Because the minimum time step of the test set of different sub datasets is different, the time window of FD001, FD002, FD003, FD004 are set to 30, 20, 30, and 19 by respectively. For FD001, we drop the Deep part and for others, we set the number of FFNN layer to 1.The number of CNN RES BLOCK is set to 2, and the filter number of first layer is set to 64, the second is set to 128. We set R_e to 125. Deep & ATTENTION is implemented by Tensorflow and optimized by Adam method. To summarise, the more complex the datasets are, the more layers the deep component has. And we set the parameters related to the datasets as those commonly used in other papers. The parameters related to the model are set to common configuration.

TABLE II
RMSE COMPARISION BETWEEN PROPOSED METHOD AND OTHER METHODS

	rmse				
Method	**FD001**	**FD002**	**FD003**	**FD004**	R_e
LSTM+FFNN	16.14	24.49	16.18	28.17	130
CNN+FFNN	12.61	22.36	12.64	23.31	125
GA+LSTM	12.56	22.73	12.1	22.66	115-135
LSTM+ATTENTION	13.95	17.65	12.72	20.21	125
DEEP & ATTENTION	12.98	17.04	11.88	19.54	125

TABLE III
SCORING COMPARISION BETWEEN PROPOSED METHOD AND OTHER METHODS

	rmse				
Method	**FD001**	**FD002**	**FD003**	**FD004**	R_e
LSTM+FFNN	338	4450	852	5550	130
CNN+FFNN	274	10412	284	12466	125
GA+LSTM	231	3366	251	2840	115-135
LSTM+ATTENTION	320	2102	223	3100	125
DEEP & ATTENTION	282	1386	222	2472	125

F. Effectiveness of Deep & Attention

Table. II and Table. III show the RMSE values and Score values of different methods, respectively. It can be seen from the table that most of the current methods are based on LSTM, and the model only adopted CNN does not show the best effect on the current datasets. In the LSTM-based method, GA+LSTM achieved the best results on the two data sets FD001 and FD003, and LSTM+ATTENTION achieved the best results on the two datasets FD002 and FD004. GA+LSTM combines semi-supervised learning and supervised learning, and can get a good result with a small amount of data. The LSTM+ATTENTION model uses the ATTENTION method to process the output of the LSTM, which can better model the timing relationship. In the four sub datasets, it is difficult to have a good performance on FD002 and FD004. The

LSTM+ATTETNION method has achieved the best results on FD002 and FD004, so Attention and LSTM play a complementary role in the modeling of time series.

Our proposed model is also evaluated on the C-MAPSS dataset. On FD001 and FD003 datasets, our proposed method and GA+LSTM show similar performance. On FD002 and FD004, our proposed method shows significantly better results than existing methods. And the method we proposed does not use the LSTM, so it can be better parallel computing in practical applications. Then, further experiment is carried out to prove the effectiveness of Deep component.

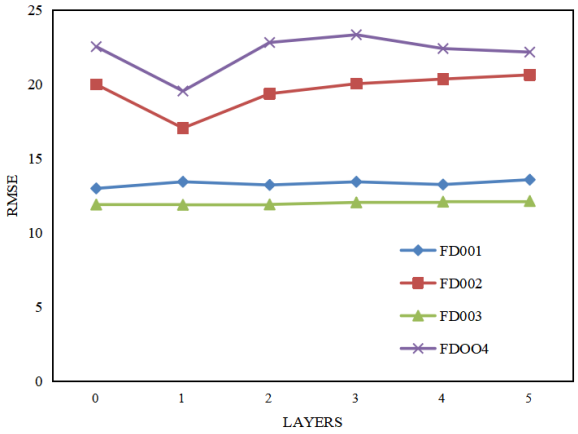

Fig. 6. Effect of hidden layers in DNN part.

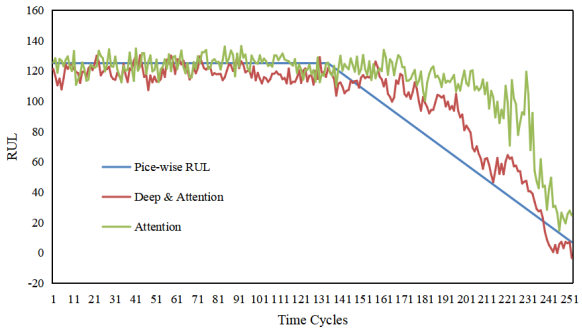

Fig. 7. Effect of DNN part: Attention is only the Attention part removing DNN layer.

G. Effectiveness of Deep Part

We compare the RMSE value of the model when the number of layers of dnn is 0, 1, 2, 3, 4, and 5 on four sub-datasets. As is shown in Fig. 6, FD002, FD003 and FD004 have the best performance when the Dnn layer number is 1; FD001 has the best performance when the Deep component is removed. And when the number of layers of dnn is different, the RMSE values vary considerably on FD002 and FD004 and the RMSE values show a little difference on FD001 and FD003. By comparing the four sub-data sets, we find that adding a Dnn structure with an appropriate number of layers could significantly improve the performance of the model.

It is most difficult to have a good performance on FD004, and the method we proposed has achieved the best results on FD004. So, we further analyze the performance of Deep component on FD004. As shown in Fig. 7, the trend of RUL value is divided into two stages. The first stage is that the real RUL value is greater than R_e and we set up the true RUL values to R_e on the first stage. The second stage is that the real RUL value is less than R_e, and RUL values change linearly on the second stage. As shown in Fig. 7, after adding the Deep component, the model shows better results in the second stage. In practical applications, we should pay more attention to the performance of the model when the machine is about to be damaged. Therefore, the improvement of the second stage, caused by adding Deep compenont, meets the real industrial demand.

V. Conclusion

In this paper, we proposed the Deep & Attention Network, which includes the Deep compenont and the Attention compenont. The Attention component models the timing relationship of the data, while the Deep component accounts for the influence of noise and removes the time relationship of the input data. We have examined this method on the C-MAPSS dataset, and the results show that our proposed model outperforms other current methods on the most complicated sub dataset. Recent state of the art are LSTM based methods, but our proposed method replaces LSTM with Attention mechanism, so our model is more conducive to parallel computing. In addition, the Deep compenont we proposed mainly improves the accuracy on the second stage, which meets the real industrial demand.

Acknowledgment

This work was supported by the National Key Research and Development Program of China No.2018YFD0400902

References

[1] Kalgren, Patrick W., et al. "Defining PHM, a lexical evolution of maintenance and logistics." 2006 IEEE autotestcon. IEEE, 2006.
[2] Wu, Dazhong, et al. "A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing." Journal of Manufacturing Systems 43 (2017): 25-34.
[3] Peng, Yizhen, Yu Wang, and Yanyang Zi. "Switching state-space degradation model with recursive filter/smoother for prognostics of remaining useful life." IEEE Transactions on Industrial Informatics 15.2 (2018): 822-832.
[4] Ma, Meng, Chuang Sun, and Xuefeng Chen. "Discriminative deep belief networks with ant colony optimization for health status assessment of machine." IEEE Transactions on Instrumentation and Measurement 66.12 (2017): 3115-3125.
[5] Hasani, Ramin M., Guodong Wang, and Radu Grosu. "An automated auto-encoder correlation-based health-monitoring and prognostic method for machine bearings." arXiv preprint arXiv:1703.06272 (2017).
[6] Wang, Peng, Robert X. Gao, and Ruqiang Yan. "A deep learning-based approach to material removal rate prediction in polishing." CIRP Annals 66.1 (2017): 429-432.
[7] Saxena, Abhinav, et al. "Damage propagation modeling for aircraft engine run-to-failure simulation." 2008 international conference on prognostics and health management. IEEE, 2008.
[8] Zheng, Shuai, et al. "Long short-term memory network for remaining useful life estimation." 2017 IEEE international conference on prognostics and health management (ICPHM). IEEE, 2017.

[9] Li, Xiang, Qian Ding, and Jian-Qiao Sun. "Remaining useful life estimation in prognostics using deep convolution neural networks." Reliability Engineering & System Safety 172 (2018): 1-11.
[10] He, David, and Eric Bechhoefer. "Development and validation of bearing diagnostic and prognostic tools using HUMS condition indicators." 2008 IEEE Aerospace Conference. IEEE, 2008.
[11] Benkedjouh, Tarak, et al. "Remaining useful life estimation based on nonlinear feature reduction and support vector regression." Engineering Applications of Artificial Intelligence 26.7 (2013): 1751-1760.
[12] Zio, Enrico, and Francesco Di Maio. "A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system." Reliability Engineering & System Safety 95.1 (2010): 49-57.
[13] Huang, Runqing, et al. "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods." Mechanical systems and signal processing 21.1 (2007): 193-207.
[14] Tian, Zhigang. "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring." Journal of Intelligent Manufacturing 23.2 (2012): 227-237.
[15] Bektas, Oguz, et al. "A neural network filtering approach for similarity-based remaining useful life estimation." The International Journal of Advanced Manufacturing Technology 101.1-4 (2019): 87-103.
[16] Zhang, Chong, et al. "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics." IEEE transactions on neural networks and learning systems 28.10 (2016): 2306-2318.
[17] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.
[18] Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal processing magazine 29.6 (2012): 82-97.
[19] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." Neural computation 1.4 (1989): 541-551.
[20] Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).
[21] Matthew, D., and R. Fergus. "Visualizing and understanding convolutional neural networks." Proceedings of the 13th European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland. 2014.
[22] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[23] Babu, Giduthuri Sateesh, Peilin Zhao, and Xiao-Li Li. "Deep convolutional neural network based regression approach for estimation of remaining useful life." International conference on database systems for advanced applications. Springer, Cham, 2016.
[24] Li, Xiang, Wei Zhang, and Qian Ding. "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction." Reliability Engineering & System Safety 182 (2019): 208-218.
[25] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.
[26] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[27] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).
[28] Yuan, Mei, Yuting Wu, and Li Lin. "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network." 2016 IEEE International Conference on Aircraft Utility Systems (AUS). IEEE, 2016.
[29] Wu, Yuting, et al. "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks." Neurocomputing 275 (2018): 167-179.
[30] Ellefsen, André Listou, et al. "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture." Reliability Engineering & System Safety 183 (2019): 240-251.
[31] da Costa, Paulo Roberto de Oliveira, et al. "Attention and long short-term memory network for remaining useful lifetime predictions of turbofan engine degradation." International Journal of Prognostics and Health Management 10 (2019): 034.
[32] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
[33] Song, Huan, et al. "Attend and diagnose: Clinical time series analysis using attention models." arXiv preprint arXiv:1711.03905 (2017).
[34] Galassi, Andrea, Marco Lippi, and Paolo Torroni. "Attention, please! a critical review of neural attention models in natural language processing." arXiv preprint arXiv:1902.02181 (2019).