Adaptive Spatio-Temporal Graph Convolutional Neural Network for Remaining Useful Life Estimation

Yuxuan Zhang
School of Aeronautics and Astronautics
Shanghai Jiao Tong University
Shanghai, China
Email: yuxuanzhang@sjtu.edu.cn

Yuanxiang Li
School of Aeronautics and Astronautics
Shanghai Jiao Tong University
Shanghai, China
Email: yuanxli@sjtu.edu.cn

Xian Wei
Fujian Institute of Research on
the Structure of Matter
Chinese Academy of Sciences
Fuzhou, China
Email:xian.wei@fjirsm.ac.cn

Lei Jia
Shanghai Jiao Tong University
School of Aeronautics and Astronautics
Shanghai, China
Email: westjia@sjtu.edu.cn

***Abstract*—Accurate remaining useful life (RUL) estimation is of crucial importance to numerous industrial applications where safety and reliability are among primary concerns. Recently, deep learning based prognostics methods have been emerging as an effective method to improve RUL prediction results. However, these methods, e.g. recurrent neural networks (RNNs), convolutional neural networks (CNNs), only capture temporal information of the sensory data while ignoring intrinsic spatial relations between sensors. To solve this problem, in this work, we propose a framework, namely, adaptive spatio-temporal graph convolutional neural network (ASTGCNN). The proposed framework consists of two parts. In the spatial domain, since the intrinsic graph structure of sensors is not provided in most situations, a dynamic graph neural network is proposed to learn the sensors' spatial relation. In the temporal domain, a stacked dilated 1D CNN is utilized to capture long range dependency of input sensor signals. These two parts are integrated in a unified framework and can be trained in an end-to-end manner. The performance of ASTGCNN is investigated on the turbofan engine dataset. Experimental results show that the proposed framework can improve the RUL prediction performance of the current deep learning methods, and learn the intrinsic spatial information of sensors.**

***Index Terms*—RUL estimation, spatio-temporal modeling, adaptive graph learning**

I. Introduction

Regular maintenance is of crucial importance in many industry areas where high reliability is required, e.g. aerospace, automotive and manufacturing. In order to reduce unnecessary maintenance and improve safety and reliability, prognostics and health management (PHM) has been emerging as a hot research topic in the last few decades, with remaining useful life (RUL) prediction being a challenging task.

By leveraging conditional maintenance (CM) data, accurate RUL prediction results can be achieved using machine learning techniques [1]. Among various machine learning algorithms, deep learning has been demonstrated as one kind of effective method with great scalability and generalization, and has gained a lot of research attention in recent years [2]–[6]. Compared with traditional machine learning prognostic methods, deep learning based prognostics do not require any specific feature extraction techniques to construct health indicator. Deep learning based RUL prediction methods consist of two stages. Firstly, run-to-failure sensory data are collected for network training. Secondly, using a sliding time window, the run-to-failure sensory data can be divided into smaller sequences which can be used as the network input and corresponding end-of-life running cycles or times are used as label. Then RUL prediction problem can be converted to a deep learning regression problem. With a well trained deep neural network, the newly observed sensory data are fed into the network for RUL prediction.

For example, Babu et al. [2], Li et al. [3] utilized 1D convolution neural networks (CNNs) to capture the information in the temporal domain to predict the RUL of aircraft engines. Recurrent neural networks (RNNs), as a kind of powerful methods to model time series data, also have been used in prognostics. For example, Zheng et al. [4] proposed a long short-term memory(LSTM) network for RUL estimation and outperformed the traditional approaches. Ellefsen et al. [5] proposed a LSTM based semi-supervised deep architecture for turbofan engine degradation prediction. Huang et al. [6] developed one kind of bidirectional LSTM framework to jointly learn operational conditions and sensor signals.

Since the sensory data are always in the sequential form, it can be observed that both CNNs and RNNs based prognostic methods only extract features in the temporal domain, while the information in the spatial domain are deprecated. For a concrete example, in the aircraft tubofan engine RUL

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

prediction task, signal sensors such as pressure, temperature and flow are located at different parts in the engine system. The temperature signal at one location could be affected by its nearby physical signals such as pressure or flow. It means there exists a underlying graph structure in the turbofan engine system, where the sensors at different locations can be considered as the nodes and the relations among sensors can be considered as the edges. So it is nature to take the underlying graph structure into consideration when modelling the time series sensory data for RUL prediction. Therefore, the challenge for RUL prediction problem is how to model the time series sensory data with capturing spatial and temporal dependencies simultaneously.

Recently, spatio-temporal graph neural networks have verified that it is effective to model the time series data with a graph structure. The typical applications include road traffic forecasting [7]–[9], wind speed forecasting [10], action recognition [11], and driver maneuver anticipation [12]. These researches either integrate graph convolution neural networks into RNNs or into CNNs, and have the following challenges when applied to RUL estimation tasks.

1) In the traffic forecasting problem task, the road Euclidean distance between nodes can be considered as the edge weights. When used in the skeleton recognition, distances between joints coordinates are regarded as edges. However, in RUL estimation tasks, although these senor signals have inter-dependency relationships, connections between sensors are always not provided. In this case, spectral-based graph convolution network (GCN) and spatial-based GCN can not be used, where a predefined graph structure is required.
2) RNN-based spatio-temporal graph networks have the inherent deficiency in learning long-term dependencies and cannot be parallelized. Although the CNN-based spatio-temporal graph networks can be parallelized, they still suffer from the problem of capturing long-term dependencies. Because in standard 1D CNNs, the receptive field grows linearly with the network depth. When the input sequences are long, more convolution layers should be added to expand the receptive field of the network .

To overcome the above-mentioned deficiencies of existing spatio-temporal graph neural network for RUL prediction, we propose a framework, namely, adaptive spatio-temporal graph convolutional neural network(ASTGCNN). Our contributions are as follows:

1) Two adaptive spatial graph convolution layers for underlying graph structure learning are proposed. The first one adopts the metric learning to adaptively learn adjacent matrix in the spatial feature space. The second one utilizes the attention mechanism to learn the graph structure in an embedded dot-product way.
2) 1D dilation convolution is utilized to capture temporal dependency with the receptive field growing exponentially with the depth of network. A unified framework is proposed to integrate these two parts, with an end-to-end training manner.

This paper is structured as follows: Section II introduces the researches related to our work. Section III gives the mathematical definition of the RUL estimation problem and introduces the proposed framework. Section IV shows the experimental results on the turbofan engine dataset. Section V draws the conclusion.

II. Related Works

A. Graph Convolution Networks

Graph convolution networks are deep learning approaches for graph structured data [13], and have been applied in domains such as network embedding, link prediction, node classification, etc. There are two kind of basic approaches exploring to generalize CNNs in the irregular graph domain. One is to expand the definition of graph convolution in the spatial domain, and the other uses the graph Fourier transforms [14] to extend GCNs in the spectral domain [15].

Let $x \in \mathbb{R}^{n \times d}$ be a graph signal, i.e., a signal defined on an undirected weighted graph $\mathcal{G} = (V, \mathcal{E}, A)$, where V is a finite set of $|V| = n$ vertices, $\mathcal{E}$ is a set of edges and $A \in \mathbb{R}^{n \times n}$ is the adjacent matrix. The row i of graph signal x is a d dimensional feature vector at the i^{th} node. Denote $*_\mathcal{G}$ convolution operator defined on graph $\mathcal{G}$. The spectral graph convolution is defined as

$$y = g_\theta *_\mathcal{G} x = g_\theta(L)x = g_\theta(U\Lambda U^T)x = Ug_\theta(\Lambda)U^T x, \quad (1)$$

where $U \in \mathbb{R}^{n \times n}$ is the eigenvectors matrix and $\Lambda \in \mathbb{R}^{n \times n}$ is the diagonal matrix of eigenvalues of the normalized graph Laplacian $L = I_n - D^{-1/2}AD^{-1/2}$. g_θ is a parametric kernel, where $g_\theta = diag(\theta)$, $\theta \in \mathbb{R}^n$ is a Fourier coefficients vector. $D \in \mathbb{R}^{n \times n}$ is the diagonal graph degree matrix, where $D_{ii} = \sum_j W_{ij}$. Note that the computation of matrix multiplication is $\mathcal{O}(n^2)$.

ChebNet [16] reduces the computation complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(K\mathcal{E})$. It uses Chebyshev polynomials to parameterize g_θ as a truncated expansion,

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}), \quad (2)$$

where $T_k(\tilde{\Lambda})$ is Chebyshev polynomials of $\tilde{\Lambda} = 2 * \Lambda/\lambda_{max} - I_n$, and $\theta \in \mathbb{R}^K$ is a vector of Chebyshev coefficients. Then the graph convolution can be written as

$$y = g_\theta(L)x = Ug_\theta(\Lambda)U^T x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x. \quad (3)$$

B. Spatiol-temporal Graph Networks

A spatio-temporal graph is a kind of network where the nodes attributes changes dynamically over time. A spatio-temporal graph is defined as $\mathcal{G}^t = (V, \mathcal{E}, A, x^t)$, with $x^t \in \mathbb{R}^{n \times d}$ changes dynamically. Thus, the researches on spatio-temporal graph networks are aimed to capture the spatial dependency and the temporal dependency simultaneously. The majority researches on spatio-temporal graph networks follows two main directions, the RNN-based and CNN-based methods. An early work of RNN-based methods [9] incorporated spectral GCN into the recurrent long short-term memory network to capture spatio-temporal dependency. [8] used diffusion convolution combined with RNN to improve model performance. The main drawback of RNN-based spatio-temporal networks is the inherent gradients vanishing and explosion problems and low efficiency when training on long sequence data. The CNN-based methods [7], [11] adopted the 1D CNN and graph convolution to make the model more efficient. The graph Wavenet [17] used dilated convolution to increase the receptive field of CNN-based methods.

III. METHODOLOGY

In this section, we first give the mathematical definition of the RUL estimation problem. Then we show the pipeline of the proposed method. After that, we introduce the two parts of the proposed model.

A. Problem Formulation and Pipeline Overview

Let $\mathcal{G}^t = (V, \mathcal{E}, A^t)$ denote the graph structures of the sensory data in an industrial system, where A^t is a dynamic adjacent matrix. Denote $x^t \in \mathbb{R}^{n \times d}$ the sensory data, and y^t the corresponding remaining useful life. In this paper, A^t is calculated dynamically from x^t, where $A^t_{ij} = f(x^t_i, x^t_j)$, $x^t_i \in \mathbb{R}^d$ is the data of the i^{th} sensor at time t. The RUL estimation problem is to predict the most likely $\hat{y}^t$ given the previous J observations,

$$\hat{y}^t = arg\max_{y^t} P(y^t | x_{t-J+1}, ..., x_t; \mathcal{G}), \tag{4}$$

where $P(y^t | x_{t-J+1}, ..., x_t; \mathcal{G})$ is the probability of y^t conditioned on the past J observations.

The pipeline of the proposed framework is shown in Fig. 1, we use the structure of WAVENET [17], [18], which has been proved effective in learning sequential data. In the proposed ASTGCNN, we have stacked k layers of ASTGCNN layer, and in each ASTGCNN layer, we use temporal CNN to extract information in the time domain. After that, an adaptive GCN is utilized to learn the adjacent matrix $A^t_{ij} = f(x^t_i, x^t_j)$, and then new spatial features are achieved using graph convolution. Residual connection is utilized in the whole framework. After k layers of stacked ASTGCNN, an regression layer is added to predict the RUL y^t.

B. Adaptive Spatial Graph Convolution Layer

As shown in Section I, a predefined graph structure is not provided in RUL prediction task, and has to be learned from sensory data. We provide two ways of calculating the pair-wise function $f(x^t_i, x^t_j)$.

1) Adaptive Metric Graph Convolution: Since Euclidean distance is not a good metric for graph structured data [19]. Generalized Mahalanobis distance is used to adaptive learn the adjacency matrix in the spatial feature space. In this case, the pair-wise distance function is formulated as:

$$D(x^t_i, x^t_j) = \sqrt{(x^t_i - x^t_j)^T M (x^t_i - x^t_j)}. \tag{5}$$

Since M is a Symmetric semi-definite matrix, we use the PP^T to parameterize it, where $P \in \mathbb{R}^{d \times d}$ is the trainable weights. In this way, the distance metric can be learned adaptively according to the characteristic of feature space.

Then, by using the Gaussian kernel, the adaptive adjacent matrix is formulated as:

$$A^t_{ij} = f(x^t_i, x^t_j) = e^{-D(x^t_i, x^t_j)^2 / 2\sigma^2}. \tag{6}$$

Using the adjacent matrix calculated by Eq. (6), the normalized graph Laplacian L^t can be derived using $L^t = I_n - D^{-1/2} A^t D^{-1/2}$. Using Eq. (3), the adaptive graph convolution can be formulated as:

$$y^t = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}^t) x^t W_k, \tag{7}$$

where $\tilde{L}^t = 2L^t/\lambda_{max} - I_n$. Using the recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. Eq. (7) is a K order polynomial of the adaptive Laplacian, which is K-localized, which means each nodes depends only on nodes that are at maximum K hops away from the central node. The parameter $W_k \in \mathbb{R}^{d \times d}$ are trainable weights, which perform transformation in the feature space.

2) Spatial Attention Graph Convolution Layer: Attention mechanism has been proved effective in relationship learning [20], which evaluates similarity in an embedded dot-product way. In this paper, we adopt the attention mechanism to learn the intrinsic spatial structure in the graph data. Thus, the similarity function f is formulated as:

$$f(x^t_i, x^t_j) = \frac{1}{\mathcal{C}(x^t)} (W_\theta x^t_i)^T (W_\phi x^t_j), \tag{8}$$

where $C(x^t) = N$, and N is the number of nodes in graph data. While in the original attention mechanism, $C(x^t) = \sum_{\forall j} f(x^t_i, x^t_j)$ is always used. We use this form to simplify gradient calculation [21]. Then the spatial attention graph convolution is formulated as:

$$y^t = \frac{1}{\mathcal{C}(x^t)} (W_\theta x^t_i)^T (W_\phi x^t_j) x^t \tag{9}$$

C. Temporal Convolution Layer

Since most RNNs cannot be parallelized and are vulnerable to gradient exploding and vanishing problem in capturing long-range dependencies, we adopted dilated CNN [17], [18] to be our temporal convolution layer. The visual comparison between dilated convolution and casual convolution is illustrated in Fig. 2. In a standard 1D-CNN, the receptive field grows linearly with the depth of network, while in a dilated CNN, the receptive field grows exponentially with the network depth, making it possible to capture long range-dependency with less layers. Denote $*$ the temporal convolution operator, the dilation convolution in the temporal dimension is formulated as:

$$z = x^t * f(t) = \sum_{s=0}^{T-1} f(s) x(t - d \times s), \tag{10}$$

where d is the dilation factor.

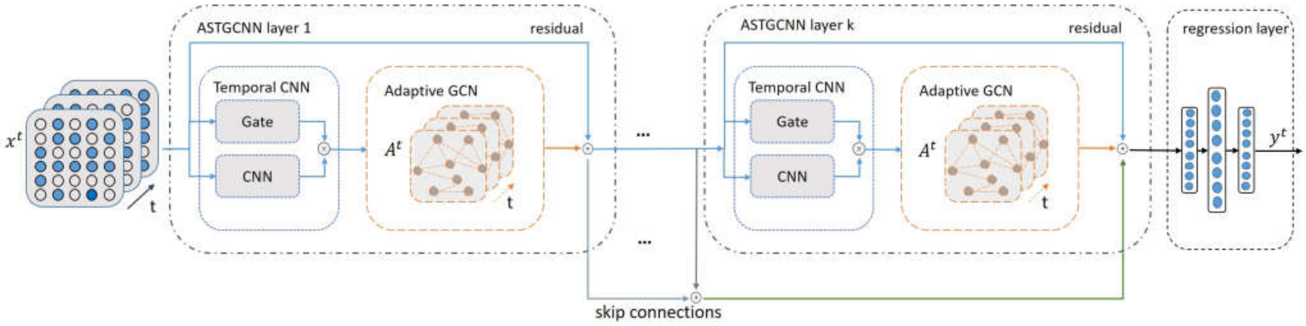


Fig. 1. A simple illustration of the proposed ASTGCNN. The network consists of k stacked ASTGCNN layer. In each ASTGCNN layer, a temporal convolution layer is used to capture dependency in the time domain. After that an adaptive graph convolution layer is added to learn the underlying graph information.

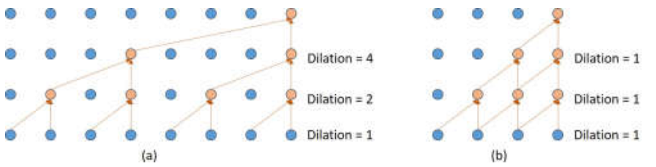


Fig. 2. A comparison with dilation convolution and causal convolution. (a) is a stacked four layers dilation convolution, and (b) is stacked four layers of casual convolution. With the same deep, dilation convolution has a receptive field size of 8, while casual convolution has a receptive field size of 4.

Gating mechanism has been proved effective in sequence learning. We adopt the same architecture as [17] used in spatio-temporal traffic forecasting. It takes the forms:

$$z = tanh(\theta_1 * X + b) \odot \sigma(\theta_2 * X + c), \tag{11}$$

where $\odot$ denotes the Hadamard product, and $\sigma(\cdot)$ denotes the sigmoid activation function. θ_1 and θ_2 denote convolution parameters in Gate CNN and CNN separately.

D. *Regression layer*

After stacked k layers of ASTGCNN layer, a regression layer is added to output the final RUL $\hat{y}^t$. The parameter k is artificially designed to get a proper receptive field size. Mean square error is chosen as the training objective of ASTGCNN for RUL estimation. It is defined as:

$$L(x_{t-J+1}, ..., x_t; \Theta) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i^t - y_i^t)^2 \tag{12}$$

IV. Experimental Results

In this section, we evaluate our method on the four subsets of NASA turbofan engine dataset [22].

A. *Dataset Description*

The turbofan engine dataset comes from the NASA Ames prognostics data repository. Engine degradation simulation was carried out using Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tool. Details of the simulation setting and the platform can be found in [23].

Run-to-failure engine degradation data in four different subsets are acquired under different combinations of operational conditions and fault modes. Several sensor channels were recorded to characterize fault evolution, and details of the C-MAPSS subsets is shown in TABLE I.

TABLE I
Information of the C-MAPSS Dataset

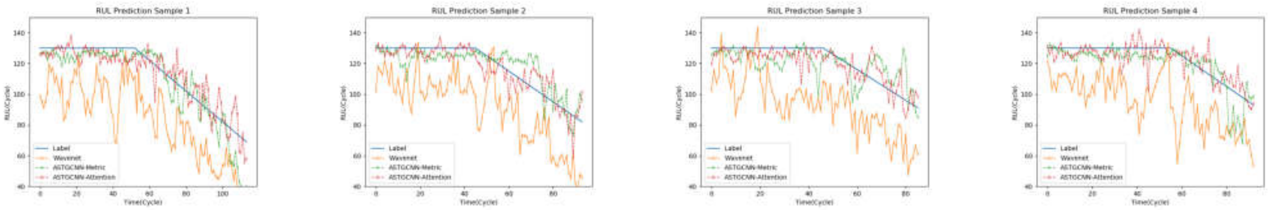
Dataset	C-MAPSS			
	FD001	FD002	FD003	FD004
Engine units	200	519	200	497
Operation conditions	1	6	1	6
Fault modes	1	1	2	2

The sensory data from 18 sensors are selected, i.e., sensor #1, #2, #4, #5, #6, #7, #9, #10, #11, #12, #14, #15, #16, #17, #18, #21, #22 and #23. A sliding time window with length of a fixed length T and stride of 1 is used to enclose the data into small multidimensional time sequences. Min-max normalization is used to covert the sensory data with different scales within the range $[0, 1]$. Based on the assumption that degradation is a piece-wise linear process,which has been validated effective for this dataset [2]–[4]. It is assumed to have a constant RUL value at the initial stage and degrades linearly afterwards. In our experiment, the constant RUL is set to 130.

B. *Baselines*

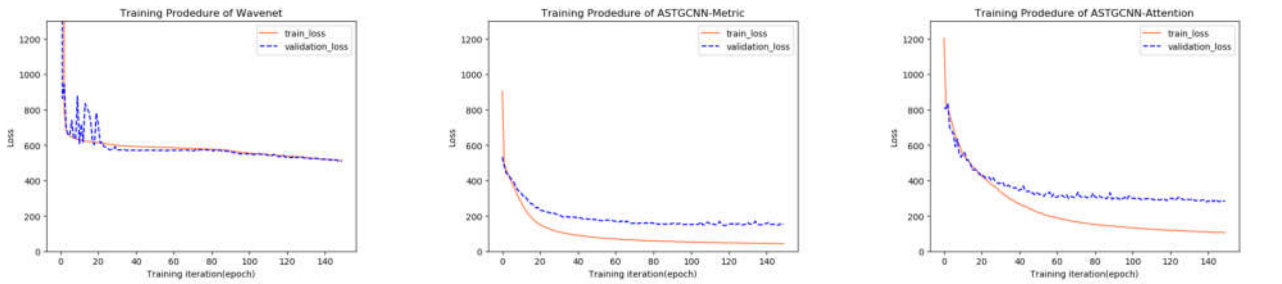
We compared the proposed ASTGCNN with the following models:

- **DCNN1.** An early CNN architecture using stacked 1D-convolution for RUL estimation [2] .
- **DCNN2.** An improved deep CNN framework for RUL prediction [3].
- **MODBNE.** A multiobjective deep belief networks ensmeble method for RUL prediction [24].
- **DeepLSTM.** Stacked RNN architecture with fully connected LSTM units [4] .
- **B-LSTM.** A bidirectional long short-term memory network for RUL estimation. [6].



(a) (b) (c) (d)

Fig. 3. Comparison sample plot of RUL prediction result. The blue line is RUL label, green line represents the ASRGCNN-Metric result, red line is the ASRTGCNN-Attention prediction result and the orange line shows the Wavenet result.



(a) Training loss plot of Wavenet (b) Training loss plot of ASTGCNN-Metric (c) Training loss plot of ASTGCNN-Attention

Fig. 4. Training and validation loss over iteration of Wavenet, ASRTGCNN-Metric and ASTGCNN-Attention. Orange line denotes the training loss and the blue line represents the loss on the validation set.

- **Wavenet.** A dilation convolution architecture for modeling sequence data [18].

C. Experimental Setting and Evaluation Metrics

We conducted the experiments on a 64-bit windows server with one Intel Core i7-7800x cpu, a 64G RAM and one Nvidia GTX 1080Ti GPU. The code is implemented using the Pytorch library, which is available at https://github.com/zhangyu233/AdaptiveSPT.

To quantitatively evaluate the performance of our methods, two evaluation metrics are used, i.e., root mean square error(RMSE) and average scoring function(ASF). The commonly used RMSE is defined as the error of estimated RUL and true label. Denote $\hat{y}_i$ the predict RUL, y_i the RUL label and Ω the indices of observed samples. The RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega} d_i^2}, \tag{13}$$

where $d_i = \hat{y}_i - y_i$ is the error between the estimated $\hat{y}$ and the true RUL y of the i-th testing sample. In prognostics areas, early prediction is always better than late prediction. The ASF metric gives penalty to the late prediction, which is defined as:

$$s_i = \begin{cases} e^{\frac{-d_i}{13}} - 1, \ d_i \leq 0 \\ e^{\frac{d_i}{10}} - 1, \ d_i \geq 0, \end{cases} \qquad ASF = \frac{1}{N}\sum_{i}^{N} s_i. \tag{14}$$

D. Comparison with other Deep Learning Methods

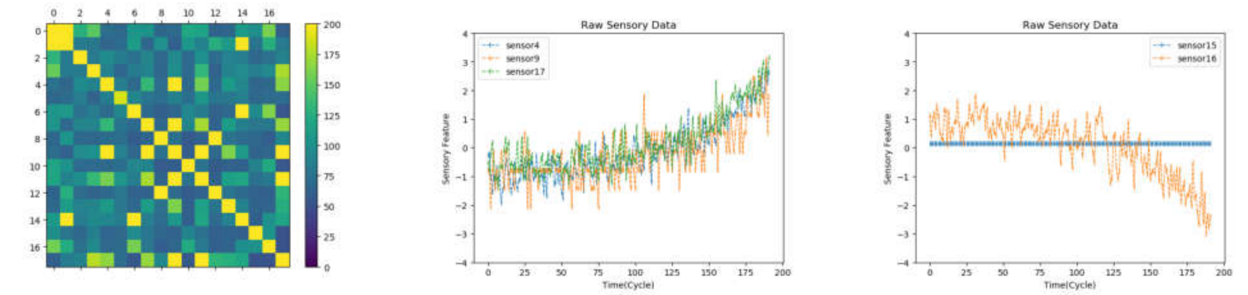
TABLE II compares the experimental results of the proposed method ASTGCNN with the baseline models on the four subsets FD001, FD002, FD003 and FD004 of turbofan engine dataset. ASTGCNN obtains the superior results on all of the four subsets. Compared with the RNNs based deep learning methods including DeepLSTM and B-LSTM, ASTGCNN-Metric and ASTGCNN-Attention outperforms them by a large margin. Compared with the CNNs based prognostic methods such as DCNN1 and DCNN2, ASTGCNN-Metric and ASTGCNN-Attention still have the best performance on these datasets. Compared with the Wavenet which only utilized the temporal dilation convolution, ASTGCNN-Metric and ASTGCNN-Attention has a better prediction accuracy, which indicates that spatial information is beneficial for RUL prediction.

Fig. 3 compares the RUL prediction result of ASTGCNN-Metric and ASTGCNN-Attention on 4 test samples. It shows that ASTGCNN-Metric generates the most stable results. ASTGCNN-Metric and ASTGCNN-Attention produces more accurate prediction results than Wavenet. Since Wavenet does not utilize any spatial information, its prediction result has a big deviation from the label.

Fig. 4 compares training and validation loss over the training iteration of ASTGCNN and Wavenet. It is illustrated that ASTGCNN-Metric converges using the least number of

TABLE II
PERFORMANCE OF THE PROPOSED METHODS AND OTHER DEEP LEARNING METHODS

Method	FD001		FD002		FD003		FD004	
	RMSE	ASF	RMSE	ASF	RMSE	ASF	RMSE	ASF
DCNN1 [3]	12.61	2.73	22.36	40.20	12.64	2.84	23.31	50.20
DCNN2 [2]	18.45	12.87	30.29	53.00	19.82	15.96	29.16	31.80
MOBDBNE [24]	15.04	3.34	25.05	21.81	12.51	4.22	28.66	26.44
DeepLSTM [4]	16.14	3.38	24.49	17.18	16.18	8.52	28.17	22.38
B-LSTM [6]	–	–	25.11	18.72	–	–	26.61	20.04
Wavenet [18]	17.45	5.29	22.48	16.15	12.04	2.61	21.34	25.84
ASTGCNN-Metirc	**8.78**	**1.46**	**12.25**	**2.58**	**7.76**	**0.89**	**13.19**	**5.64**
ASTGCNN-Attention	10.23	2.04	16.67	8.27	8.35	1.37	13.84	8.26



(a) Dynamic spatial graph adjacent matrix (b) Raw sensory data of sensor 4, 9 and 17. (c) Raw sensory data of sensor 15 and 16

Fig. 5. Illustration of learned spatial graph. (a): Heat-map of learned adjacent matrix. (b)(c): Plot of raw sensory data.

TABLE III
COMPARISON OF COMPUTATION TIME

Model	Computation Time	
	Training(s/epoch)	Inference(s)
Wavenet	37.75	3.06
ASTGCNN-Attention	46.96	5.04
ASTGCNN-Metric	63.65	7.11

training steps and has the smallest training loss. ASTGCNN-Attention has a worse convergence than ASTGCNN-Metric, however it still outperforms Wavenet by a large margin, which verifies the importance of spatial information.

E. Dynamic Spatial Graph Analysis

In order to show the effectiveness of the proposed adaptive spatial graph convolution layer, we plot the learned graph adjacent matrix and the corresponding sensory data in Fig. 5. It is shown in Fig. 5(a) that sensor 9 has a high level correlation with sensor 4 and sensor 17. In addition, sensor 15 and sensor 16 have a low level correlation. Fig. 5(b) plots the raw sensory data of senor 4, sensor 9 and sensor 17. It is obvious that these signals have similar trends, indicating these signals have a high level of correlation. Fig. 5(c) plots the raw sensory data of sensor 15 and sensor 16. It can be observed that these two signals are obviously not similar to each other, which is consistent with our learned adjacent matrix. These results verify that our model could learn the intrinsic spatial information of sensors.

F. Computation Time

TABLE III shows the computation time of the proposed method ASTGCNN and Wavenet in our experiment. It can be observed that the proposed ASTGCNN requires more training time and inference time than the Wavenet for the extra added spatial graph convolution layer. In addition, ASTGCNN-Metric requires more computation time than ASTGCNN-Attention.

V. CONCLUSION

In this paper, a deep learning framework ASTGCNN was proposed for RUL prediction. To utilize the intrinsic spatial information in industrial systems, two kinds of adaptive spatial graph convolution layer, i.e., ASTGCNN-Metric and ASTGCNN-Attention, were proposed to learn the spatial information of sensory data. Dilation convolution was utilized to capture long range temporal information.

Experimental results have shown that the proposed methods have a good performance on the benchmark turbofan engine dataset and have outperformed the existing deep learning based approaches. ASTGCNN-Metric has a better performance than ASTGCNN-Attention, however, ASTGCNN-Metric requires more computation time. In addition, experimental results verifies that the proposed ASTGCNN could also learn the underlying graph structure of sensory systems which is beneficial to physical analysis.

ACKNOWLEDGMENT

This work was partially supported by National Natural Science Fund of China under Grant No:U1406404, Civil Aviation Special Project of China under Grant No:MJZ-2016-S-44, National Science Found for Young Scholars under Grant No. 61806186, State Key Laboratory of Robotics and System(HIT) under Grant No. SKLRS-2019-KF-15, the program 'Construction of Fujian Research Institute on Intelligent Logistics Industry Technology' under Grant No. 2018H2001, CAS Pioneer Hundred Talents Program (Type C) under Grant No.2017-122, and the program 'Quanzhou Science and Technology Plan' under Grant No. 2019C112, No. 2019C011R and No. 2019STS08.

REFERENCES

[1] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation–a review on the statistical data driven approaches," *European journal of operational research*, vol. 213, no. 1, pp. 1–14, 2011.

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

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

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

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

[6] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bi-directional lstm prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, 2019.

[7] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[8] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[9] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 362–373.

[10] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 670–681, 2018.

[11] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5308–5317.

[13] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[15] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[16] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[18] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[19] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[22] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, 2008.

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

[24] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2306–2318, 2016.