Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network

Conference Paper · October 2018

DOI: 10.1109/PHM-Chongqing.2018.00184

4 authors:

Wang Jiujian
Hunan University

Guilin Wen
Guangzhou University

Shaopu Yang
Shijiazhuang Tiedao University

Yongqiang Liu
Shijiazhuang Tiedao University

Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network

Jiujian Wang, Guilin Wen
State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body
Hunan University
Changsha, 410082, China
E-mail: wangjiujian@hnu.edu.cn

Shaopu Yang, Yongqiang Liu
Key Laboratory of Traffic Safety and Control in Hebei
Shijiazhuang Tiedao University
Shijiazhuang, 050043, China

***Abstract*—Remaining Useful Life (RUL) estimation plays a crucial role in Prognostics and Health Management (PHM). Traditional RUL estimation models are built on sufficient prior knowledge of critical components degradation process which is not easily available in most situation. With the development of integrated circuit and sensor technique, data-driven approaches show good potential on RUL estimation. This paper proposes a new data-driven approach with Bidirectional Long Short-Term Memory (BiLSTM) network for RUL estimation, which can make full use of the sensor date sequence in bidirection. By visualized analysis of the hidden layers, the model can expose hidden patterns with sensor data of multiple working conditions, fault patterns and degradation model. With experiment using C-MAPSS dataset, BiLSTM approach for RUL estimation outperforms other traditional approaches for RUL estimation.**

Keywords—prognostics and health management, remaining useful life, bidirectional LSTM, deep learning

I. INTRODUCTION

Engineering maintenance and prognostics play a very crucial role in modern industry such as aerospace, locomotive, manufacturing and so forth. Traditional method as breakdown corrective maintenance and scheduled preventive maintenance can't satisfy the increasing demand of economic efficiency, reliability and safety [1]. Therefore, advanced condition-based maintenance (CBM) which is known as prognostic and health management (PHM) has been proposed [2]. The targets of PHM include reducing the maintenance cost, improving the reliability and enhancing the performance ability during long working period [3]. Key techniques of the PHM are sensor technique and sensor selection for monitoring, data acquisition, data processing, algorithms for fault identification, failure prediction, remaining useful life estimation and so on.

Remaining useful life (RUL) is a crucial part of PHM. RUL is defined as the life time length from current time to the time of totally failure of the equipment. Generally, the methods for RUL can be grouped into model-based approaches [4] and data-driven approaches [5]. Model-based approaches build precisely failure models based on the physical failure mechanism. These approaches are very useful when lacking of failure data. However, model-based approaches are very difficult and complex to build which need extensive prior knowledge about physical mechanism. So in reality, physical failure models for many equipment don't exist. On the other hand, data-driven model approaches build the degradation model with historical sensor data. The data-driven model can learn the mapping relation between sensor data and degraded life time by training with the historical collected sensor data. Especially with the development of integrated circuit and sensor technique, it is easier to get lots of raw historical sensor data at a low cost. So the data-driven approaches are more feasible to build accurate RUL estimation model.

In recent years, date-driven approaches to discovering the relationship between monitored data and the corresponding RUL gradually become prosperous. Many machine learning techniques, especially the neural network-based approaches have been researched on RUL estimation task. The feedforward neural network such as MLP [6], ANN [7] and CNN [8] [9] are used where sensor data are segmented into a series of sliding windows. Each sliding window has a RUL label value and is considered as independent from other windows. So the length of window is limited the accuracy of RUL model, and this limitation makes the model can't consider the long range dependencies of the sensor data.

The sensor data are time sequence with long time range dependencies in nature. Sequence machine learning approaches such as Hidden Markov Model (HMM) [10] and Recurrent Neural Network (RNN) [5] are also researched. Hidden states of HMM are drawn from a limited seized discrete state space. So when the number of hidden stages is large, it is difficult to calculate and storage the hidden stages. Moreover, each hidden state only depends on the immediately previous state without long range dependency [11]. RNN is a time sequence neural network and its output can be continuous value which is suitable for RUL estimation task. However, with the vanishing or exploding gradient problem, it is not feasible to long-term time dependency data.

Long Short-Term Memory Network (LSTM) is a type of RNN network which avoids the shortcoming of vanishing or exploding gradient problem with a system of gating units. The LSTM network can remember information for long periods of time which makes it suitable for the RUL estimation task [12]. However, traditional LSTM or other types of RNN calculate the hidden state only by using the forward direction of input

2166-5656/18/$31.00 ©2018 IEEE
DOI 10.1109/PHM-Chongqing.2018.00184

data, which do not consider the future input date. So it doesn't take full advantage of all collected sensor data. Considering this shortcoming the bidirectional network architecture is proposed.

Bidirectional LSTM (BiLSTM) is a bidirectional network with the LSTM cell for sequence learning, and has been widely used on speech recognition and machine translation with good performance [13]. The BiLSTM network can learn the dependencies of sensor data in both forward and backward direction, which can take full advantage of input data. In this work, the BiLSTM based approach is proposed for RUL estimation, which uses multiple layers of BiLSTM layers and uses the full connected neural network layer before the output.

The rest of this paper is organized as follow. Section II introduces BiLSTM model, data preparation and evaluating value; Section III uses the CMAPSS Dataset to test the BiLSTM based model and evaluates the performance comparing with other network model; Section IV makes a conclusion of this paper and discusses future work for RUL estimation task.

II. BIDIRECTIONAL LSTM NETWORK FOR RUL ESTIMATION

In this section, the proposed deep BiLSTM network architecture for RUL estimation will be demonstrated as well as some important processes such as loss function, normalization methods, and evaluation functions and so on.

A. Bidirectional LSTM Model

With an input time sequence $\mathbf{x}=(x_1,...,x_T)$, a RNN [13] gives results of the output vector sequence $\mathbf{y}=(y_1,...y_T)$ and hidden vector sequence $\mathbf{h}=(h_1,...h_T)$ .The iterating calculation equations are as follow with the time range from $t=1$ to T :

$$h_t = \mathrm{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = W_{hy}h_t + b_y \tag{2}$$

where the W terms denote weight matrices, the b terms denote bias vectors and H is the activation function which is usually a sigmoid function.

In theory, the RNN can consider the dependency of all the forward input data. However, the gradient of the RNN is easily decreased to zero or increased to infinite during the training process. This is because the W_{hh} weight metrics of hidden vector sequence will be multiplied t times if simplify the nonlinear section of the Eq.(2) in the process of calculation of y_t . So if the eigenvalue of the $(W_{hh})^t$ matrices is less or larger than 1, the result will converge towards zero or infinite which is called vanishing or exploding gradient problem.

In order to train the model with back propagate algorithm, it is hoped the product of derivatives keep closed to the value of one. One way to realize this is to creating paths through time that have derivatives which neither vanish nor explode by setting gates. With the control of gates, the time scale of integration can be changed dynamically. One effective sequence model of the gated RNN is LSTM.

LSTM [15] is a special designed RNN which adding a system of gating units that controls the flow of information. The activation function of gates is usually sigmoid function that the output of which is between 0 and 1. The gates do not offer information but only limit the information flow. The structure of LSTM is shown in Fig. 1 and the calculation equations are as follow:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

where i, f, o denote respectively the input gate, forget gate and output gate, and c denotes memory cell and h denotes the hidden vector sequence. The σ is the activation function which is sigmoid function.

Although the respective equations of the three gate have same structure, but the weight matrixes are with different values as well as with difference in their functions. Input gate i controls information flowing into memory cell c_t . Forget gate f controls information of last memory cell c_{t-1} accumulated in the current memory cell c_t . And output gate o influences information flowing into hidden state h_t . With the system of gating units, the gradient is well controlled avoiding the gradient vanishing or exploding problem.

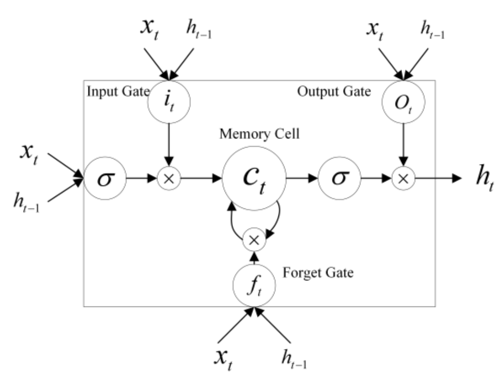

Fig. 1. Long Short-term Memory Cell

The conventional RNN is a forward network which can only utilize previous input data. As for the input date sequence $\mathbf{x} = (x_1,...,x_T)$, the single cell can only make use of the input data from t=1 to t ,but not the data from t=t+1 to T. In order to make full use of the total input data, bidirectional RNNs (BRNNs) structure is proposed [16]. BRNNs make the input data sequence in two directions to be respectively fed into two hidden layers as showed in Fig. 2. In comparison to conventional RNN, a BRNN adds a backward layer which computes backward input hidden sequence $\bar{h}_t$ from t=T to 1. Replacing the RNN cell by LSTM cell in BRNNs, the bidirectional LSTM (BiLSTM) is given [17], which can exploit long-range information in both input directions.

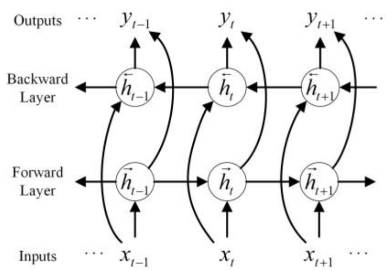

Fig. 2. Bidirectional Recurrent Neural Network

Because deep architectures make the network model learning higher level representations of the raw input data, deep architectures are widely used and usually have good performance. Stacking the BiLSTM by connecting output of lower layer to the input of the higher layer gives the deep BiLSTM [13] as illustrated in Fig. 3.

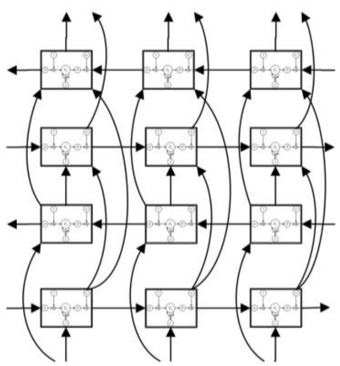

Fig. 3. Deep Bidirectional Long Short-Term Memory Network

B. Learning Network Model

The learning process of the network model is to find the optimal parameter (weights and bias) to minimize the loss function. RUL^t_{est} denotes the estimated RUL of the model with input data at time t, and RUL^t_{act} denotes the actual RUL which is recorded during the experiment. And the loss function is to describe the distance between RUL^t_{est} and RUL^t_{act} . So minimizing loss function means making the estimated RUL of the model as closed as possible to the actual RUL. In this paper the Root Mean Square Error (RMSE) is used as the losing function as follow:

$$J = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(RUL^t_{est} - RUL^t_{act}\right)^2} \tag{8}$$

During the training process, one problem is over fitting that should be paid attention. As parameters of model are only optimized according to training data, the trained model easily over fits the train data and has bad performance on the test data. So during the training process, we can use some tricks such as dropout and L2 regularization to improve the generalization ability of network.

In addition, the amount of data is usually huge and the parameters of model are in great quantity, so GPU cards are better used to speed up process of training with parallel computing technique.

C. Data Prepration

In actual situation, the final acquired data usually contains raw sensor data, work conditions, experiment units, time loop numbers and so on. Because the value scale of raw sensor data of different sensor may vary, it is need normalized before using. Also, the work conditions may be continuous or discrete, so it also should be normalized and be integrated into RUL estimated model. There are many normalization methods, of which the z-score and min-max normalization are commonly used.

z-score normalization:

$$x_i^{'} = \frac{x_i - u_i}{\sigma_i} \tag{9}$$

min-max normalization:

$$x_i^{'} = \frac{x_i - \min x_i}{\max x_i - \min x_i} \tag{10}$$

where u_i denotes the mean of i -th raw sensor data x_i , σ_i denotes the corresponding standard deviation and $x_i^{'}$ is the normalized data. With the normalization method, input data can be scaled into a unified scale. In this paper, the min-max normalization method is used to scale the raw data within range of $[0,1]$.

D. Model Evalution

The performance of RUL estimation model on test data needs to be measured by some indicators. One indicator is a RUL scoring function which usually used in other papers and the PHM competitions [5]. The definition of RUL scoring function is shown as below:

$$S = \begin{cases} \sum_{i=1}^{n}(e^{-\frac{h_i}{13}} - 1), h_i < 0 \\ \sum_{i=1}^{n}(e^{\frac{h_i}{10}} - 1), h_i \ge 0 \end{cases} \tag{11}$$

where $h_i = RUL_{est,i} - RUL_{act,i}$, and n is the total number of test data samples. Also, the RMSE indicator is commonly used as evaluation indicator. The difference of the two commonly used indicators is shown in Fig. 4.

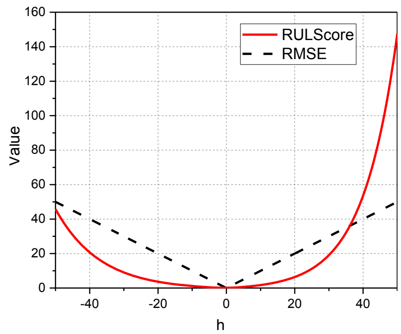

Fig. 4. Comparision of RUL Scoring Function and RMSE

The two indicators both get zero at $h = 0$ which means the predicted RUL is equal to real RUL. And with h far from 0 which means the difference of predicted RUL and real RUL being larger, and the indicator values also becoming larger. So, the smaller value, the better performance of estimated model is.

III. Experiment Study

In this experiment section, the deep BiLSTM model for RUL estimation is used on C-MAPSS Data Set. The data preparation details, architecture of network model, and the correlating hyper parameters are demonstrated. Then we compare deep BiLSTM model with other models such as Multi-Layer Perceptron (MLP), Support Vector Regression (SVR), Relevance Vector Regression (RVR), Convolutional Neural Network (CNN) [8] and LSTM [12].

A. C-MAPSS Dataset

Full name of C-MAPSS is Commercial Modular Aero-Propulsion System Simulation which is developed by NASA. The data set is about turbofan engine degradation which is a widely used as benchmark data set of RUL estimation [18].

TABLE I. C-MAPSS Dataset

Dataset	C-MAPSS			
	FD001	***FD002***	***FD003***	***FD004***
Units of training	100	260	100	249
Units of testing	100	259	100	248
Operating condition	1	6	1	6
Faults modes	1	1	2	2

C-MAPSS dataset has 4 sub datasets with different operation conditions and fault patterns. Each sub dataset has a training dataset and a testing dataset. The details of dataset are shown in Table I. The training dataset is the full life cycle data, but testing dataset doesn't reach end of full life which is used to evaluate the performance of models. The dataset contains 26 columns: 1st column is engine unit, 2nd column is the current cycle number, 3-5 columns are the working conditions and 6-26 columns represent 21 raw sensor data [18]. However, some columns don't provide useful information for RUL estimation. Therefore, 14 of the 21 sensors data are picked out as the raw input feature to the model [9]. The picked columns are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20 and 21.

Fig. 5 shows one example of 14 raw sensor data with entire life time cycle. It is difficult to find the mapping relation of raw sensor data and degraded life time. Before feed into the model, the data is normalized with min-max normalization method (Eq. (10)).

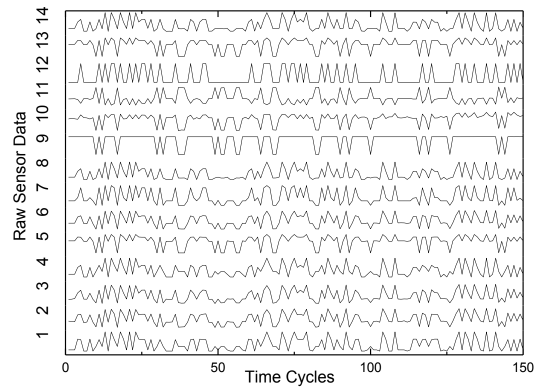

Fig. 5. Illustration of one traning sample with 14 picked raw sensor data

B. RUL Target Function

RUL is the target value of the estimation model. Traditionally, RUL is considered as a linear value decreased with time. In practice, however, the degradation of the machine performance is not obvious at the begin of the entire life. So at the beginning period, the RUL can be considered as constant, and when the degradation happens at the end of life the RUL can be considered as a linear function. So, a piece-wise linear RUL has been proposed in paper [5] [8]. In this paper, the maximum RUL limit constant is set as 125 time cycles, as shown in Fig. 6.

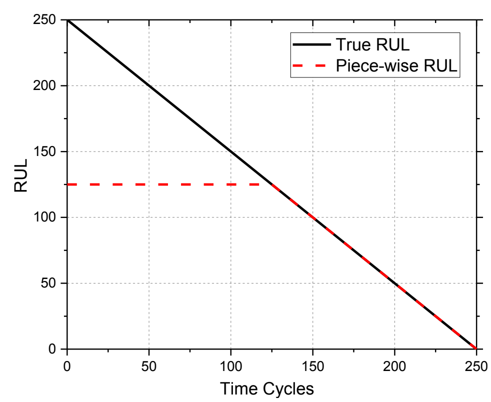

Fig. 6. Piece-wise RUL Function of C-MAPSS Dataset

C. Experiment Result

The flow chart of the proposed RUL estimation method is presented in Fig. 7. The picked 14 sensors data are feed into the designed BiLSTM model after min-max data normalization pre-processing. Then the model is trained with the back-propagation algorithm and evaluated by the RMSE and RUL score after training arrived at the maximum epochs. If the performance is not good enough, the parameters of the BiLSTM model is modified and do the training processing until the good performance is got. In this experiment, the maximum number of training epochs is set to 300 by default.

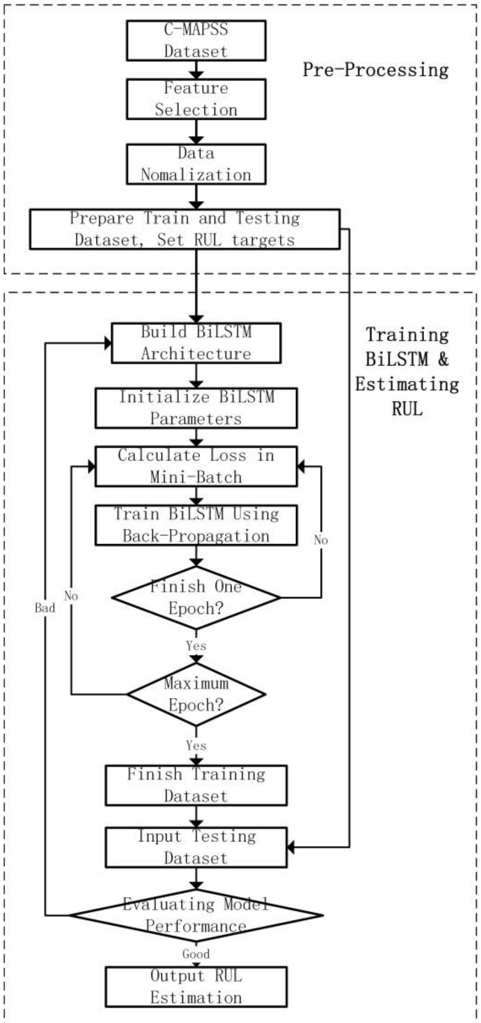

Fig. 7. Flow chart of the proposed method for RUL estimation

The architecture shown in Fig. 8 performs best which has 64 nodes in the first BiLSTM layer, 32 nodes in the second BiLSTM layer, 16 and 8 nodes in the third and fourth full connected standard feed forward neural network layers and the output layers with 1 node for RUL estimation. The activation function of the full connected standard feed forward neural network layers is Relu function.

Fig. 9 shows the hidden features extracted from the sensor data by the second BiLSTM layer. As the nodes number of second BiLSTM layer is 32 with both forward and backward layer, the number of output features is 64. Because the output activation function of LSTM is Tanh function, the values are in range of [-1,1]. The Fig. 9 has 165 columns representing RUL time cycles from 165 to 0. We can find earlier than 125 RUL cycles, the value of 64 features keep unchanged as the RUL is set as constant of 125 when the RUL is larger than 125. When the RUL is less than 125 cycles, the features begin gradually evolving over time cycles. The trends along the time cycles in extracted features are much more clear than the raw sensor data.

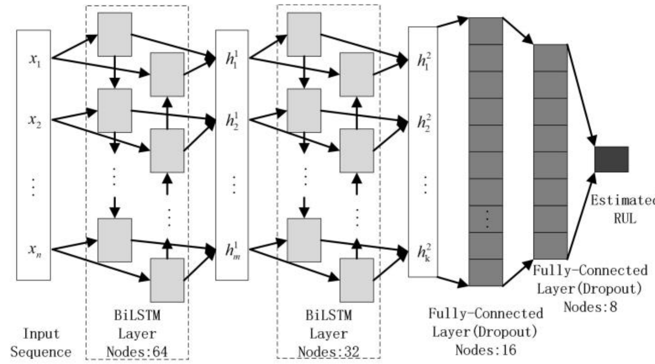

Fig. 8. Proposed Deep BiLSTM architecture for RUL estimation

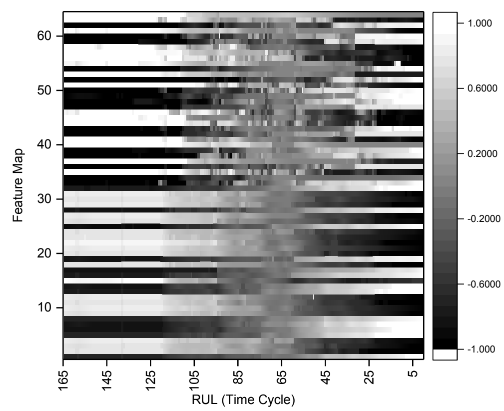

Fig. 9. Visualization of extracted features of 2nd hidden layer output

The RUL estimation result of the 4 C-MAPSS sub dataset is shown in Fig. 10. It can be observed that in the early periods in the 4 cases, the proposed model estimate the RUL closed to the constant. Then the estimated RUL comes to linear degradation, and the RUL^{t}_{est} is closer to RUL^{t}_{act} when the RUL goes to the end. This is because the closer to the end of RUL, the more data the model can use. And furthermore, the fault features are more clear as the failures are more serious at the end time as shown in Fig. 9.

The Table II and III shows the result of proposed BiLSTM approach network model comparison with other reported results. Both RMSE and RUL score are calculated to evaluate the performance. The results show BiLSTM approach model outperforms all other approaches.

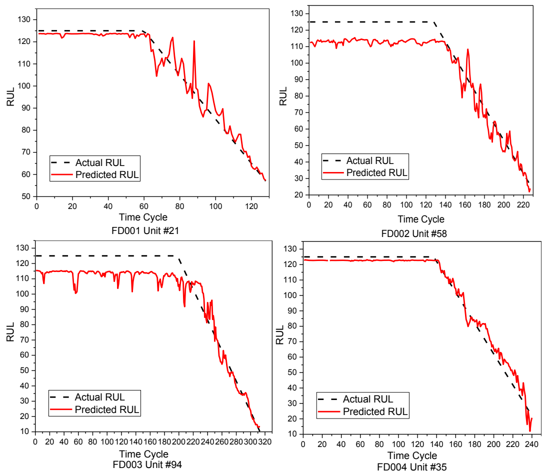

Fig. 10. Four examples of RUL estimation result

TABLE II. RMSE COMPARISON ON C-MAPSS DATASET

Model Method	C-MAPSS			
	FD001	*FD002*	*FD003*	*FD004*
MLP[8]	37.56	80.03	37.39	77.37
SVR[8]	20.96	42.00	21.05	45.35
RVR[8]	23.80	31.30	22.37	34.34
CNN[8]	18.45	30.29	19.82	29.16
LSTM[12]	16.14	24.49	16.18	28.17
BiLSTM	**13.65**	**23.18**	**13.74**	**24.86**

TABLE III. RUL SCORES COMPARISON ON C-MAPSS DATASET

Model Method	C-MAPSS			
	FD001	*FD002*	*FD003*	*FD004*
MLP[8]	1.80×10^4	7.80×10^6	1.74×10^4	5.62×10^6
SVR[8]	1.38×10^3	5.90×10^5	1.60×10^4	3.71×10^5
RVR[8]	1.50×10^3	1.74×10^4	1.43×10^4	2.65×10^4
CNN[8]	1.29×10^3	1.36×10^4	1.60×10^4	7.89×10^3
LSTM[12]	3.38×10^2	4.45×10^3	8.52×10^2	5.55×10^3
BiLSTM	$\mathbf{2.95\times10^2}$	$\mathbf{4.13\times10^3}$	$\mathbf{3.17\times10^2}$	$\mathbf{5.43\times10^3}$

IV. CONCLUSION AND FUTRUE WORK

In this paper, bidirectional LSTM (BiLSTM) approach for Remaining Useful Life (RUL) estimation is proposed and its benefits of taking sequence data in bidirection. The C-MAPSS dataset is used to verify the BiLSTM approach model outperforms other approaches. Although we have found the benefits of using the proposed BiLSTM approach model, further architecture optimization is still necessary. Moreover, some areas such as automatic piece-wise function detection, better optimization functions, high frequency sensor data, online learning models and so on should be paid effort to enhance the performance result of RUL estimation.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (U1534204, U11572206, U11790282, U11672104, U11225212) and the Natural Science Foundation of Hebei Province (A2016210099).

REFERENCES

[1] A. Azadeh, S. M. Asadzadeh, N. Salehi, and M. Firoozi, "Condition-based maintenance effectiveness for series–parallel power generation system—A combined Markovian simulation model," Reliability Engineering & System Safety, vol. 142, pp. 357-368, 2015.

[2] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications," Mechanical Systems & Signal Processing, vol. 42, no. 1-2, pp. 314-334, 2014.

[3] Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining Useful Life Prediction of Aircraft Engine based on Degradation Pattern Learning," Reliability Engineering & System Safety, vol. 164, 2017.

[4] M. Pecht and J. Gu, "Physics-of-failure-based prognostics for electronic products," Transactions of the Institute of Measurement & Control, vol. 31, no. 3, pp. 309-322, 2009.

[5] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in International Conference on Prognostics and Health Management, 2008, pp. 1-6.

[6] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," Mechanical Systems & Signal Processing, vol. 21, no. 1, pp. 193-207, 2007.

[7] Z. Tian, An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. Springer-Verlag New York, Inc., 2012, pp. 227-237.

[8] G. S. Babu, P. Zhao, and X. L. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," in International Conference on Database Systems for Advanced Applications, 2016, pp. 214-228.

[9] 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, 2018.

[10] P. Baruah and R. B. Chinnam, "HMMs for diagnostics and prognostics in machining processes," International Journal of Production Research, vol. 43, no. 6, pp. 1275-1293, 2005.

[11] F. A. Gers and N. N. Schraudolph, Learning precise timing with lstm recurrent networks. JMLR.org, 2003, pp. 115-143.

[12] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," in IEEE International Conference on Prognostics and Health Management, 2017, pp. 88-95.

[13] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM," in Automatic Speech Recognition and Understanding, 2014, pp. 273-278.

[14] Ian Goodfellow , Yoshua Bengio , and A. Courville, Deep Learning. MIT Press, 2016.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[16] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[17] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5-6, pp. 602-610, 2005.

[18] E. Ramasso and A. Saxena, "Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets," International Journal of Prognostics and Health Management, vol. 5, no. 2, pp. 1-15, 2014.