A Novel Framework for Machine Remaining Useful Life Prediction Based on Time Series Analysis

Tao Song[a] Chao Liu[a,b,†] Dongxiang Jiang[a,c]
songd17@mails.tsinghua.edu.cn cliu5@tsinghua.edu.cn jiangdx@tsinghua.edu.cn
[a]Department of Energy and Power Engineering
[b]Key Laboratory for Thermal Science and Power Engineering of Ministry of Education
[c]State Key Laboratory of Control and Simulation of Power System and Generation Equipment
[a,b,c]Tsinghua University, Beijing 100084, China

***Abstract*—The use of Deep Neural Network (DNN) has made great strides in data-driven engineering maintenance and prognostics. As a powerful neural network structure, recurrent neural network (RNN) has achieved very good results in the remaining useful life (RUL) estimation problem, because of its extraction of long-term dependency information from the historical sensor signal. This work focuses on a two-step method containing two RNN models for RUL estimation. A series prediction model is trained for predicting subsequent signal series from existing sensor signal sequences, whereas another model is trained for estimating RUL from the combination of the existing signal sequences and the predicted signal series. Better results are obtained by this method on FD-001 and FD-003 in NASA C-MAPSS dataset.**

I. Introduction

As of the development of maintenance strategy in machinery, condition based maintenance (CBM) is largely built on the assessment of the health status of the equipment, where the state estimation and remaining useful life (RUL) prediction are two crucial topics in the community. The existing approaches for RUL prediction can be categorized into model based, data driven, and hybrid approaches. Model based RUL prediction methods can be implemented by forming the degradation models by exploring the failure mechanism in physical principles, or the statistical models that can capture the deterioration tendencies. These model needs intensive analysis on the specific components of the equipment including the structure details, the operating conditions, material properties, interaction with other components in the system, etc. As a result, model based RUL prediction approaches are specific and may achieve high accuracy as of the comprehensive understanding of the mechanisms. However, they need lots of inputs in terms of costs, and cannot be applied in a different situation because of the changes of the circumstances, which make model-driven approaches are less applicable in varying scenarios.

Data-driven RUL prediction methods attracts more and more attention as of its applicability in different scenarios, which needs less inputs for transferring the method to another application. Also, there are intensive research on the model generalization and transferability for data driven models among different applications or scenarios [1], [2], [3], [4], [5]. Generally, data driven algorithms for RUL prediction mostly consider it as a regression problem, where the inputs are the monitoring parameters (the latest state, the deterioration tendency, etc.) and the target is to get the inputs projected to the historical failure times. Neural network is a commonly used data-driven method, and its various derivative models are applied to RUL prediction problems, such as DNN[6], RNN[7] and CNN[8]. Data-driven method provides general and accurate approaches for RUL estimation.

† Corresponding author.

One of the main drawbacks for the aforementioned methods is that the error accumulates as the time goes on, which makes the RUL prediction is less accurate for the long-term prediction. In this context, the idea presented in this work is, the RUL prediction may perform better, if we can foresee the states of the equipment in the next several time stamps. In this context, we formulate the RUL prediction problem as a bi-level optimization problem, that the first level is the time-series forecasting to get predicted sequences beyond the current time, and the second level is the regression for the RUL given the predicted sequences.

The contributions of this work include: 1) We analyzed the trend of the decay signal sequence and the life prediction error, and proposed the sequence prediction error analysis hypothesis. 2) Based on this hypothesis, we designed a bi-level Long Short-term Memory (LSTM) framework for RUL prediction, and the accuracy was improved on the FD001 and FD003 in NASA C-MAPSS dataset [9].

The remaining sections are organized as follows. Section II provides the proposed framework for RUL prediction including preliminaries of long short term memory (LSTM) networks, the problem formulation, and the flowchart of the proposed framework. Section III demonstrates the proposed method specifically in experiment and the obtained result comparing with other existing methods. Finally, the paper is summarized and concluded in Section IV.

II. Methodology

A. RNN and long short-term memory

Our work is based on a special neural network structure, Long Short-term Memory (LSTM), a variant from Recurrent neural network (RNN). A considerable amount of literature [10], [11], [7] has proved that the extraction of long-term

978-1-7281-0860-5/19/$31.00 ©2019 IEEE

dependency information by RNN units (or LSTM units) can effectively enhance the model's ability to fit historical data, and thus improves the prediction accuracy. And LSTM has been successfully applied in diverse domains and outperforms or is comparable to state-of-the-art approaches [12], [13], [14], [15].

RNN is a special kind of artificial neural network which can obtain temporal dynamic behavior. In contrast to Feed-forward neural network (FNN) where information can only be passed between layers in one direction, RNN has recurrent hidden states establishing association between the output and the previous input.

Proposed by Hochreiter and Schmidhuber[16], LSTM is one of the most widely used RNN variants. solving the problem of exploding and vanishing gradient encountered by traditional RNNs. At the time step t, a memory cell c_t is maintained in LSTM by encoding memory of observed input information, whose behavior is determined by three gates: input gate i_t , output gate o_t and forget gate f_t. The following equations indicate the updated behavior:

$$\begin{aligned} i_t &= \operatorname{sigmoid}\left(U_i h_{t-1} + W_i x_t + b_i\right) \\ f_t &= \operatorname{sigmoid}\left(U_f h_{t-1} + W_f x_t + b_f\right) \\ o_t &= \operatorname{sigmoid}\left(U_o h_{t-1} + W_o x_t + b_o\right) \\ \widetilde{c}_t &= \tanh\left(U_c h_{t-1} + W_c x_t + b_c\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t \\ h_t &= o_t \odot \tanh\left(c_t\right) \end{aligned} \tag{1}$$

where parameters $U \in \mathbb{R}^{d\times d}, W \in \mathbb{R}^{d\times k}, b \in \mathbb{R}^{d}$ can be learned throughout training, d is the hidden layer size and the operator $\odot$ denotes the element-wise multiplication. Learning the memory control behavior of the three gates, LSTM can capture important information even if there is unknown duration between significant events in the input sequence.

B. Problem formulation

As discussed in the Introduction section, the existing RUL models are mostly designed to estimate RUL directly from the several time-step of the sensor signal, a bi-level optimization problem is defined in this work, including (1) time-series forecasting, (2) RUL regression based on the predicted sequences. More formally, the RULs Y_{RUL} are defined as a function of predicted sequences $g\big(\mathbb{X}\big)$, where $g\big(\mathbb{X}\big)$ is a function of the observed time-series $\mathbb{X}(\approx) = \{x_1(t), x_2(t), \cdots, x_f(t)\}$, f is the number of observed parameters.

$$Y_{RUL} = f\big(g\big(\mathbb{X}(\approx)\big)\big) = f\big(g\big(\{x_1(t), x_2(t), \cdots, x_f(t)\}\big)\big) \tag{2}$$

In the training phase, a minimization problem is formulated as follows, which is intended to find out the best approximation of the prediction function $f\big(g\big(\mathbb{X}_{\approx\setminus}(\approx)\big)\big)$ that obtains a minimal error between the predicted values and the ground truth $Y_{RUL,tr}$.

$$\min_{N}\left(f\big(g\big(\mathbb{X}_{\approx\setminus}(\approx)\big)\big) - Y_{RUL,tr}\right) \tag{3}$$

The bi-level optimization problem described in Eq.3 is not computationally tractable to get an exact solution for this optimization problem, especially for large systems. Therefore, a two-step strategy is presented to approximate the problem.

1) **Series prediction.** A series prediction model is trained for predicting subsequent signal series from existing signal sequences. As the device operates, the data measured by the sensor forms a time series, which contains device running status information. During the aging process of the equipment, the measured data will show a certain decline regularity, which can be existed by the trained RNN model. Therefore, the sensor signal data with the regularity can be predicted.
2) **RUL estimation.** Another model is trained for estimating RUL from the combination of the existing signal sequences and the predicted signal series. This model is similar to the existing models except that the input data is different.

From the perspective of error accumulation, two types of errors are included in the presented bi-level optimization problem, time-series forecasting error and RUL estimation error.

1) **Time-series forecasting error:** $E_{ser-pre}$**.** The time series estimate error increase as the predicted sensor signal time steps increases, which means the error can be controlled within a small range in the case where the prediction time step is small. The relationship between series predict error and the predicted sensor signal time steps can be drawn as Figure 1 (a).
2) **RUL estimation error:** $E_{RUL-est}$**.** The RUL estimate error decreases in the equipment's life cycle, which means RUL can be estimated more accurately near the end of the life of the device. The relationship between the RUL estimate error and equipment life time step can be drawn as Figure 1 (b).

Comparing with the single model RUL-estimator, the two-step method predicts sensor signal data time series before RUL estimate, leading to series predict error ($\Delta E_{ser-pre} \geq 0$). On the other hand, the combination of the existing signal sequences and the predicted signal series gets closer to the end of life of the device, resulting in less RUL estimate error ($\Delta E_{RUL-est} \leq 0$).

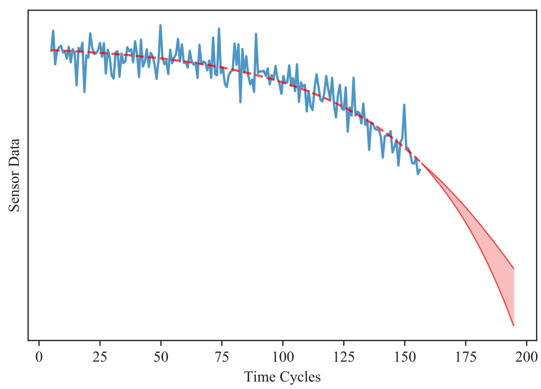
Based on the above analysis of error assumptions, the error caused by the two-step method can be characterized as a trade-off between the increased error of time-series forecasting error and decrease of RUL estimation error.

$$\Delta E = \Delta E_{ser-pre} + \Delta E_{RUL-est} \begin{cases} \Delta E_{ser-pre} & \geq 0 \\ \Delta E_{RUL-est} & \leq 0 \end{cases} \tag{4}$$

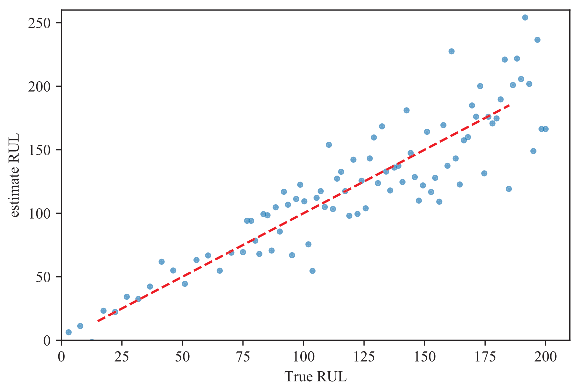
The proposed two-step method works if the models can be obtained with the following condition satisfied.

$$\Delta E \leq 0 \rightarrow |\Delta E_{ser-pre}| \leq |\Delta E_{RUL-est}| \tag{5}$$

In other words, the reduction of the final RUL estimation error can neutralize the increase of the error caused by the sequence



(a) Time-series forecasting error



(b) RUL prediction error

Figure 1. Schematic diagrams for the hypotheses of time-series forecasting error and RUL estimation error

prediction. As the error accumulation characteristics presented on both time-series forecasting and RUL estimation processes, the proposed two-step framework makes sense as it narrows the error of each process and would harvest predictions that with smaller errors.

C. Bi-level LSTM framework for RUL prediction

For the bi-level LSTM framework for RUL predictions consists of two steps:

1) **LSTM model for time-series forecasting**. A deep structure with LSTM networks is defined for multivariate time-series forecasting, where the inputs are the time-series data till the current time and the outputs are the time-series in the next several steps.
2) **LSTM model for RUL estimation**. An LSTM regression structure is formed for RUL predictions, where the inputs are the predicted time-series data, and the output is the predicted RUL. The LSTM structure consists of a few LSTM layers to capture the recurrent featurs along the time-series data and several fully connected layers to approximate the relationship between the learned recurrent features and the target RUL values.

To validate the proposed framework, case studies on a turbofan engine degradation data set is carried out in the following section.

III. Results and Discussions

A. Dataset

The machine degradation widely exists in diverse fields, and this work adopts a turbofan engine deterioration data set for validate the efficacy of the proposed approach. The turbofan engine degradation dataset [9], [17], is consisted of simulated data from a model-based simulation program developed by NASA, i.e. Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). Since its publication in PHM data challenge in 2008, this dataset has been used as a benchmark for RUL prediction algorithms and different kinds of methods are proposed with the purpose of increasing prediction accuracy and reducing the uncertainty[9]. Among the four sets of degradation data included in C-MAPSS dataset, FD001 and FD003 are simulated with less operating conditions and are supposed to present similar degradation trends during the service life. FD002 and FD004 contain more operating conditions which makes the time-series forecasting difficult in the proposed framework. As a result, the two datasets are not applied in this work. In the future work, the operating conditions can be isolated in the aforementioned operating conditions, and performance on all of the data will be evaluated and compared.

FD002 and FD004 are both composed of one training set and one test set. Each of the sub-dataset includes multivariate temporal data from several different engine models collected by 21 sensors under different operational conditions and fault modes. The detailed description of the dataset is listed in Table I.

TABLE I: Information for C-MAPSS dataset

Sub-dataset	FD001	FD003
Engine units for training	100	100
Engine units for testing	100	100
Operating conditions	1	1
Fault modes	1	2

B. Data preparation

The raw data from sensors is not suitable for feeding directly to the model, because of its various value scale, data redundancy, etc. Data preprocessing should be conducted before model training.

1) Min-Max normalization: Since the raw signal data from C-MAPSS are collected by 21 different sensors, the value scale of the sensor data may be various. It is necessary to normalize the data from each sensor. Min-max normalization is adopted to the raw data, scaling it within the range of $[0, 1]$, which is demonstrated in equation 6 where $\mathbf{x}_i$ is the raw data from the ith sensor.

$$\mathbf{x}'_i = \frac{\mathbf{x}_i - \min\ \mathbf{x}_i}{\max\ \mathbf{x}_i - \min\ \mathbf{x}_i} \tag{6}$$

2) Principal components analysis and selection: Whilst the multivariate data consists of 21 time series from different sensors, there is similarity and redundancy between the sequences, which not only cannot provide valuable information for the RUL estimation, but also brings difficulties to sequence prediction. Only 14 sensors out of the 21 sensors are used in FD001, between which there is still a correlation found by observation.

In order to eliminate the impact of redundant data on prediction and estimation, principal component analysis (PCA) is performed on the data. Meanwhile, three signal components with good trend and low noise are selected for final model input. Specifically, the first, third and fourth signal components of the sub-dataset FD001 and the first, second and fifth signal components of the sub-dataset FD003 are selected. The process of PCA and signal components selection transforms the multivariate data from $s \in \mathbb{R}^{L\times 24}$ to $s \in \mathbb{R}^{L\times 3}$.

3) Time window processing: For LSTM model, it is significance to capture temporal dependencies through data sequences. However, long sequence input can lead to problems such as too few training sequences and difficulty in dependence extraction. In our experiments, a time window of length t is adopted to the data, which changes it to $s \in \mathbb{R}^{t\times 3}$. The length t is optimized as a hyperparameter to $t_1 = 64$ for Series predict and $t_2 = 30$ for RUL estimate.

C. LSTM networks and parameters

To implement the two-step estimate method, two models were established, which both have LSTM layers following the inputs and fully connected layers before the output.

Here, the full time-series signal $s \in \mathbb{R}^{L\times n}$ collected throughout the life cycle (the whole process from the healthy condition to the failure of the equipment), is often visible to the estimate model when training, while the model can only observe the first part of the data $s \in \mathbb{R}^{l\times n}$ and estimate the RUL $T_{RUL} \approx L - l$ when testing and application.

Series predict model is built to predict next d steps of subsequent signal series from feeding sequences data, i.e. predict $s \in \mathbb{R}^{d\times 3}$ from the input series $s \in \mathbb{R}^{64\times 3}$, where predict length $d = 4$ is the result of hyperparameter optimization. RUL estimate model is set up to estimate RUL from the input time window, i.e. Fit the RUL value $T_{RUL} \in \mathbb{R}$ from the input signal $s \in \mathbb{R}^{30\times 3}$. Specifically, the internal structure of the model is indicated in the Figure2.

Training and testing

Both two models are trained with Adam optimizer proposed in [18], dropouts, piecewise linear degradation RUL label proposed in [7] and MSE training metrics defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left[(T_{RUL})_{\text{Pre}} - (T_{RUL})_{\text{True}}\right]^2 \qquad (7)$$

In addition to training the two models separately, the combined data is fed to the RUL predict model, which is composed of the origin data and the time sequence data generated by the series predict model, promoting coupling between the two models.

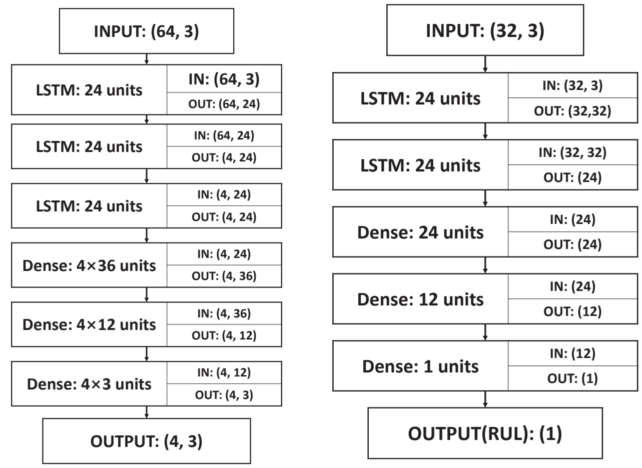


Figure 2. Network structure for series predict model and RUL estimate model

For testing, the preprocessed raw data $s_{[1,l]} \in \mathbb{R}^{l\times 3}$ gets sliced to $s_{[l-63,l]} \in \mathbb{R}^{64\times 3}$, which is sent to the series predict model to generated next $d = 4$ steps data $s_{[l+1,l+4]} \in \mathbb{R}^{4\times 3}$. The combined data $s_{[l-25,l+4]} = \left[s_{[l-25,l]}, s_{[l+1,l+4]}\right] \in \mathbb{R}^{30\times 3}$ is fed to the RUL estimate model to obtain RUL T_{RUL}.

D. Performance evaluation

To compare the performance of the proposed algorithm with state-of-the-art approaches, Root Mean Square Error (RMSE) is applied, and it is defined by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[(T_{RUL})_{\text{Pre}} - (T_{RUL})_{\text{True}}\right]^2} \qquad (8)$$

Through hyperparameter searching, the performance of the proposed framework is obtained and listed in Table II, as well as the state-of-the-art approaches reported in [19]. As listed

TABLE II: Performance comparisons of different methods on the two C-MAPSS sub-dataset characterized by RMSE

Sub-dataset	FD001	FD003
NN	14.80	15.22
DNN	13.56	13.93
RNN	13.44	13.36
LSTM	13.52	13.54
DCNN	12.61	12.64
RUL estimation only (with LSTM)	11.95	13.79
Bi-level LSTM prediction	11.67	12.42

in Table II, RUL prediction with LSTM (the first colored row) performs well in dataset FD001, while it is not as good as DCNN for dataset FD003. Withe the bi-level LSTM framework (the last row), the RMSEs on datasets FD001 and FD003 are smaller that the state-of-the-art approaches, and this validates the efficiency of the proposed framework. It implies that the combination of the errors induced by time-series forecasting and RUL estimation is less than that of the

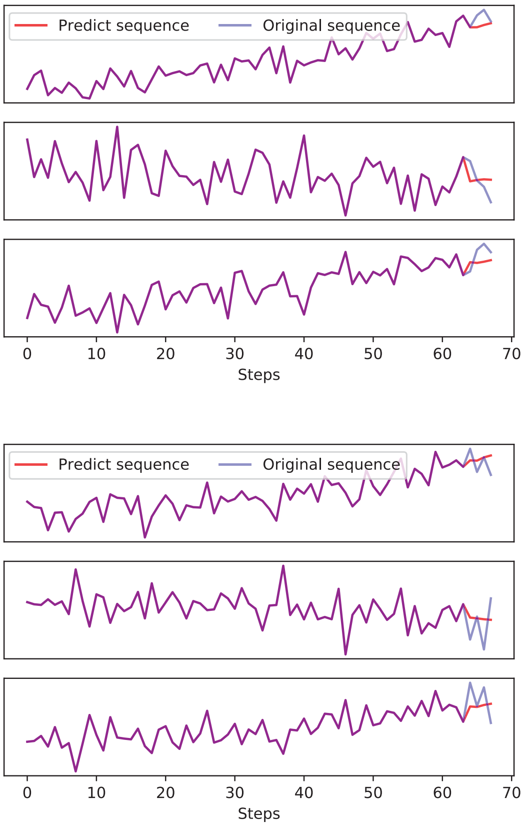


Figure 3. Comparison between 4-step series predict sequence and original sequence from #35 and #90 in FD001 test set

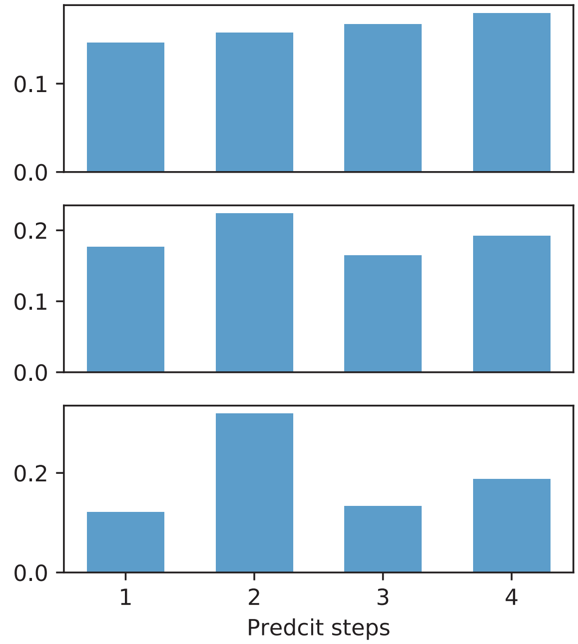


Figure 4. Mean error of 4-step predict for FD001 test set

RUL estimation with LSTM and other existing approaches reported in [19].

To further analyze the error terms by time-series forecasting and RUL estimation, results on the time-series forecasting are shown in Figs. 3-4, and RUL predictions are shown in Figure 5.

As shown in Figure 3, the time-series forecasting basically captures the degradation trends of the extracted components from PCA, which are shown to be effective indicators for the engine deterioration. For the average error of the 4-step time-series forecasting shown in Figure 4, the top panel shows that the forecasting accuracy decreases as the time goes on. However, as of the complexity of the engine system and the influence of noise injection in the simulation model, the maximal forecasting error may not occur during the longest prediction step, although the minimal forecasting error mostly happens on the first step.

Enjoying the privilege of the time-series forecasting (as shown in Figs. 3 and 4), the RUL estimation error is significantly decreased. The RUL predictions and ground truth for dataset FD001 and FD003 are shown in Figure 5, which is sorted by the real RULs at different times of the test engine. For both FD001 and FD003, the error of RUL prediction decrease as the engine approaches the end of its life (for RULs close to 0 in the x-axis). And this verifies the hypothesis discussed in Section II-B (as shown in Figure 1) that the bigger the error, the longer the RUL.

E. Discussions

Time-series forecasting and RUL estimation have been widely analyzed in the community in a independent manner, this work intends to track the error accumulation in the RUL prediction process, and presents a novel way to suppress the RUL prediction error by forecasting the time-series data. Although it cannot be highly reliable and introduces new error term, the time-series forecasting plays significant effects on reducing the RUL estimation error. The case study in this section shows that the proposed framework outperforms the state-of-the-art approaches.

As more than one operating condition included in FD002 and FD004 of C-MAPSS dataset and PCA doesn't work well in this context, this work doesn't include the results on the two sets. In the future work, different time-series forecasting will be implemented to get a better prediction of time-series data in the case of diverse operation conditions. Also, applications of the proposed algorithm on degradation of different machines will be studied.

IV. Conclusions

The RUL prediction is essential in prognostics and health management, and this work presents a novel framework, the two-step LSTM prediction method, for the RUL prediction problem. By formulating the RUL prediction problem as a bi-level optimization problem, time-series forecasting and RUL

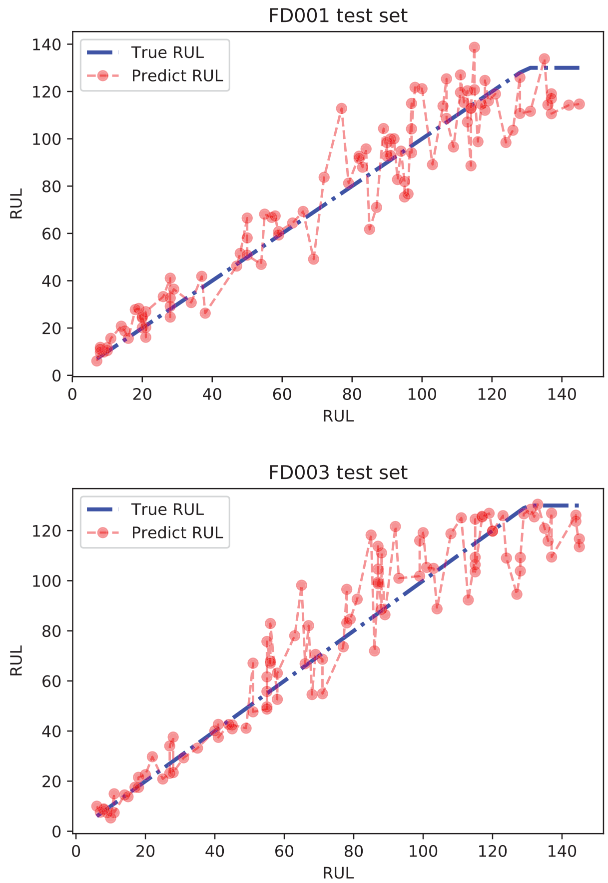


Figure 5. RUL estimate result for test set

estimation are formed for RUL prediction and LSTM networks are applied for the time-series forecasting and RUL estimation. By combining the two-step predictions, the case study on the turbofan engine dataset show that time-series forecasting properly captures the trends of the degrdation and boosts the RUL prediction with smaller RMSE error. Future work will be focused on applying the presented framework on more datasets.

V. Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 11802152, 11572167).

References

[1] T. Han, C. Liu, W. Yang, and D. Jiang, "Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions," *ISA Transactions*, 2019.

[2] T. Han, C. Liu, L. Wu, S. Sarkar, and D. Jiang, "An adaptive spatiotemporal feature learning approach for fault diagnosis in complex systems," *Mechanical Systems and Signal Processing*, vol. 117, pp. 170–187, 2019.

[3] T. Han, C. Liu, W. Yang, and D. Jiang, "A novel adversarial learning framework in deep convolutional neural network for intelligent diagnosis of mechanical faults," *Knowledge-Based Systems*, vol. 165, pp. 474–487, 2019.

[4] W. Yang, C. Liu, and D. Jiang, "An unsupervised spatiotemporal graphical modeling approach for wind turbine condition monitoring," *Renewable energy*, vol. 127, pp. 230–241, 2018.

[5] T. Han, C. Liu, W. Yang, and D. Jiang, "Deep transfer network with joint distribution adaptation: a new intelligent fault diagnosis framework for industry application," *arXiv preprint arXiv:1804.07265*, 2018.

[6] N. Gebraeel, M. Lawley, R. Liu, and V. Parmeshwaran, "Residual life predictions from vibration-based degradation signals: a neural network approach," *IEEE Transactions on industrial electronics*, vol. 51, no. 3, pp. 694–700, 2004.

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

[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*. Springer, 2016, pp. 214–228.

[9] 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.

[10] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017.

[11] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla lstm neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[12] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder," *arXiv preprint arXiv:1608.06154*, 2016.

[13] J. Lei, C. Liu, and D. Jiang, "Fault diagnosis of wind turbine based on long short-term memory networks," *Renewable Energy*, vol. 133, pp. 422–432, 2019.

[14] Z. Jiang, C. Liu, N. P. Hendricks, B. Ganapathysubramanian, D. J. Hayes, and S. Sarkar, "Predicting county level corn yields using deep long short term memory models," *arXiv preprint arXiv:1805.12044*, 2018.

[15] N. Gugulothu, V. TV, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, "Predicting remaining useful life using time series embeddings based on recurrent neural networks," *arXiv preprint arXiv:1709.01073*, 2017.

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

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

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] 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.