

Deep Learning-Based Prognostics for Turbofan Engine Remaining Life Prediction

Md. Omar Faruk, Israt Jahan Keya, Md. Ashraful Islam*

Department of Mechanical Engineering, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh

ABSTRACT

In modern times, especially after the Industrial Revolution 4.0, artificial intelligence has become one of modern technology's most discussed and fascinating aspects. Artificial intelligence is now being incorporated into almost all aspects of our lives. Safety is an important field in which AI is now being implemented. Its decision-making capability from the numerical data makes it very reliable. In recent days, AI has been implemented in the predictive maintenance of various complex machines as well. This study aims to investigate the life prediction of turbofan engines. Deep learning and machine learning analyse the estimation of Remaining Useful Life (RUL). These have not only saved resources and time but also ensured greater safety to mankind. A reliable deep-learning model has been developed that can predict the RUL of a turbofan jet engine with very good reliability. The C-MAPSS dataset, which was published by NASA, was utilized to train the model. The deep learning model used here is the BiLSTM because it performs very well with sequential data and propagates both backward and forward directions. The dataset given by NASA has already been split into training and testing sets. After normalizing the data, the model has been trained with three BiLSTM layers, two fully connected layers, and one output layer. The result is quite satisfactory, with an R^2 value of 0.893728 and an RSME of 13.5468187.

Keywords: AI, BiLSTM, RUL, C-MAPSS, CNN, Normalizing.



Copyright @ All authors

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In any machining operation, safety is always the first priority. Every system requires maintenance at some point in its lifespan [1]. A direct investigation takes both a longer time and requires human resources. On the other hand, predictive maintenance uses sensor-based data to estimate the time-to-failure of a machine or equipment [2]. This method uses time-to-failure duration with sensor data, which contributes to minimized downtime, reduced cost, better safety and maximized machine lifetime.

Engines are the main powerhouse of an aircraft that produces the required thrust for the plane to fly. Turbofan jet engines are used in modern aircraft for their longer flight time and very little maintenance time. However, different operating conditions, weather conditions, combustion pressure, and other parameters can lead to variations in the remaining useful life of the engine. Therefore, predictive maintenance is required to exterminate unplanned downtime and lengthen the lifecycle of the engine [3].

The deep learning approach requires a deep understanding of the underlying physics of the life-to-parameters relation to form a mathematical model to predict the remaining life. This process incorporates different learning algorithms to find the pattern in the data. Each model has its strengths and weaknesses. So, it's wise to compare different models to find out which one predicts better. Generally, CNN, LSTM, GRU, etc., are used for various predictive tasks. It's a general rule of thumb that LSTM works better with sequential data, i.e., time-series data [4]. This research will focus on BiLSTM to predict RUL.

The dataset used for the research is the C-MAPSS dataset provided by NASA [5]. There are four types of datasets there. This research focuses only on the first dataset.

2. Literature review

James Carroll et al. [6] investigate the prediction of failure and remaining useful life (RUL) of gearboxes for modern multi - megawatt wind turbines using SCADA and vibration data. Three popular machine learning algorithms were trained and tested in this research: an ANN, a support vector machine (SVM), and a logistic regression approach. The ANN provides the most correct failures and least missed failures, with 72.5% of failures being correctly predicted out of the three algorithms tested.

Sukwon et al. [7] developed the regression, XGBoost, LightGBM, and DNN models to predict the life span of buildings using the demolition permit dataset. That model was considered more accurate, reducing the root mean square error (RMSE) and increasing the coefficient of determination (R^2). Among them, DNN was the most accurate with 95.5% prediction accuracy, which was 2.2% greater than the linear regression model, whereas XGBoost had weaker performance than the linear regression model.

Support Vector Machine, Random Forest, Decision tree, K-means and CNN can be used for feature classification. P. K. Ambadekar et al. [8] tested CNN on the tool wear images to predict the remaining useful life of a cutting tool. Depending upon tool wear, 1183 images were classified in either class A, class B or class C, and CNN was applied to these images, which provided an accuracy of 87.26%.

To predict the residual life of the tool, Shuo Wang et al. [9] used four prediction models which are the CNN-LSTM model, CNN-PSO model, and CNN-LSTM-PSO model, and found that the CNN-LSTM-PSO model showed the best accuracy with 1.3520 of RMSE value, 1.0892 of MAE value and determination coefficient R^2 value is 0.9961

considering when R2 value is closest to 1, provides the highest accuracy of prediction.

Xiang Li et al. [10] have proposed a new deep-learning method based on CNN. In this paper, the C-MAPSS dataset and two metrics, including scoring function and root mean square error, have been used to check the performance of this newly proposed method. Then, comparing this method with basic neural network (NN), deep neural network (DNN), recurrent neural network (RNN), and long short-term memory (LSTM), it is observed that the proposed method shows good prognostic performance with its simple architecture, lower computing load, and five convolution layers.

Rudra Karnik et al. [11] proposed an LSTM model for RUL prediction of aircraft turbofan using the CMAPSS dataset. This LSTM model consists of LSTM, dropout, and fully connected layers and activation functions, including tanh, sigmoid, and ReLu. Among four sub-datasets, FD001 and FD003 show more correlation with target RUL. So, these two sub-datasets are used to obtain the RMSE value. The proposed model (LSTM) shows the lowest RMSE value in comparison with SVR, CNN, Deep LSTM, Hybrid Model and CNN+LSTM, Which is 14.28 for FD001 and 16.15 for FD003.

Caifeng Zheng et al. [12] proposed a new method based on sliding time window (TW) and extreme learning machines (ELM) to predict the RUL of aircraft engines using the FD001 dataset. The performance of this proposed method was tested by comparing its RMSE and score function with other methods, including multilayer perceptron (MLP), random forest regression (RFR), long short-term memory (LSTM) and support vector machine (SVM). This proposed method showed the most efficient result, and its RMSE and score values are 13.78 and 267.31, respectively.

3. Problem statement

Conventional maintenance depends upon physical inspection, which leads to more downtime and increased costs. Failure in the turbofan jet engine while in flight can cause fatal accidents, leading to loss of lives and huge amounts of money. Therefore, a system is to be developed so that it can be predicted before the accident so that proper maintenance can be done [13]. With the rise of AI, the demand for more secure maintenance systems has shot up. Combined with the sensor data, these models can perform very well, given that the data provided is large enough to train.

4. Description of C-MAPSS turbofan jet engine dataset

The C-MAPSS turbofan jet engine dataset was originally developed and published by NASA back in 2008 at the international conference on PHM [5]. Despite being very old, this dataset is still being utilized for research purposes for predictive maintenance system development of turbofan jet engines. The dataset is divided into four units, FD001, FD002, FD003, and FD004, according to operating condition and failure mode. The dataset contains a total of 709 engines for training and 709 engines for testing as well. The parameters of the dataset are subdivided into two parts. The first is the operating conditions and the second is the sensor readings. There are three types of operating conditions and 21 sensor readings for each observation. The operating conditions are given in Table 1. The sensor parameters are shown in Table 2.

Table -1. Operating conditions

Operating conditions	Operating range
Mach number	0 to 0.90
Altitude	Sea level to 40,000 ft
Sea level temperature	-60 to 103° F

Table -2. Sensor parameters.

Sensor	Parameter	Description
1	T2	Fan inlet temperature
2	T24	Temperature at LPC outlet
3	T30	Temperature at HPC outlet
4	T50	Temperature at LPT outlet
5	P2	Pressure at fan inlet
6	P15	Pressure at bypass-duct
7	P30	Pressure at HPC outlet
8	Nf	Fan speed
9	Nc	Core speed
10	Epr	Engine pressure ratio
11	Ps30	Static pressure at HPC outlet
12	Phi	Ratio of fuel flow to Ps30
13	NRf	Corrected fan speed
14	Nrc	Corrected core speed
15	BPR	Bypass ratio
16	FarB	Burner fuel-air ratio
17	HtBleed	Bleed enthalpy
18	NF-dmd	Demanded fan speed
19	PCNR-dmd	Demanded corrected fan speed
20	W31	HPT coolant bleed
21	W32	LPT coolant bleed

5. Methodology

In this investigation, we have used a BiLSTM-based approach to estimate the RUL of turbofan jet engines. BiLSTM was selected over others because it performs remarkably well with sequential data, i.e., time series data, unlike most other models. The key difference between LSTM and BiLSTM is that LSTM is unidirectional but BiLSTM is bidirectional, which can propagate both in a forward and backward direction, making it more suitable for the operation. The model's accuracy depends on the number of layers and neurons in each layer. With hyperparameter tuning, optimum values, batch size, learning rate, dropout probability, etc. can be obtained. Using these increases the accuracy while decreasing RMSE significantly.

The steps of the model are described in the sub-sections.

5.1 Correlation analysis

First, we need to convert the dataset to an appropriate and useful form for further analysis. We need to put labels on the columns to identify them. Then, this data frame is passed into correlational analysis. Correlational analysis is carried out using the correlation coefficient 'r', which can be mathematically written as,

$$r = \frac{cov(x,y)}{s_x s_y} \quad (1)$$

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x})^2)(\sum_{i=1}^n (y_i - \bar{y})^2)}} \quad (2)$$

Here, we can find out the correlation of the features to the target feature. With the correlation coefficients, a heatmap was drawn which is known as correlation matrix that is given in Figure 1.

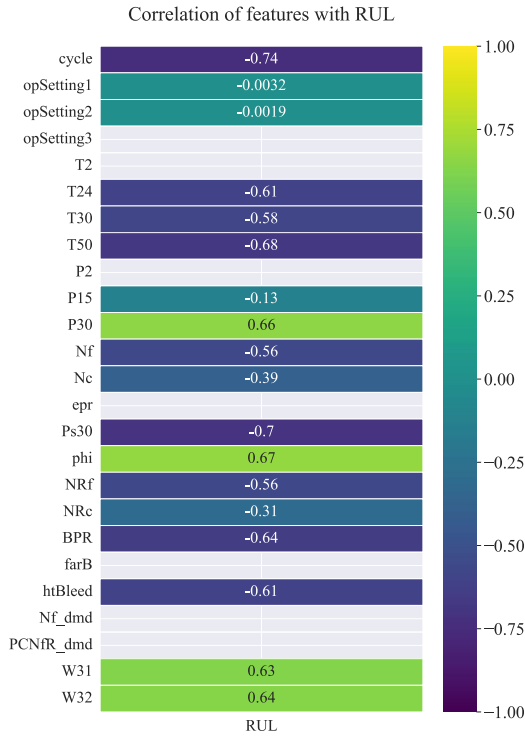


Figure – 1. Correlation matrix of the dataset.

5.2 Data Filtering

The correlation data is then analyzed using a median filter to filter the raw data. The features with very low correlation coefficients are excluded for the simplicity of the model [14].

From the correlation matrix, it can be easily visualized that certain parameters do not contribute significantly to the accuracy but rather decrease the model's performance. By using the median filter, features with a correlation coefficient close to zero are omitted. Parameters that are omitted are opSetting3, T2, P2, epr, farB, Nf_dmd and PCNfR_dmd.

The final correlational heatmap is given in Figure 2.

5.3 Data Normalization

The parameters in the dataset range differently for each parameter. There is no boundary in which the parameters vary. If this data is to be used for training, the resulting model will be very inaccurate [15]. Therefore, the data has to be normalized. We take the Z-score normalization, which is formulated as,

$$z = \frac{x_i - \mu}{\sigma} \quad (3)$$

Here, Z is the normalized data, x_i is the raw data, μ is the mean of all data, and σ is the standard deviation [16].

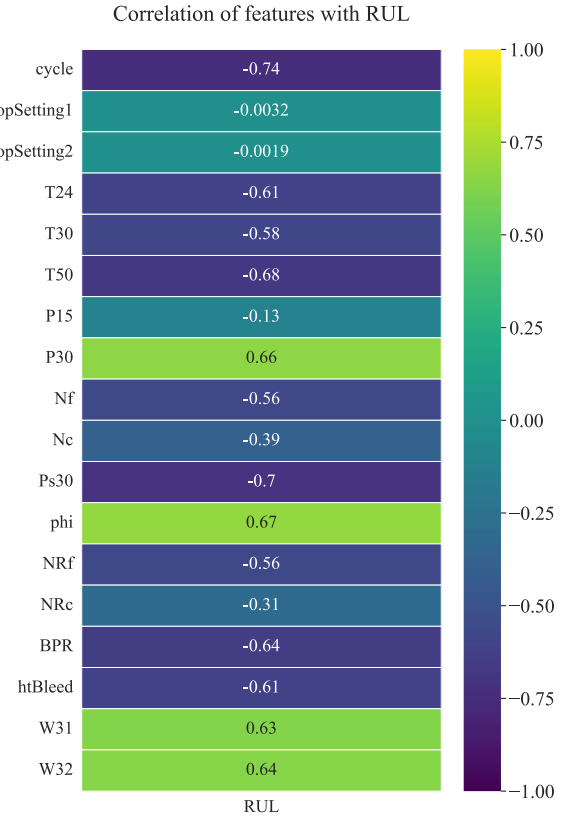


Figure – 2. Filtered correlation matrix.

5.4 Linear degradation modeling

It is seen that the remaining useful life (RUL) decreases linearly with time [17]. But no data relates to degradation. The process that was followed was to group the data frame by the engine column. Then, the maximum cycle of that engine is taken as the end-of-life (EOL). Now, it takes an additional parameter called early RUL. Then, it compares the EOL to early RUL. If the total life is less than the early RUL, it keeps the original RUL values. Otherwise, the algorithm creates a sequence of descending RUL by subtracting the maximum cycle of an engine from successive engine cycles.

A further approach was taken to assign the RUL to each of the engines at every cycle while maintaining the correct order. In order to do that, the entire training dataset is first divided into batches of equal length. With the window length and shift size, it iterates through the whole of the dataset. For each window, it assigns the RUL sequence received before to the engine group.

5.5 BiLSTM model development

The proposed BiLSTM model utilizes the normalized data before training. The model consists of three bidirectional LSTM models along with two dense layers. The benefit of using bidirectional LSTM is that it can process input data sequences in both forward and backward directions. The activation function used here is the hyperbolic tangent function, which is most widely used in LSTM modeling.

Right after the BiLSTM layers, there are two fully connected dense layers of 96 and 128 units, both using the ReLu activation function. Finally, one output dense layer is there to output the final regression result.

5.6 Evaluation matrix

After building the model, the model was trained with the dataset to predict the RUL of each engine. We have used two widely used evaluation methods here, RMSE and scoring function, which is the most widely used and reliable of all evaluation matrices. Mathematically, it is written as,

$$RSME = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}} \quad (4)$$

Here, x_i is the actual observation, $\bar{x}_i$ is the estimated value, and N is the number of samples.

In addition to the RSME, the PHM08 data challenge incorporated another evaluation matrix, namely the score function [18]. The score function is as follows:

$$score = \sum_{j=1}^N s_j \quad (5)$$

$$s_j = \begin{cases} e^{\frac{-h}{13}} - 1, & h_j < 0 \\ e^{\frac{-h}{10}} - 1, & h_j \geq 0 \end{cases} \quad (6)$$

The better the model performs, the lower the RSME value and the closer the score to zero.

6.0 Results and Discussion

Detailed performance of the proposed model is evaluated. It is found that a number of factors influence the performance and accuracy of a model. The first and foremost are the window size, threshold values and initial RUL. And then the hyper-parameters. These two factors significantly affect the performance and reliability of the model. In this study, we have utilized a somewhat straightforward approach rather than hypertuning the model. This has a significant impact on the performance of the model. Hypertuning gives the optimum parameters of a model. In the model, we have hardcoded the parameters ourselves as shown in Table 3.

Table – 3. Model parameters.

Leaning rate	Batch size	Max epoch	No. of layers	No. of neurons in each layer	Optimizer
0.001	50	30	3	128,64,32	Adam

The model is then trained with the pre-processed data as input, with the window length as 30 and an early RUL of 125. The performance scores of the model were quite satisfactory. The performance factors of the model are shown in Table - 2.

Table – 2. Model performance matrix.

Model	RSME	Score	R ²
BiLSTM	13.5468187	0.927647512	0.893728

The comparison between actual RUL and predicted RUL is illustrated in Figure 3, which shows acceptable RUL predictions.

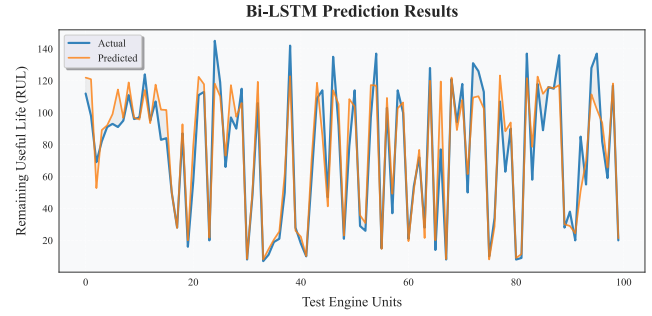


Figure – 3. Prediction results.

In the line plot, the x-axis represents the number of engines, while the y-axis represents the actual and predicted RUL. A significant gap between the actual RUL and the predicted RUL can be observed here. This is due to several factors. First of all, the raw data contains a lot of noise, which contributes to the loss of performance by increasing the gap between the actual and predicted RUL. Secondly, the model used in this investigation is rather straightforward without hyper-parameter tuning. Even without doing these two, the model is performing quite well due to the fact that the gap between predicted and actual RUL is not considerably high. So, upon fixing these two factors, the model is believed to have a deviation between actual and predicted RUL close to zero.

6.0 Conclusions

Predictive maintenance is now a trending field of research due to its incredible reliability. This was a work on the predictive maintenance and estimation of RUL of a turbofan jet engine. There are numerous models that can be utilized to predict. Among them, LSTM has a very solid stand to be able to work with sequential data. With good data pre-processing, each of the models can show very high accuracy, but again, LSTM can handle sequential data better. Feature selection, normalization, regularization, filtering, etc., play a very crucial role in the performance of a model. In our model, we have incorporated three layers of BiLSTM layers with two fully connected layers. In the future, hyperparameter tuning can be done in the model to find out the optimal number of layers, number of neurons, batch size, learning rate, and many more. Also, more efficient models like transformers can be implemented here for better accuracy and reliability.

NOMENCLATURE

<i>LSTM</i>	: Long-Short Term Memory
<i>CNN</i>	: Convolutional Neural Network
<i>RUL</i>	: Remaining Useful Life
<i>EOL</i>	: End of Life
<i>BiLSTM</i>	: Bidirectional LSTM
<i>GRU</i> :	Gated recurrent unit
<i>SCADA</i> :	Supervisory Control and Data Acquisition
<i>ANN</i> :	Artificial Neural Network
<i>XGBoost</i> :	Extreme Gradient Boosting
<i>LightGBM</i> :	Light Gradient-Boosting Machine
<i>DNN</i> :	Deep Neural Network
<i>PSO</i> :	Particle Swarm Optimization
<i>C-MAPSS</i> :	Commercial Modular Aero-Propulsion System Simulation
<i>SVR</i> :	Support Vector Regression
<i>PHM</i> :	Prognostics and Health Management
<i>RMSE</i> :	Root Mean Square Error

References

- [1] O. Merkt, "On the Use of Predictive Models for Improving the Quality of Industrial Maintenance: An Analytical Literature Review of Maintenance Strategies," presented at the Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, 2019.
- [2] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0," *Sustainability*, vol. 12, no. 19, 2020.
- [3] N. Bolander, H. Qiu, N. Eklund, E. Hindle, and T. Rosenfeld, "Physics-based remaining useful life prediction for aircraft engine bearing prognosis," in *Annual Conference of the PHM Society*, 2009, vol. 1, no. 1.
- [4] Y. Liao, L. Zhang, and C. Liu, "Uncertainty Prediction of Remaining Useful Life Using Long Short-Term Memory Network Based on Bootstrap Method," in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 11-13 June 2018 2018, pp. 1-8.
- [5] 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*, 6-9 Oct. 2008 2008, pp. 1-9.
- [6] J. Carroll, S. Koukoura, A. McDonald, A. Charalambous, S. Weiss, and S. McArthur, "Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques," *Wind Energy*, vol. 22, no. 3, pp. 360-375, 2018.
- [7] S. Ji, B. Lee, and M. Y. Yi, "Building life-span prediction for life cycle assessment and life cycle cost using machine learning: A big data approach," *Building and Environment*, vol. 205, 2021.
- [8] P. K. Ambadekar and C. M. Choudhari, "CNN based tool monitoring system to predict life of cutting tool," *SN Applied Sciences*, vol. 2, no. 5, 2020.
- [9] S. Wang, Z. Yu, G. Xu, and F. Zhao, "Research on Tool Remaining Life Prediction Method Based on CNN-LSTM-PSO," *IEEE Access*, vol. 11, pp. 80448-80464, 2023.
- [10] 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.
- [11] R. Karnik and K. Pandya, "Long-Short-Term Memory Model for Remaining Usable Life Estimation."
- [12] C. Zheng *et al.*, "A Data-driven Approach for Remaining Useful Life Prediction of Aircraft Engines," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 4-7 Nov. 2018 2018, pp. 184-189.
- [13] K. Liu, N. Gebraeel, and J. Shi, "A Data-Level Fusion Model for Developing Composite Health Indices for Degradation Modeling and Prognostic Analysis," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, pp. 652-664, 07/01 2013.
- [14] M. Ohki, M. E. Zervakis, and A. N. Venetsanopoulos, "3-D Digital Filters," in *Control and Dynamic Systems*, vol. 69, C. T. Leondes Ed.: Academic Press, 1995, pp. 49-88.
- [15] H. M. Elattar, H. K. Elminir, and A. M. Riad, "Towards online data-driven prognostics system," *Complex & Intelligent Systems*, vol. 4, no. 4, pp. 271-282, 2018/12/01 2018.
- [16] C.-G. Huang, H.-Z. Huang, W. Peng, and T. Huang, "Improved trajectory similarity-based approach for turbofan engine prognostics," *Journal of Mechanical Science and Technology*, vol. 33, no. 10, pp. 4877-4890, 2019/10/01 2019, doi: 10.1007/s12206-019-0928-3.
- [17] G. Lan, Q. Li, and N. Cheng, "Remaining Useful Life Estimation of Turbofan Engine Using LSTM Neural Networks," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, 10-12 Aug. 2018 2018, pp. 1-5.
- [18] E. Ramasso and A. Saxena, "Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset," presented at the Annual Conference of the Prognostics and Health Management Society 2014., Fort Worth, TX, USA., United States, 2014-09-29, 2014. [Online]. Available: <https://hal.science/hal-01145003>.