**Remaining Useful Life Estimation for Predictive Maintenance Applications**

**Objective:**

Predictive maintenance, aimed at predicting when equipment will fail and scheduling timely maintenance to prevent unexpected breakdowns, has emerged as a critical field in various industries. The estimation of Remaining Useful Life (RUL) of machinery and equipment is at the forefront of this domain, leveraging advanced data analytics and machine learning techniques to foresee equipment failure and optimize maintenance schedules. Remaining Useful Life (RUL) of a component or a system is defined as the length from the current time to the end of the useful life. Accurate RUL estimation plays a critical role in Prognostics and Health Management (PHM). Data driven approaches for RUL estimation use sensor data and operational data to estimate RUL.

**Introduction:**

Industrial systems, ranging from small machines to Jet engines, rely on maintenance for their durability. In recent years, companies have started to invest in techniques that can prevent faults in advance. One such technique is Preventive Maintenance [1]. Remaining Useful Life (RUL) of a component or a system is defined as the length from the current time to the end of the useful life [2]. The criteria to define whether the component or system is still usable is already known to the domain experts of the component or system. In industry operational research, there is an interest in modeling methods for RUL estimation given component or system condition and health monitoring information. Prognostic technologies are very crucial in condition based maintenance for diverse application areas, such as manufacturing, aerospace, automotive, heavy industry, power generation, and transportation. While accessing the degradation from expected operating conditions, prognostic technologies estimate the future performance of a subsystem or a component to make RUL estimation. If we can accurately predict when an engine will fail, then we can make informed maintenance decision in advance to avoid disasters, reduce the maintenance cost, as well as streamline operational activities. This works proposes a data driven approach to predict RUL of a complex system when the run-to-failure data is available.

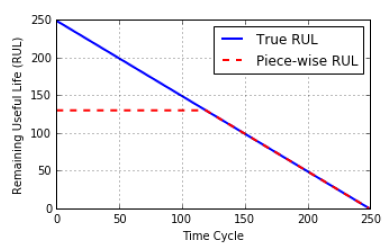
In this work, we propose a comprehensive methodology to analyze the open-source dataset of NASA's Turbofan Jet Engine for predicting Remaining Useful Life (RUL). The project encompasses a series of steps including Exploratory Data Analysis (EDA), Feature Engineering (FE), and the application of multiple machine learning algorithms to develop a robust predictive model. The objective is to provide an in-depth understanding and a high-performing model that can accurately forecast the RUL of jet engines, thereby facilitating predictive maintenance.

**Data Preparation:**

Data set consists of multiple multivariate time series. The data set is further divided into training and test subsets. Each time series is from a different engine i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition. The data is contaminated with sensor noise. The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

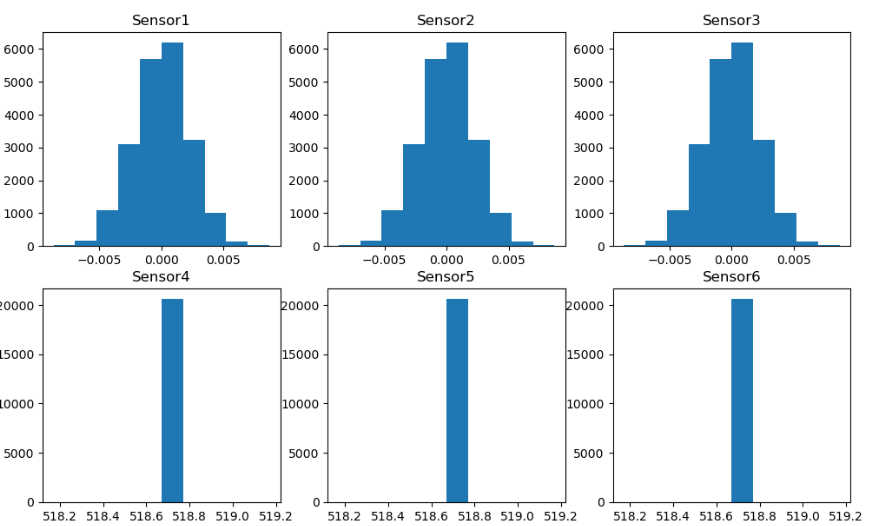
***CMAPSS Dataset:*** The data contains 26 columns of numbers, where each row is a snapshot of data taken during a single operational cycle. The columns corresponds to Engine ID, Time in Cycles, Setting1, Setting2, Setting3 and remaining all are sensor data.

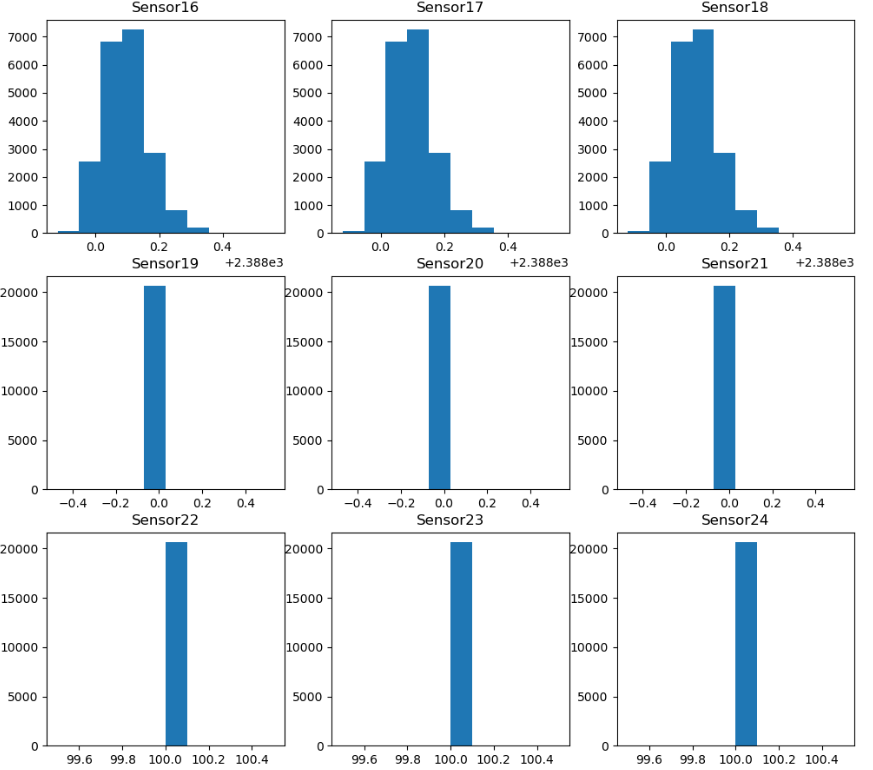
***Target RUL:*** The target variable is not explicitly provided in the dataset; therefore, it is inherently calculated by subtracting the number of cycles from the maximum cycle for the corresponding engine. This results in the creation of a new column labelled "Remaining Cycles," which represents the Remaining Useful Life (RUL) of the engine. The health of a system degrades linearly along with time. In practical applications, degradation of a component is negligible at the beginning of use, and increases when component approaches end-of-Life. To better model the Remaining Useful Life changes along with time, in [3] [4], a piece-wise linear RUL target function was proposed, as in Figure below, which limits the maximum RUL to a constant value and then start linear degradation after a certain degree of usage. We set the maximum limit as 130 time cycles for all the engines.

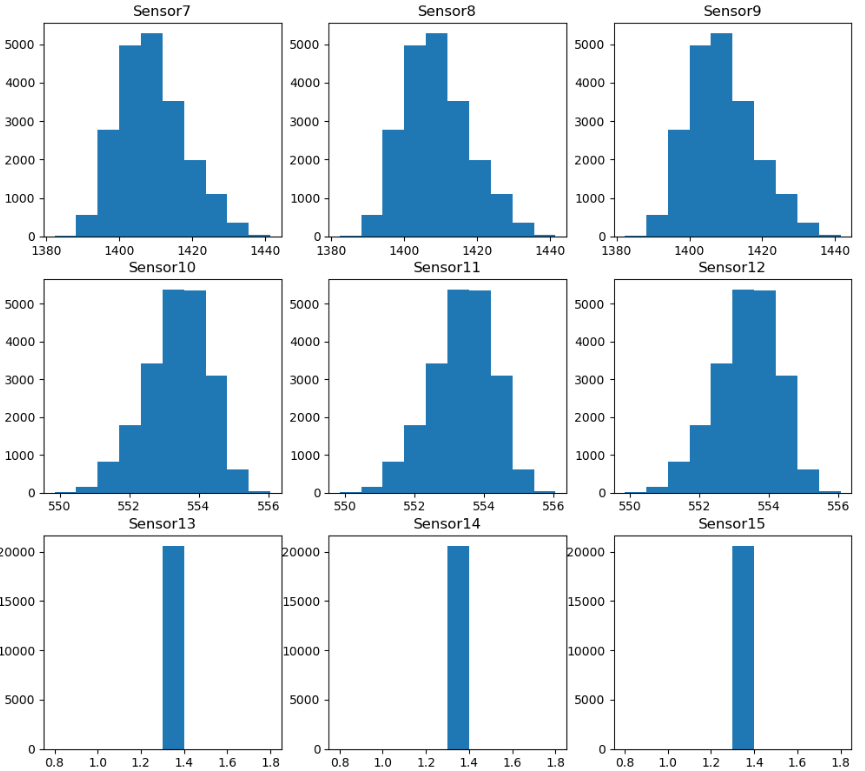


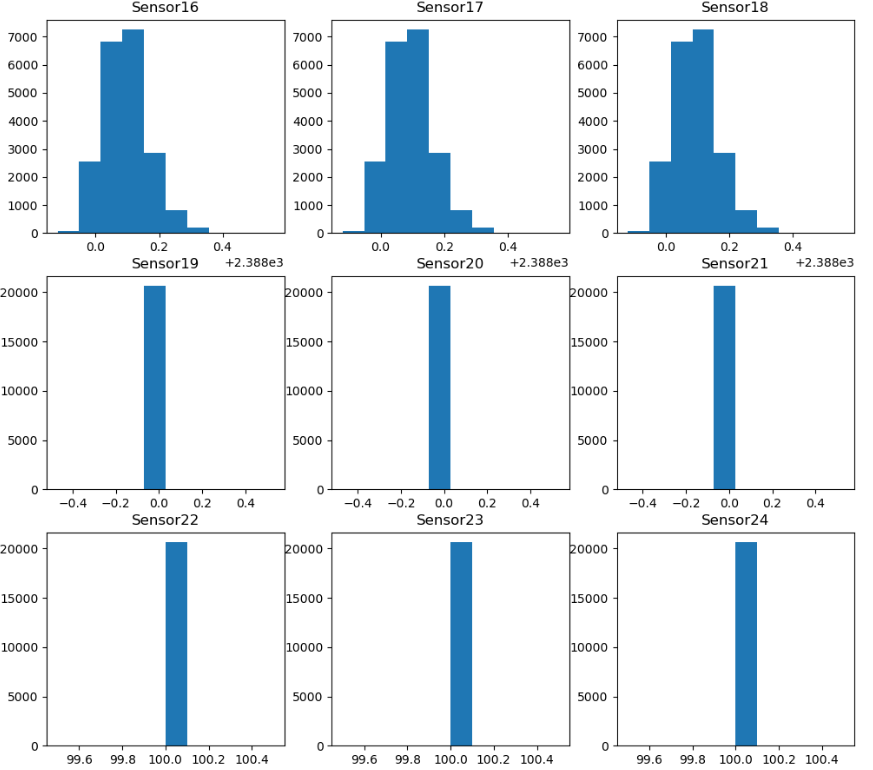
*Fig. Piece-wise RUL of the Data Set (Piece-wise maximum RUL is 130 time cycles)*

**Exploratory Data Analysis:**

The data distribution of each variable in the dataset varies significantly. The setting variables and sensor variables include constant, discrete, and continuous data types. As per figure, some sensors, such as sensor4, sensor5, and sensor6, exhibit constant values. In contrast, sensors such as Sensor1, Sensor2, and Sensor3 show a normal distribution. The remaining sensors display skewed distributions, either to the right or left. This diverse range of data distributions necessitates tailored pre-processing techniques to ensure optimal model performance.





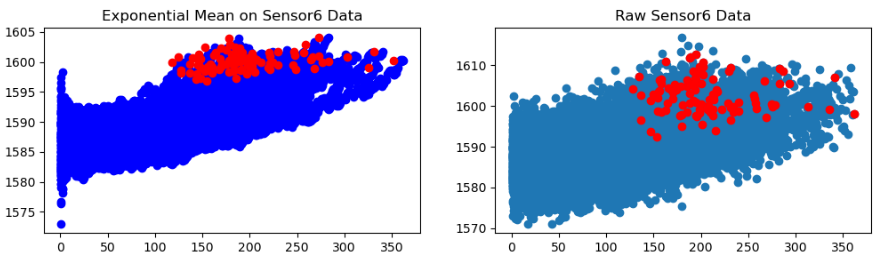


*Fig. Frequency Distribution Plot for Sensor Data*

**Data Pre-processing:**

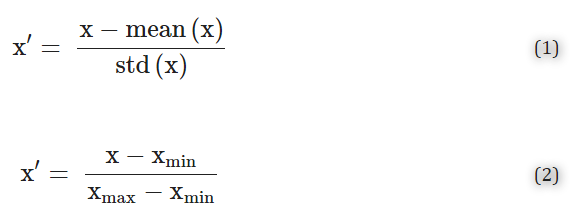
Some features are constant in the dataset, and thus, their variance is zero. All zero variance variables are removed before the training stage because they do not contain useful information for machine learning.

***Data Filtering:*** The sensor data in the dataset are noisy and sporadic, necessitating the application of smoothing filters to improve data quality. Two widely used smoothing techniques, Simple Moving Average (SMA) and Exponential Moving Average (EMA), were applied with various weights to determine the most effective method. Upon evaluation, EMA with an alpha value of 0.1 visually outperformed other configurations. Consequently, EMA with this alpha value was selected and applied to the sensor data, resulting in a smoother and more reliable data.



*Fig. Comparison between Raw sensor data and exponential moving averaged sensor data*

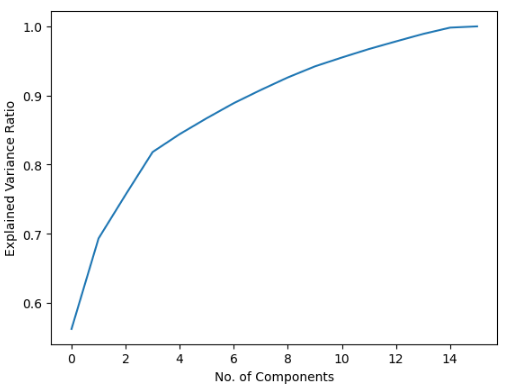
***Data Scaling:*** Since the value range is substantially different in different variables, it can be difficult to find the optimal point for the cost function. Therefore, the training and testing datasets need to be normalized. There are two widely used methods for normalization, which are Z-scores (Equation (1)) and min-max-scale (Equation (2)). Both methods are applied, and the one with the best evaluation result is selected.



**Principal Component Analysis:**

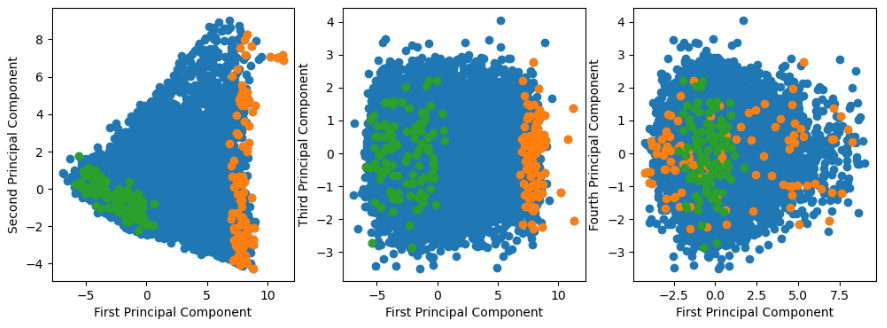
As part of feature engineering, Principal Component Analysis (PCA) is applied to visualize and understand the high-dimensional sensor data. PCA is a widely used dimensionality reduction technique that transforms the data by projecting it onto a set of orthogonal axes. This method works by identifying the eigenvectors and eigenvalues of the covariance matrix of the dataset. The eigenvectors, known as the "Principal Components," represent the directions of maximum variance in the data. By projecting the data onto these principal components, PCA reduces the dimensionality while retaining the most significant features, facilitating better visualization and analysis of the complex sensor data.

***Explained Variance Ratio:*** The number of components needed can be determined by looking at the cumulative explained variance ratio as a function of the number of components as shown in the below graph.



*Fig. Explained Variance Ratio of Principal Components*

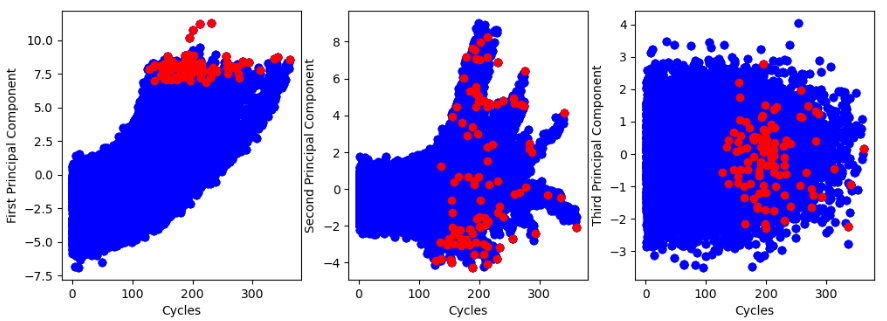
This curve quantifies how much of the total, the high dimensional variance is contained within the first N components. For example, we see that the first two components contain approximately 70% of the variance, while we need 12 components to describe close to 100% of the variance. Which means we can reduce our data dimension to 24 from 12 without much loss of the data.



*Fig. First 3 principal components are plotted*

The first, second, and third principal components are studied extensively as they represent 75% of the dataset's variance. The above figure represents the scatter plot for the complete dataset, where green dots represent the starting points of the engine cycles, and orange dots represent the failure points. It becomes evident from the scatter plot that thresholding the first principal component can effectively determine the failure point of all engines, irrespective of the other principal components.

**Feature Selection:**



**From the heat map it is evident that lot of features are highly correlated and it leads to avoid multi collinearity problem.** Multicollinearity refers to the situation where two or more features in a dataset are highly correlated, which can negatively impact the performance of machine learning models. In the presence of multicollinearity, the model coefficients can become unstable, leading to high variance and unreliable predictions. Therefore, it is crucial to select features that contribute the most to the model's performance while minimizing redundancy. To tackle the issue of multicollinearity and ensure the selection of the most relevant features, we applied the Select K Best algorithm. This algorithm ranks all the features according to a specified statistical criterion and selects the top K features that are most significant for predicting the target variable.

**Model Selection:**

According to our literature survey, Remaining Useful Life (RUL) prediction performs particularly well using neural network-based deep learning algorithms. To identify a highly robust model in terms of performance, we compared the results of several machine learning and deep learning models in our work. These models include Linear Regression, Random Forest Regression, eXtreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM) networks, and a hybrid model combining Convolutional Neural Networks (CNN) with LSTM. This comprehensive comparison allows us to evaluate each model's effectiveness in predicting RUL and select the best-performing approach for our predictive maintenance application

***Linear Regression Model:***

***Random Forest Model:***

***XGBoost Model:***

***MultiLayer Perceptron Model:***

***LSTM Model:*** Long Short-Term Memory Network (LSTM) [5] is a type of RNN network for sequence learning tasks and has achieved great success on speech recognition and machine translation. LSTM does not have long-term time dependency problems by controlling information flow using input gate, forget gate and output gate. Remembering information for long periods of time is practically their default behaviour. Due to inherent sequential nature of sensor data, LSTMs are well-suited for RUL estimation using sensor data. In this work, we propose a LSTM based approach for RUL estimation, which uses multiple layers of LSTM cells in combination with standard feed forward layers to discover hidden patterns from sensor and operational data with multiple operating conditions, fault and degradation models.

LSTM cell structure at time t is shown in Figure 2 [5]. We differentiate output of LSTM cell and cell state denoted as h (t) and c (t), respectively. Vector size of the output and cell state is the same and it is defined by number of nodes in the cell. This LSTM cell also takes sensor data xt as an input. There are three gates that control the information flow within cell:

(1) Input gate it controls what information based on output h (t-1) and sensor measurements x (t) will be passed to memory cell,

(2) Output gate controls what information will be carried to the next time step, and

(3) Forget gate controls how memory cell will be updated as shown in Figure 2. In this work, all LSTM cells that are used in the models are implemented as follows:

There are many variants of LSTMs and experimentation on this is part of our future work. For example, the nonlinear sigmoid and hyperbolic tangent activation function can be replaced using other activation functions. One can also choose to use the cell state c (t-1) as an extra input into the three gates. Connection between different layers of LSTMs in Figure 2 is achieved such that the output of one layer is as an input to the next layer. Sensor measurements are input only to the first LSTM layer.

****

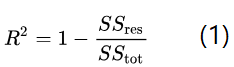
*Fig. LSTM Cell*

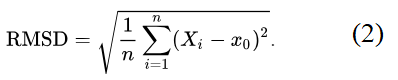
*LSTM works only on time series data, hence the dataset is split into fixed length time series data of length 30.*

**CNN + LSTM**

**Model Evaluation:**

In order to evaluate the performance of a RUL estimation model on the test data, Root Mean Square Error (RMSE) (Equation 2), gives equal penalty weights to the model when the estimated RUL is smaller than true RUL and when the estimated RUL is larger than true RUL and R2 Score (Equation 1), which is also widely used as an evaluation metric for the estimation of RUL.





All the models are evaluated based on RMSE and R2 score on the test dataset and the results are shown in the table below.

|  |  |  |
| --- | --- | --- |
| Model | Test RMSE | Test R2 Score |
| Linear Regression |  |  |
| Random Forest |  |  |
| Multilayer Perceptron |  |  |
| LSTM |  |  |
| CNN + LSTM |  |  |
|  |  |  |

**Building an application:**

**References:**

[1] S. Duffuaa, M. Ben-Daya, K. Al-Sultan, and A. Andijani, “A generic onceptual simulation model for maintenance systems,” Journal of Quality in Maintenance Engineering, vol. 7, pp. 207–219, 09 2001.

[2] 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

[3] Heimes, F.O.: Recurrent neural networks for remaining useful life estimation. In: International Conference on Prognostics and Health Management, 2008. PHM 2008.

[4] 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

[5] Shuai Zheng, Kosta Ristovski, Ahmed Farahat and Chetan Gupta, “Long Short-Term Memory Network for Remaining Useful Life Estimation,” in IEEE International Conference on Prognostics and Health Management (ICPHM), 2017