Predicting the Remaining Useful Life (RUL) of aircraft engine

Vijay Kothari, M.Sc.
Aerospace Vehicle Design
Cranfield University
Cranfield, UK
Email: vijayaeronautics@gmail.com

Debabrata Adhikari, PhD
Bernal Institute
University of Limerick
Limerick, Ireland
Email: deb1729@gmail.com
Orcid: 0000-0002-2254-7846

Keshava Kumar S, PhD
Blades Technology
Siemens Gamesa Renewable Energy
Bengaluru, India
Email: keshav@iitbombay.org
Orcid: 0000-0003-4006-5637

***Abstract*—The RUL models for turbofan engines are evaluated, and the engine data is shared by NASA. The data set contains the time history of 100 engines with two failure modes. The different ML models, such as Linear Regression, K-Nearest Neighbour Regression, Polynomial Regression, SVM Regression, Random Forest Regression, and LOSSO Regression, are studied for RUL estimation. Among all the models, Random Forest gives the best result based on RMSE and R2 scores both in train and test data sets.**

***Index Terms*—RUL, Machine Learning, Random Forest, Support Vector Machine, LOSSO, Linear Regression, Polynomial Regression, KNN, and Ensemble Learning.**

I. Introduction

The predictive health monitoring of aircraft turbofan jet engines is of utmost importance. If serviced well in time, the aircraft engines will result in economical flight operations and reduce fatal accidents. The data-based remaining useful life estimation for aircraft engines is advantageous compared to physics-based methods, as the physics-based model is challenging to develop. Also, the computation cost required for simulated physics RUL estimation is high.

A turbofan engine has a fan, low-pressure and high-pressure compressor, combustion chamber, turbine, and exhaust, as shown in Figure 1. The fan and compressor chamber draw the atmospheric air and compress it, and the compressed air is mixed with fuel, and combustion occurs in the combustion chamber. The combusted gas in turn, drives the turbine, and the part of the power generated by the turbine is used to drive the compressor stages.

The data provided by NASA is synthetically generated by a system model called C-MAPSS [1]. The closed-loop model has been used to simulate the data for different operation altitudes. The model takes 14 input parameters, among the 14 input parameters 1 is fuel flow and the reaming 13 are health-parameter inputs that allow the user to simulate the effect of faults and deterioration of five engine rotating components (Fan, High-Pressure Compressor (HPC), Lower Pressure Compressor (LPC), High-Pressure Turbine (HPT) and Lower Pressure Turbine (LPT)). The output includes the 21 variables measured via sensors, and the names of measured outputs which are the input data for our RUL model, are mentioned in Table 2 of reference [1]. The approach to finding the RUL of the turbine is approached by first analysing the training data provided by NASA, and the feature analysis of data is performed to eliminate unnecessary sensor data columns. The sensor data is normalised so that each sensor gets an appropriate weight. The training data is split into training and test data for model calibration; finally, the trained model is validated with validation test data provided by NASA [2].

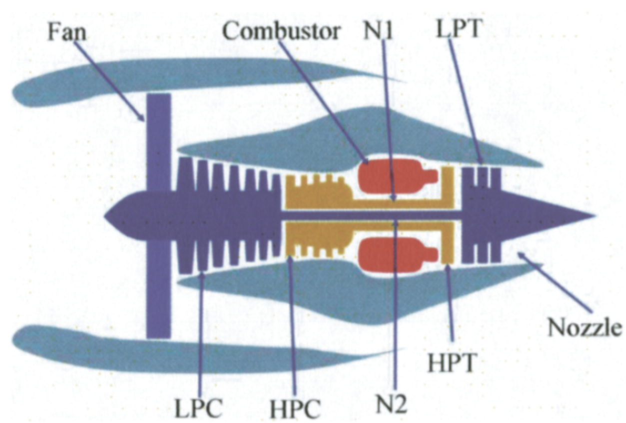


Fig. 1. Turbo fan C-MAPSS simulated jet engine components from NASA [1].

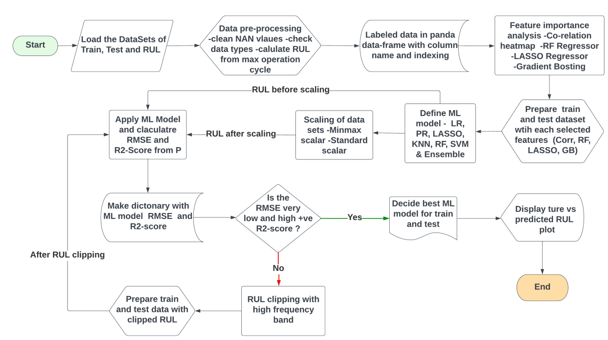


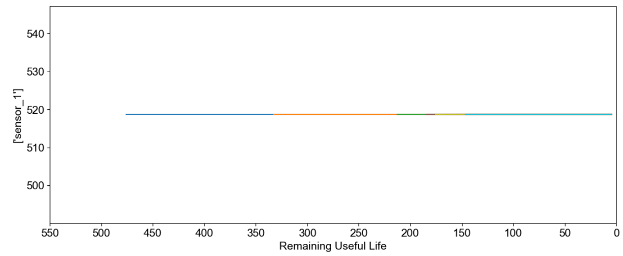
Fig. 2. Flow-chart of the project workflow

II. Problem description and methodology

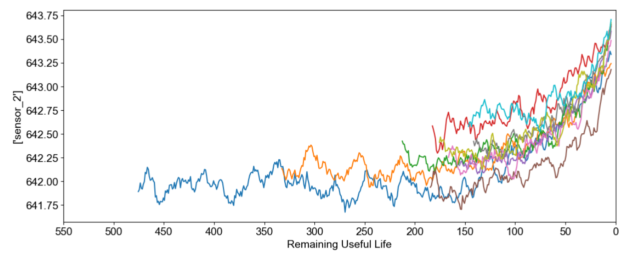
The data for aircraft gas turbine engines is shared by NASA [2]. The data shared by NASA is multivariate time series data, and there are multiple data sets; we have chosen data set 3 for project implementation. The data set contains 26 columns, the first column corresponds to the unit number, each unit a particular gas turbine. The second column refers to the data time in cycles (operation cycle of the engine), the columns 3 to 5 log the operational settings data. The remaining columns from 6 to 26 capture the sensor data, in effect, there are 21 sensors which capture the data of the turbine in operation. The number of rows for a given unit number corresponds to the total number of cycles the gas turbine was operational, and the last row for the given unit number is the end of life for that particular turbine. The data set 3 provided by NASA is for 100 gas turbines operating at sea level, the two fault modes of the engine failure are High-Pressure Compressor (HPC) Degradation and Fan Degradation. Along with the 100 turbine time series data for the training machine learning models, 100 turbine time series data for test validation are also provided. The other data sets provided are for different operational conditions and different fault mode combinations. The degradation is the measure of the efficiency of the particular turbine parameter. The 21 sensors gather the input data for the machine-learning model. Table 2 of the reference article by Saxena [1] mentions the parameter's names. Examples of plots for sensors 1, 2,8 and 15 signals for every 10th turbofan engine are shown in Figure 3. Time series signal plots for sensors 1 remain constant and Sensor 2, 8 and 15 shown nonlinear behavior with RUL. However, the workflow to evaluate the RUL is provided in the flow-chart Figure 2.

III. Data preparation

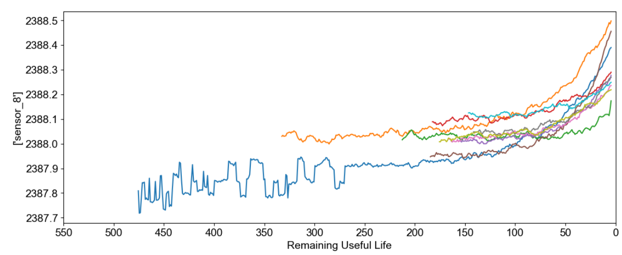
The data is imported through the pandas csv_read method, and by printing the data, it is observed that column numbers 26 and 27 are NaN (not a number), hence they have been dropped from further processing. After cleaning up the data, the statistical data parameters are evaluated and printed using the described method, and data is visualised through multiple plots, and the one plot which gives the correlation between different columns is obtained through a heatmap correlation shown in figure4. The white space for the sensors does not have any correlation with RUL. The correlation values for different sensors are plotted along with the heat map. It is noted that if a feature has zero variance (i.e., constant), its correlation with other features and the target variable will be NaN since the feature's variance is zero, and division by zero is undefined. Therefore, if a feature has zero variance, its correlation value will be NaN. The sign of the correlation coefficient indicates the direction of the relationship between two variables. A positive correlation coefficient indicates that as one variable increases, the other variable also increases. On the other hand, a negative correlation coefficient indicates that as one variable increases, the other variable decreases. So, the sign of the correlation coefficient is essential because it helps us understand the nature of the relationship between two variables. However, if we are only interested in the strength of the relationship and not the direction, we can take the absolute value of the correlation coefficient. Based on the correlation values with RUL (see Supplementary document), the essential features of the ML model are shown above. These features have a relatively higher correlation with RUL than other data set features. However, it's important to note that correlation does not necessarily imply causation, and other features might be necessary for the ML model. Therefore, we will perform additional feature selection techniques like the feature importance scoring algorithms to identify the most relevant features for the ML model.



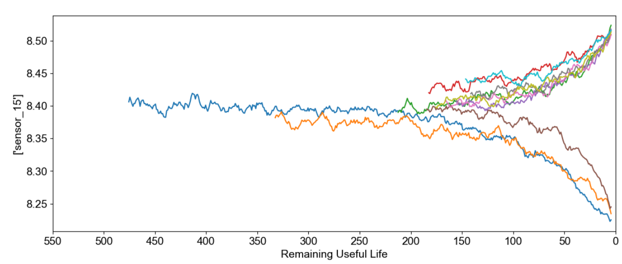
(a)



(b)



(c)



(d)

Fig. 3. Examples of plots for sensors (a) 1, (b) 2, (c) 8 and (d) 15 signals for every 10^{th} turbofan engine

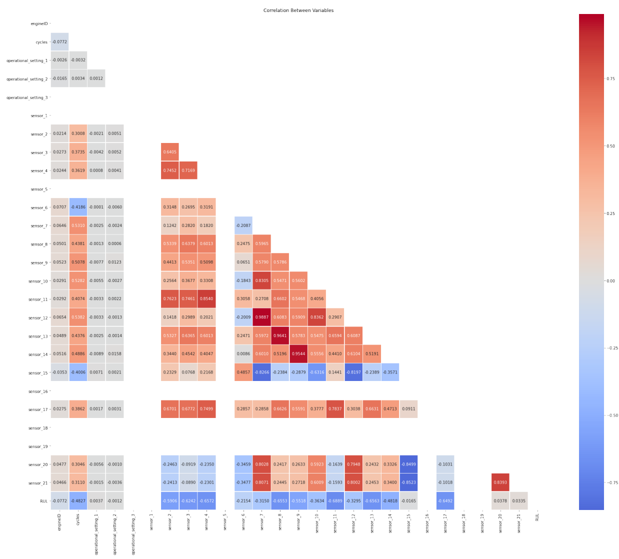


Fig. 4. Co-relation between different sensors

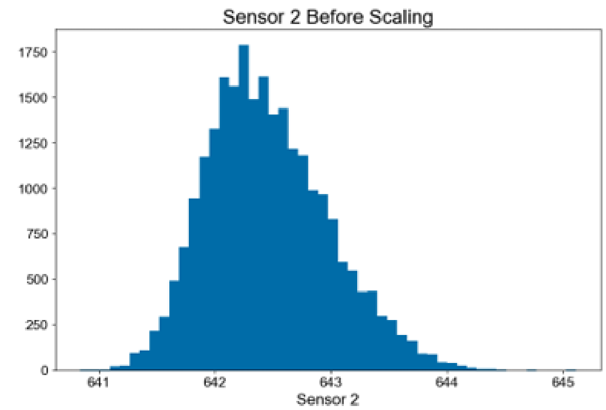
A. Feature Importance Analysis

Feature importance values indicate how much each feature contributes to the model's ability to make accurate predictions. The importance values are relative to the other features in the data set. They are calculated based on how much the model's performance decreases when a feature is randomly shuffled or removed from the data set.
So, if a feature has a high importance value, the model relies heavily on that feature to make accurate predictions. If a feature has a low importance value, it means that the model does not rely on that feature very much, and removing it from the data set may not affect the model's performance very much.
However, it is essential to note that feature importance values are not absolute measures of feature usefulness. They can be influenced by the other features in the data set and the specific model and hyper-parameters used. It is worth to mention when we drop a feature from the analysis, the feature importance values for the remaining features may change. Here four different types of essential features have been performed as follows:

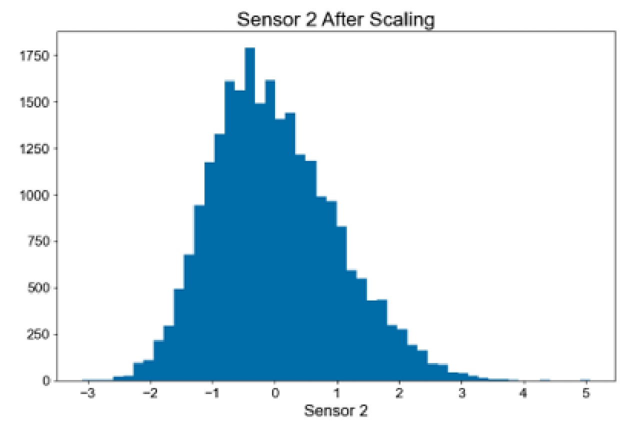
- Correlation (CORR) with RUL
- Random Forest (RF) feature importance
- Lasso feature importance
- Gradient Boosting (GB) feature importance

Later, the resulting features were merged into a single data frame and displayed the selected features from each approach. It is noted that the number of selected features from CORR with 0.1 thresholds, RF, Lasso and GB are 13, 12, 18, and 17, respectively. This implies that RF has only 12 essential features to contribute more to prediction or RUL. Then the train and test data sets have to be re-framed into a new data set by the selected features from each method.

The features are normalised to give equal weight to all the



(a)



(b)

Fig. 5. (a) Data set before and (b) after standard probability scaling of train data set

sensor data, which have been selected for training the machine learning algorithm. The plot of features histogram before and after normalisation with standard probability scaler is shown in Figure 5. Also, the MinMaxScaler has been performed to scale the data sets into [0, 1] close intervals. However, it is observed that since the data set itself follows a normalised probability function, it is customary to use the StandardScaler scaling method of the data sets.

IV. MACHINE LEARNING MODEL

The scikit-learn package is used to deploy the machine learning algorithms. The following methods are employed:

- Linear Regression (LR): Linear regression determines the linear relation that best fits a set of discrete data points. At the start of the project, the fundamental Machine Learning method of linear regression is used here because of its simplicity and essential features relation.
- Polynomial Regression (PR): Polynomial regression is a type of linear regression wherein the relationship between the dependent and independent variables is predicted by the nth degree of the polynomial.
- K-Nearest Neighbour Regressor: K-NN is a non-parametric supervised learning classifier method that uses

proximity to make classifications or predictions about the grouping of an individual data point by computing the local probability. However, KNN regression is used to predict the output variable using the local average. Here, K -means the number of the classifier as we increase or decrease the prediction gets better. Herein, we have tuned the hyper-parameters K to find the best optimal prediction based on the rank of the classifier.

- LASSO Regression: Least Absolute Shrinkage and Selection Operator (LASSO) is used for regularization and feature selection. This model uses the L1 regularization technique, where the penalty term α determines the amount of shrinkage that can occur in the model equation. In the present work, we have also parameterized with α as hyper-parameters. Moreover, due to convergence with the model number of iterations is also parameterized. Finally, a grid_search_cv has been performed to obtain the best hyper-parameters for the present data sets.
- Random Forest (RF) Regression: Random Forest is an ensemble technique capable of performing regression and classification tasks using multiple decision trees. Initially, every decision tree will have its own high variance, but when a parallel combination of multiple decision tree ensemble together, the resultant variance of each tree gets low. It happens because the resultant variance is now perfectly trained on that particular sample data and no longer depends on a decision tree but on ensemble multiple decision trees. As a result of the regression problem, the final output is the mean of all the output from every tree. As hyper-parameter tuning, we have used max features selection as the square root of the number of features from selected feature sets. We have also used another parameter, random state 42, to sets a seed to the random generator for feature clustering.
- Support Vector Machine (SVM) Regression: Support vector machine regressor is a supervised learning algorithm that uses a similar support vector machine principle for discrete data set clustering. The main idea behind SVR is to obtain a bet fit line known as a hyperplane with the maximum number of points.
- Finally, Ensemble Learning: Ensemble learning is a machine learning approach where the voting ensemble makes model predictions of data points of different models. Ideally, a group of weak learners provides the prediction as one strong learner. This work uses an ensemble learning model combined with random forest and support vector regressor.

A. *Grid_search_CV*

In addition to the predefined ML model, in the next section, we perform *grid_search_CV* module from scikit-learn to determine the hyper-parameters tuning on a couple of ML models. Here we performed *grid_search_CV* for KNN regression and LASSO regressor as follows.

- In KNN regression, the model hyper-parameters *n_neighebors* have been chosen from 1-20 to observe the parameter ranking based on the minimum RMSE score. We found that the model performs well as we increase the *n_neighbors*. However, it is observed that the difference in RMSE score of rank 16 and 17 shows 0.75; therefore, for the time being, we restrict our model parameters *n_neighbours* to 5 in the KNN regression model. The detailed ranking of KNN run-time results is in the GitHub repository (Appendix).
- In LASSO regressor, we perform *grid_search_CV* on two hyper-parameters, α, as in $[0.1, 0.2, 0.3]$ and *max_iteration* $[10, 1000, 10000]$. We also rank the ML model hyper-parameters based on the RMSE score. It shows α 0.1, and *max_iteration* 1000 provides the rank 1 for the data set FD003 turbofan engine. The detailed ranking of LASSO regression is in the GitHub repository (Appendix).

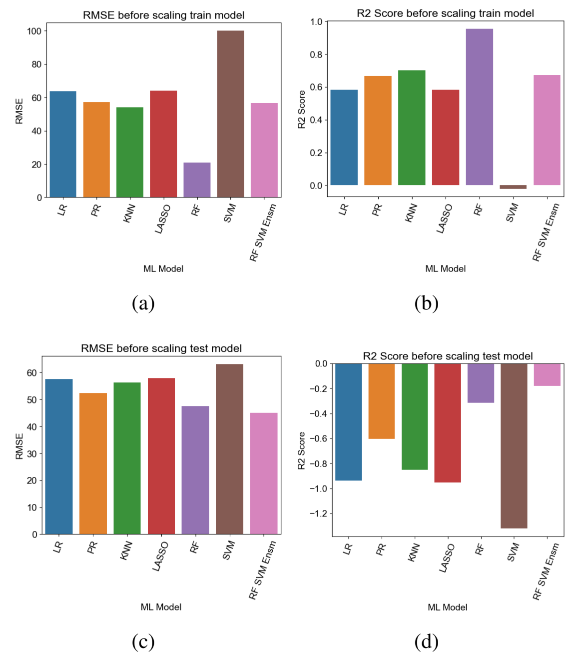


Fig. 6. Before scaling train data (a) RMSE (b) R2-score and for test data (c) RMSE, (d) R2-score.

It is noted in Figure 6 that before scaling the train data sets, the RMSE values of random forest regression (RF) are low, and R2-score is also high, which means the RF is well-trained compared to other ML models. Although the RMSE values on test data sets show low in RF and RV-SVM ensemble models, the R2- score in all the models is negative, implying that the model's prediction is worse than a constant function that always predicts the mean of the data. Scaling the data sets before training and testing the ML model is necessary.

After scaling the train data sets, similarly, in Figure 7, the RMSE values of random forest regression (RF) are low, and R2-score is high, meaning the RF is well-trained compared to other ML models. The RMSE values on the test data set for SVM are lower than all the other models. However, the R2- score in all the models still shows negative, which implies

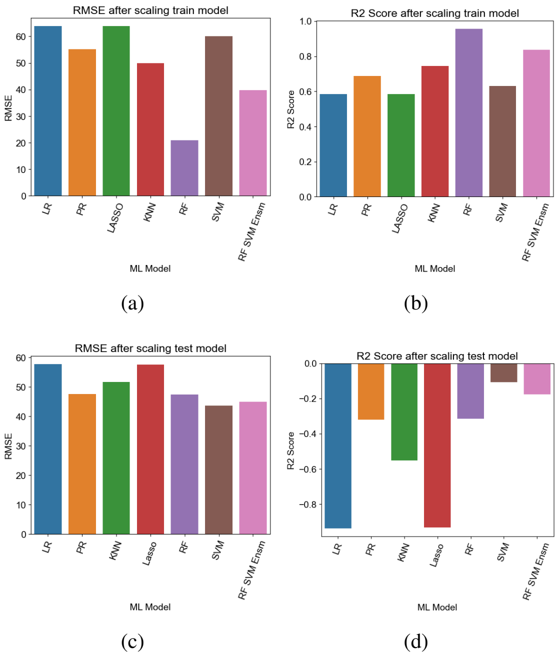


Fig. 7. After scaling train data (a) RMSE (b) R2-score and for test data (c) RMSE, (d) R2-score.

that the model's predictions are worse than a constant function prediction.

B. RUL Clipping

RUL clipping was performed to address overestimation in our algorithm's predictions by restricting the Remaining Useful Life (RUL) values based on a threshold. The RUL distribution in Figure 8 shows that the RUL between 0 to 145 has more occurrences than the other part of the data set. Also, RUL 145 is already a high value in the test data set. Therefore, the application clipping was performed wherein RUL values above 145 were set to 145 in both the train and test data sets. As shown in the red-box of in Figure 8, the clipping of data sets has been performed based on the maximum RUL threshold of 145. This helps to reduce overhead without affecting the main objective of the ML analysis. In the next section, training and testing of all the ML models will be performed with RUL-clipped data sets.

It is observed in Figure 9 that RMSE values for the RF model after RUL clipped are very low and high R2-score which means that the random forest model has trained better than other ML models. However, in the test sets after RUL clipping, the Polynomial regression (PR) model up to polynomial degree 4 performs better than the RF model. PR has a lower RMSE and high R2-score than all the other ML models. After RUL clipping showed a significant improvement in the algorithm's predictions, with a decreased RMSE and increase in R2 score, indicating that RUL clipping effectively improved model performance and ensured predictions do not exceed the remaining life of the engine, indicating more reliable predictions.

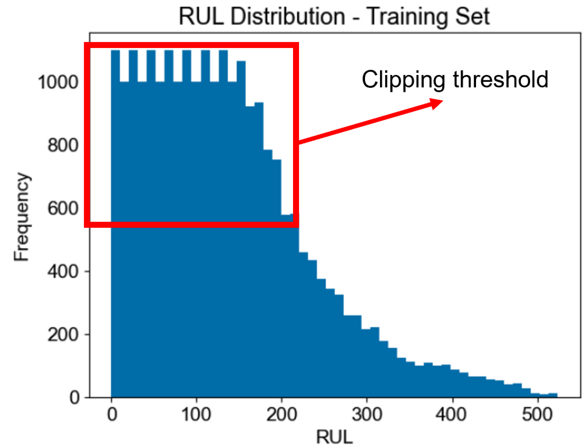


Fig. 8. RUL clipping of train data set.

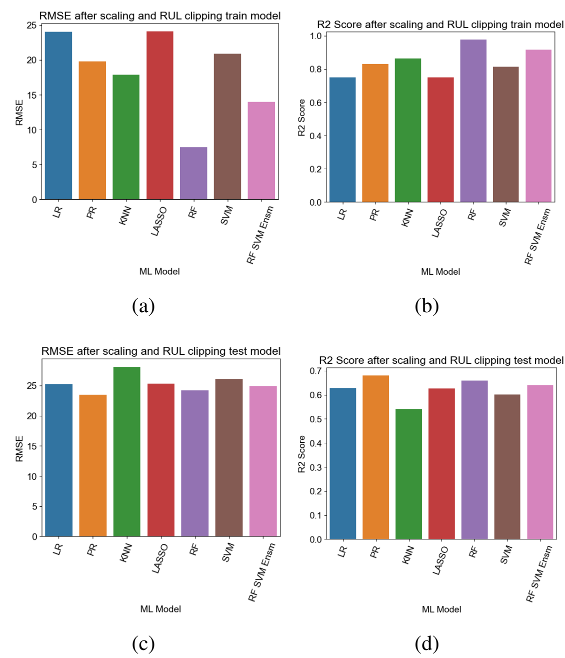


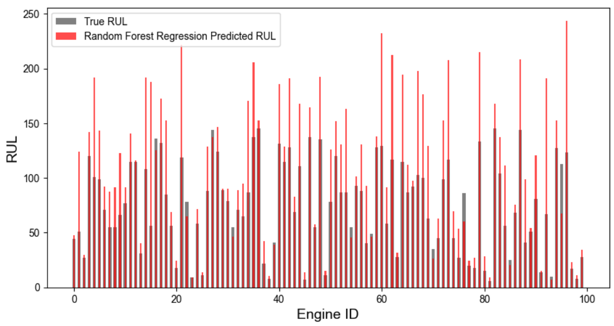
Fig. 9. After RUL clipping train data (a) RMSE (b) R2-score and for test data (c) RMSE, (d) R2-score.

In Figure 10, a comparison of the true vs model predicted RUL on test data after clipping shows a significant improvement in prediction.

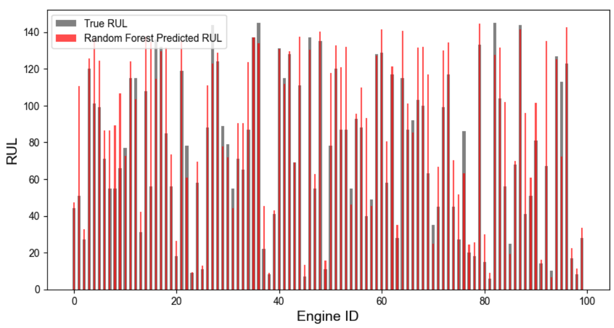
However, a study with all the selected features from CORR, RF, LASSO and GB has been performed to propose a better-predicted ML model. The details of RMSE and R2-score with all ML model is shown in GitHub file *ML_model_prediction_summary.png*. It indicates that the random forest (RF) model performs better in train test data sets than any other model in the present work.

V. Conclusion and Future Work

First and foremost, the data set has been loaded using the Python library. The data cleaning and pre-processing of



(a)



(b)

Fig. 10. True vs predicted RUl (a) before and (b) after RUL clipping on test data.

test data sets have been done to prepare the data frame and employed feature selection, scaling, and model training to analyze the data and build the model. Evaluated model performance error metrics with RMSE and R2 Score. Introduced four different feature selection techniques (CORR, RF, SVM, and Ensemble learning) to identify the most relevant features for the prediction. However, ML models were performed with different feature selections from CORR, RF, LASSO and GB. The RF regression feature importance analysis yielded the best performance. Additionally, hyper-parameters tuning has been performed using techniques like Grid_search_CV for improved model performance. Then the best-performing models were determined based on the train and test data sets' lowest RMSE and highest R2-score. However, the exercise of ML model and data munging, and feature engineering has practical implications for many real-world applications with multiple features.

To improve the model predictions, the future scope of improvement in the present work is anomaly detection, clustering analysis, hyper-parameter tuning, signal smoothing, feature engineering, define the novel loss function.

Acknowledgment

We would like to express our gratitude to Professor Shashikumar Ganesan and the teaching assistant for their guidance, feedback, and support throughout the course. Their valuable insights and suggestions have been instrumental in helping us develop skills and knowledge in machine learning. Thank you for sharing your expertise and for being available to answer our questions. Also, thank you Centre for Continuing Education, Indian Institute of Science, Bangalore, INDIA for this opportunity.

Appendix

Link to the GitHub repository is in https://github.com/VijayAerospace/ML_IISC_CCE_RUL_FD003_V1_2023

References

[1] 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.

[2] "Cmapss jet engine simulated data," https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6.