Predictive Maintenance of Aircraft Engines: Machine Learning Approaches for Remaining Useful Life Estimation

Deepika J
School of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India.
*deepika.j@vit.ac.in

Pappuru Manikanta Reddy
School of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India.
manikantapappuru@gmail.com

Kothaluri Murari
School of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India.
murarikothaluri@gmail.com

Boyana Rahul
School of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India.
boyinarahulsagar@gmail.com

***Abstract*—In the realm of predictive maintenance for aircraft engines, ensuring precise and timely predictions of Remaining Useful Life (RUL) is paramount to enhancing safety, reducing costs, and minimizing downtime. While deep learning models dominate this field, their complexity, high computational demand, and lack of interpretability can hinder practical deployment in real-world scenarios. This study demonstrates the potential of machine learning models as an efficient and interpretable alternative for RUL prediction, leveraging the publicly available C-MAPSS dataset from NASA. Using C-MAPSS dataset, we integrated PostgreSQL for scalable data storage and Flask for interactive real-time visualization, creating a seamless pipeline from data ingestion to result interpretation. Preprocessing steps, including normalization, noise removal via box plots, and feature engineering, ensured a clean and reliable dataset. Among the models evaluated—XGBoost, Random Forest, SVM, KNN, and Linear Regression—XGBoost emerged as the best performer, achieving an RMSE of 23.8 and an R^2 score of 0.67. These results nearly matches the accuracy typically associated with deep learning models while maintaining computational efficiency. Feature importance analysis through Explainable AI (XAI) further revealed critical insights into sensor contributions, enhancing interpretability and trust in the predictions. This work underscores the practicality and effectiveness of machine learning in predictive maintenance, offering a robust, transparent, and resource-efficient approach to RUL prediction. The findings highlight its potential to meet industry needs where simplicity, accuracy, and interpretability are critical for decision-making and operational efficiency.**

Keywords—PostgreSQL, Flask Web Application, XGBoost, Machine Learning, Aircraft Engines, Predictive Maintenance, Remaining Useful Life (RUL) Prediction, and Feature Engineering.

I. INTRODUCTION

Aircraft engines are critical for flight, operating under extreme conditions like high temperatures, pressure, and mechanical stress, making them susceptible to degradation and failure. Such failures can lead to costly repairs, flight disruptions, and safety risks. Predictive maintenance offers a proactive approach by leveraging real-time sensor data and machine learning to forecast the Remaining Useful Life (RUL) of components. Unlike reactive maintenance, which addresses failures after they occur, or preventive maintenance, which follows fixed schedules, predictive maintenance detects potential issues early, ensuring timely interventions. This approach minimizes unscheduled maintenance, enhances operational safety, and reduces costs, making it essential for the aviation industry's stringent safety standards.

The goal of this study is to create a predictive model that forecasts an aircraft's remaining useful life (RUL) or cycles by analyzing various sensor data collected from onboard sensors. This model uses machine learning algorithms to produce performance metrics comparable to deep learning and Long Short-Term Memory (LSTM) methods, which are increasingly used for similar predicting tasks. For this study, we utilized NASA's C-MAPSS FD001 dataset for the purposes of training, validation, and testing. This dataset encompasses a total of 26 features that capture critical operational parameters of the aircraft engines. Initially, these datasets are stored in PostgreSQL before being retrieved for analysis within the codebase. The primary aim is to ascertain the maximum number of remaining cycles by thoroughly analysing the sensor values at any given time. Notably, the training dataset does not include a target column; hence, we computed this column through the application of specific equations and strategically defined threshold values to enhance model accuracy. After extensive experimentation with various threshold values, we adopted a threshold of 130 because of multiple evaluations.

We have used Machine learning models over Deep Learning models in this paper because if dataset size is limited, such as your NASA CMAPSS dataset, machine learning (ML) models outperform deep learning (DL) due to their ability to deliver high accuracy without requiring massive data volumes Deep learning models, on the other hand, rely heavily on large datasets to generalize well and avoid overfitting. This makes ML models like Random Forest or XGBoost more practical and efficient when working with moderately sized datasets. Another advantage of ML models is their computational efficiency. For real-time applications, such as monitoring engine health, ML models can be deployed easily on standard servers or edge devices without the need for specialized hardware like GPUs. Deep learning models, due to their complexity, demand more computational power and are often unsuitable for systems with limited resources. Interpretability is another area where ML models excel. When understanding the impact of each sensor on RUL predictions is critical, ML models provide insights through feature importance metrics. This level of transparency is not easily achievable with DL models, which often act as "black boxes," making it hard to justify predictions. Also, the preprocessing requirements also highlight the strength of ML models. While DL models can handle raw data, they are highly sensitive to noise and may require additional layers to address this. One drawback of data-driven techniques is their need on training data to identify correlations, create patterns, and assess data trends, which might result in failure. [4]. Since the multivariate sensor data quantifies a variety of dependent and independent situations as signals, it is a great source for predicting the RUL. [15]. RUL has been successfully predicted by combining signal processing methods with variety ML and deep learning algorithms. Several well-known techniques, such as support vector machines (SVM), regression models, Artificial Neural Networks (ANN), Decision and Classification trees, Principal Component Analysis (PCA), Naive Bayes, and Extreme Learning Machines (ELM), are often employed in the RUL prediction of mechanical components. [16].

In summary, the main objective of the work is:

- The primary objective is to develop robust machine learning models capable of accurately predicting the Remaining Useful Life (RUL) of aircraft engines with performance near to deep learning and LSTM methods.
- A critical focus is on thorough feature engineering and analysis to enhance model performance.
- The project aims to integrate the predictive model with a PostgreSQL database for effective data management and retrieval, along with a Flask-based web interface for dynamic visualization.

II. LITERATURE REVIEW

Remaining Useful Life (RUL) prediction has become a critical focus in industries relying on the longevity and performance of machinery, particularly in aviation and aerospace sectors. The ability to predict the remaining cycles of aircraft engines not only helps in scheduling maintenance but also ensures optimal performance and safety. In recent years, more studies have leveraged deep learning techniques over traditional machine learning ways to address the challenges in RUL prediction, with a focus on accurately forecasting engine failure based on sensor data, to improve operational efficiency and reduce unforeseen downtimes.

In contrast to neural networks, which had a score value of 1046, Khelif et al. [1] used SVM to model and achieved a competitive score value of 448. This is explained by the excellent classification accuracy achieved when modeling with a kernel function. Deep learning has already seen some success in the industrial sector because of its potent self-learning capabilities [2]. The determination of the exact output value for each input data point is a fundamental difficulty in forecasting the Remaining Useful Life (RUL) of a system. It is challenging to accurately evaluate the health condition of the system at every time step in real-world applications due to the lack of a precise physics-based model [2]. Previous research has often assumed that system deterioration begins only after a specific utilization threshold, prompting the development of piecewise linear degradation models tailored for C-MAPSS [3]. These models, while effective in capturing general degradation trends, often require careful calibration to achieve optimal performance. Zhao et al. [3] highlighted how deep learning may be used to model the potential of deep learning to model such complex degradation behaviors, offering enhanced flexibility and feature extraction capabilities. However, our study demonstrates that traditional ML models, when applied effectively, can achieve competitive performance metrics even without the sophisticated self-learning mechanisms of DL.

Using a health index function to assist find threshold values in the deterioration process, a double LSTM architecture that can detect change points in sensor data was introduced by Shi and Chehade [5]. applying a linear piecewise function with a 130 threshold, Ayodeji et al. [6] suggested a Causal Augmented Convolution Network (CaConvNet), which was further refined with a dynamic hyperparameter search approach to eliminate human tweaking and uncertainty. Traditional ML models require careful feature engineering and parameter tuning to achieve optimal results. Unlike deep learning, which learns features directly from raw data, ML approaches rely on domain expertise to extract relevant features, making them less flexible but often more interpretable, as demonstrated in our study. Using the piece-wise linear model, Zhang et al. [8] developed a novel technique for figuring out the threshold value depending on the properties of the dataset. In order to forecast aero-engine RUL, they created a GRU-based model with a high-level feature fusion block in place of conventional fully connected layers. They used a linear piecewise function with a 120 threshold across all C-MAPSS subsets [9] and the new activation function Mish. This threshold, which has been generally accepted as a standard model for RUL prediction in related studies, is experimentally observed and varies with dataset [10]. Interestingly, the study's threshold value, which is derived from [10] and is based on data from the first subset (FD001), is 130. Support Vector Regression (SVR) and Random Forest Regressor models were both used in the study by Zhiqiang Lyu and associates [11] to predict the Remaining Useful Life (RUL) of lithium-ion batteries. With an RMSE of 33.5 for the SVR model and 30.6125 for the Random Forest Regressor. This work can be viewed similar to prediction of RUL for air craft engines.

Remaining Useful Life (RUL) prediction was thoroughly examined using the C-MAPSS dataset in the study by Mathew Toby and associates [12]. Ten distinct machine learning models were used in the study, with a particular emphasis on the Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting models for Dataset 1. According to the results, the SVM model had an RMSE of 48.17, Random Forest model’s RMSE is 25.86 was significantly lower, and the Gradient Boosting model had an RMSE of 27.45. These results underline the robustness of Random Forest for RUL prediction in different contexts. A unique self-adapting deep learning network (CSDLN) was created in the study by Sheng et al. [13] to forecast the Remaining Useful Life (RUL) of aircraft engines. With an impressive RMSE of 15.977, this model demonstrated notable gains in prediction accuracy over earlier techniques. The CSDLN incorporates a trend identification unit to detect deterioration patterns and a multi-branch 1D involution neural network for feature extraction. The model's performance was validated using sophisticated approaches, including comparison and ablation tests, showcasing its ability to handle challenging RUL prediction tasks in aerospace applications. Fernando Sánchez Lasheras and associates [14] developed a hybrid model combining Support Vector Machines (SVM) and the Autoregressive Integrated Moving Average (ARIMA) approach to forecast the Remaining Useful Life (RUL) of aviation engines. Using the C-MAPSS dataset, their model achieved a Root Mean Squared Error (RMSE) of 39.6843, demonstrating significant prediction accuracy. Additionally, studies such as

Kang et al. [16] have shown the applicability of Artificial Neural Networks (ANNs) for production line equipment, further emphasizing the versatility of predictive models across domains. The use of explainable AI techniques, as proposed by Youness et al. [17], has proven essential for building trust in ML-based RUL systems, which our study also aims to address.

III. METHODOLOGY

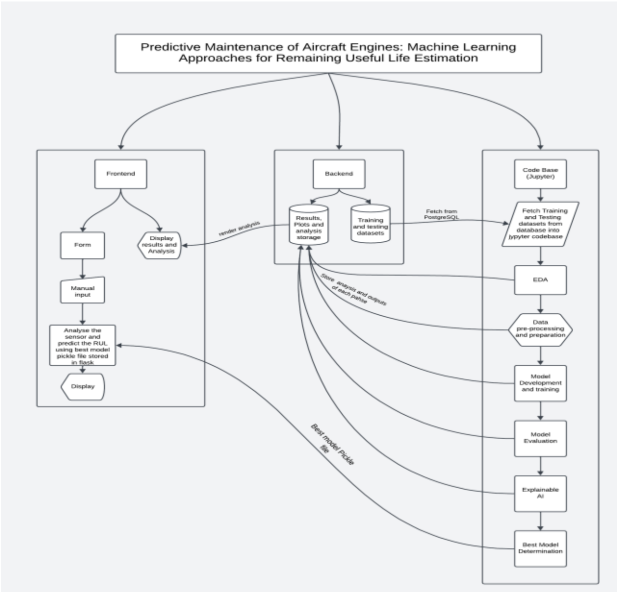

Fig. 1. Architecture of the project

In this research, a structured and systematic pipeline has been meticulously developed, leveraging techniques from diverse fields, including machine learning, database management, and web development. Advanced machine learning approaches, such as Explainable AI (XAI) techniques [17] [18] which can be used to explain the prediction method. Figure 1 depicts the overall architecture of the proposed work. The methodology for estimating aircraft engines' Remaining Useful Life (RUL) which integrates frontend interaction, backend data management, and a robust machine learning pipeline.

The backend acts as a central hub for data storage and retrieval, managing raw datasets, model outputs, and results. Training and testing datasets are stored in PostgreSQL, ensuring scalable and persistent storage. Results generated by the models, including performance metrics and visualizations, are saved back into the database for retrieval and display on the frontend.

The core machine learning pipeline, developed in a Jupyter environment, begins by importing training and testing datasets from the backend. Exploratory Data Analysis (EDA) is to comprehend the data and identify any patterns or noise. The data is then pre-processed and several machine learning models are trained on the pre-processed data. Performance metrics like RMSE and R^2 are calculated to evaluate each model. The best-performing model is saved as a pickle file and sent to the backend and these are visualized in a Flask based web page.

IV. IMPLEMENTATION

This section describes the end-to-end workflow, beginning from data collection, through model implementation and evaluation, to real-time analysis visualization on a Flask-based web interface as in figure 2. The operational workflow underscores the robustness and efficiency of the proposed system. Sensor data, collected and stored in PostgreSQL, undergoes rigorous preprocessing, including normalization and feature engineering, before being analyzed by machine learning models. Key outputs, such as predictive metrics and visualizations, are stored back in the database for frontend retrieval. The workflow ensures a bidirectional data pipeline, enabling both dynamic updates and historical analysis.

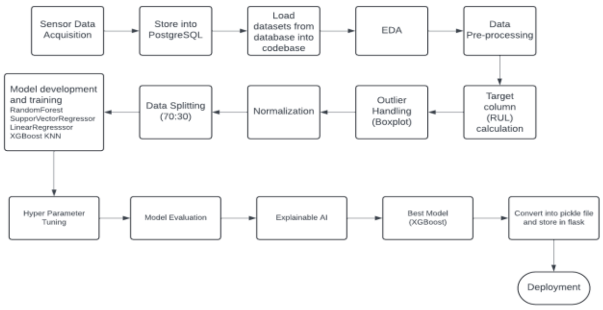

Fig. 2. Operational Workflow for Engine RUL Analysis and Prediction

A. *Problem Definition*

Even minor failure modes can have catastrophic effects on people and the economy. Consequently, a thorough Failure Mode and Effect Analysis (FMEA) and recovery techniques are needed. Data is gathered from the sensors, which are placed throughout the system in different locations. Gas, vibration, temperature, and pressure are examples of sensor types. The sensor data might also be continuous, discontinuous, or numerical [7]. The challenge addressed by this project is to develop a data-driven predictive model that estimates the engine's remaining cycles before failure (i.e., predicting the number of cycles left before shutdown), based on machine learning algorithms that estimates the Remaining useful lifetime with performance nearly equal to deep learning methods. This is critical for both operational efficiency and safety in the aerospace industry.

B. *Data Acquisition*

The dataset is collected from Kaggle named "NASA Turbofan Jet Engine Data Set". It is also available in NASA websit. In this there are different sub-datasets available, but we have used FD001 dataset which is Single operating mode and single fault condition:

C. *Data Storage:*

The collected datasets are imported and stored in PostgreSQL, a robust relational database system renowned for its effective handling of complicated queries and big datasets. Storing raw data and results in PostgreSQL creates a unified backend for benchmarking, tracking improvements, and powering real-time analysis in the Flask web app.

D. *Data Preparation:*

Both the training and testing datasets were retrieved from PostgreSQL using the psycopg2 library. And did basic preprocessing steps i.e., null values check, data type check, etc., which are critical for developing effective ML models. Some major steps in this phase are:

1) *Exploratory Data Analysis (EDA)*

Exploratory Data Analysis (EDA) is a cornerstone of our RUL prediction work, as it provides critical insights into the dataset and ensures a solid foundation for building accurate models. Through EDA, we can understand the distribution of features such as sensor readings and operational settings, identify trends in the Remaining Useful Life (RUL), and uncover underlying patterns, relationships between features and the target variable. It helps detect outliers and noise, which, if left unaddressed, could adversely impact model performance. During this phase, we investigated the following aspects:

a) *Engine Count:* The number of unique engines was identified from the dataset's Engine column. This count is critical for creating the target column representing the Remaining Useful Life (RUL) for each engine. The visualization in the histogram in Figure 3 categorizes engines by their maximum operational cycles, providing insights into the frequency of specific lifespan ranges and the Figure 4 reveals the operational lifespan distribution of the 100 engines in the training dataset. A clear pattern is visible, showing a clustering of engines around specific maximum cycle ranges, which indicates potential trends in engine degradation rates. This clustering informs the choice of RUL thresholds and highlights the consistency of degradation behavior across engines, reinforcing the need for tailored preprocessing and thresholding (e.g., capping RUL values at 130 cycles).

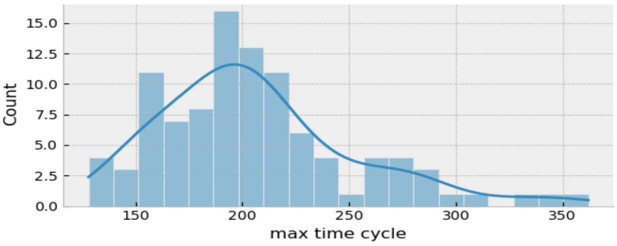

Fig. 3. Count of Engines by Maximum Cycles (100 Engine Sample)

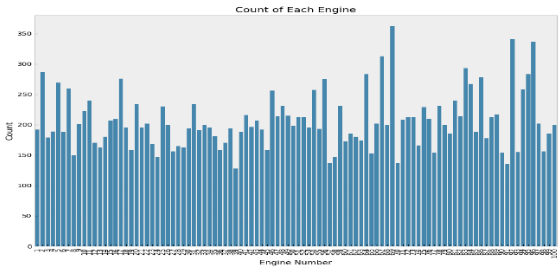

Fig. 4. Max cycle count of each engine (100 engines)

b) *Identified features' trend:* A time-series analysis of each feature was conducted across multiple engines. We visualized the trends of sensor readings over cycles to identify increasing, decreasing, and static trends. Features with a low standard deviation (i.e., less than 0.02) were considered irrelevant due to their low variability, and thus were removed from both training and testing datasets. Specifically, we identified the following features as having minimal contribution to RUL prediction: 'op_setting_1', 'op_setting_2', 'op_setting_3', 's1', 's5', 's6', 's10', 's16', 's18', 's19'. Figure 5 visualizes the trend of each feature for 10 random engines.

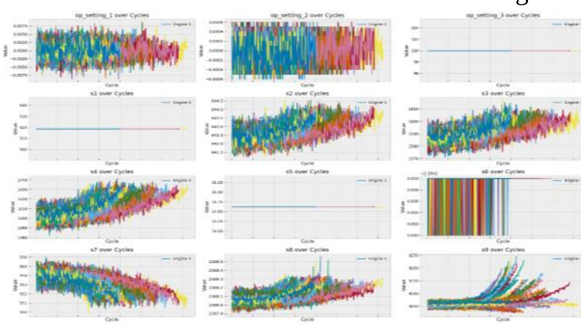

Fig. 5. Feature Behaviour Across 10 Randomly Chosen Engines

c) *Histograms and Boxplots:* These techniques were employed to examine the distribution of values within each sensor reading. Histograms helped us understand the overall distribution of sensor values, while box plots were used to visually identify and remove outliers by displaying data points outside the whiskers, which represent 1.5 times the interquartile range (IQR). This method effectively highlights extreme values allowing for the detection of noise and anomalies in the sensor readings and operational settings. By focusing on these outliers, you were able to clean the dataset and guarantee that only legitimate data was utilized to train the model, increasing the predictions' precision and dependability. Features with extreme outliers were either corrected or removed to aid in improving the overall performance of the model. Figures 6 and 7 visualize the box plot and histogram of sensor 2.

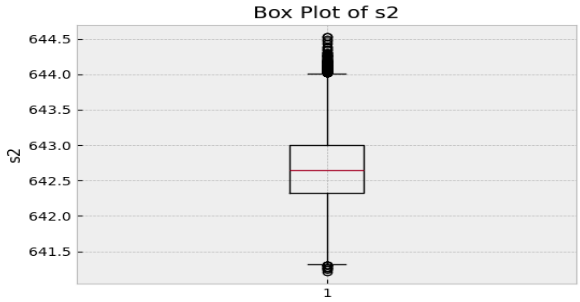

Fig. 6. Box plot analysis for sensor 2 (outlier detection)

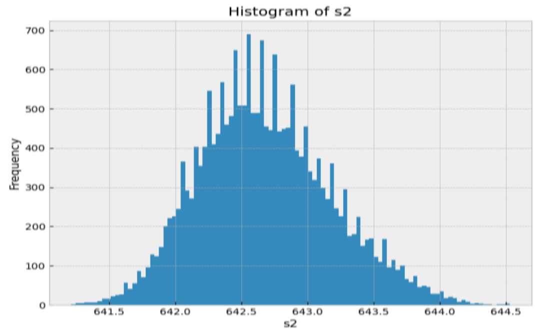

Fig. 7. Distribution of Sensor 2 Feature Values (Histogram)

d) *Correlation Matrix:* A correlation matrix was constructed to measure the degree of correlation between each sensor feature. Features that exhibited high collinearity were removed to prevent multicollinearity. Figure 8 shows the correlation between different features.

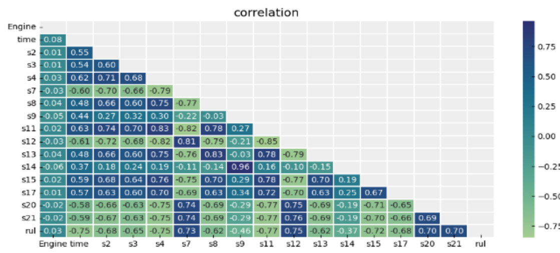

Fig. 8. Correlation Matrix

2) *Data Processing*

a) *Target column Construction:* The target column, representing the Remaining Useful Life (RUL) for each engine, was formulated based on the engine's total operational cycles. As in equation 1, RUL was determined by calculating the difference between the current cycle and the maximum cycle recorded for each engine. It was also observed that engines do not exhibit degradation from the initial cycles; therefore, a threshold of 130 was introduced. This means that any calculated RUL value exceeding 130 was capped at 130. This assumption significantly enhanced model performance. While various studies have employed different threshold values, none have provided a clear rationale. This derived feature serves as the principal target variable for all machine learning models in this study. Accurate construction of the RUL column is pivotal, as it directly influences the models' predictive capabilities in estimating the remaining operational cycles before engine failure.

$$RUL_i=\min(C_{max,i}-C_{current,i},\ T) \quad (1)$$

- RUL_i represent the Remaining Useful Life of engine i,
- $C_{max,i}$ represent the maximum cycle for engine i,
- $C_{current,i}$ represent the current cycle for engine i,
- Threshold T=130. (assumed)

b) *Data Normalization:* The dataset was processed using various scaling techniques, including MinMaxScaler and StandardScaler. It was observed that superior model performance was achieved when using StandardScaler for data normalization. This is attributed to the fact that MinMaxScaler scales data between the minimum and maximum values within the dataset, which can vary across training, validation, and testing sets. As a result, MinMaxScaler can introduce inconsistencies, making it difficult to generate accurate predictions. In contrast, StandardScaler standardizes the data based on the mean and standard deviation, ensuring consistency across datasets while normalizing and improving model reliability.

$$x_j' = (x_j-\mu_j)/\sigma_j \quad (2)$$

- x_j' is the scaled value,
- μ_j is the mean of feature j,
- σ_j is the standard deviation of feature j.

c) *Data splitting:* Training data is divided into feature and target columns and then splitted into training and validation dataset at 70:30 ratio for better validation.

- Features: 's2', 's3', 's4', 's7', 's8', 's9', 's11', 's12', 's13', 's14', 's15', 's17', 's20', 's21'
- Target: RUL

E. *Model Building*

1) *Model Selection*

During this phase, Several models of machine learning were developed and trained using the normalized training data.We have taken basic LR model to Ensemble learning XGBoost model. The predictive results were stored in PostgreSQL for future reference and for dynamic rendering on a web interface. The primary objective of these models was to accurately estimate the number of operational cycles remaining from the current cycle. The models used are Random Forest, Support Vector Regressor, Linear Regressor, XGBoost, and KNN Regressor.

The selection of these algorithms i.e., Random Forest, Support Vector Regressor (SVR), Linear Regressor, XGBoost, and KNN Regressor for predicting Remaining Useful Life (RUL) reflects a balanced and strategic approach to highlight the power of machine learning. For example, Random Forest offers feature relevance metrics to assist in identifying the most important sensors causing engine deterioration and is ideally suited for capturing both linear and non-linear correlations in complicated datasets. When working on a project that involves multi-sensor data, this interpretability is essential. Support Vector Regressor, on the other hand, is best suited for small to medium-sized datasets with high dimensionality. Its capacity to describe nonlinear interactions using kernel functions assures reliable predictions, especially in situations where typical linear models may fail. Despite its simplicity, the linear regression serves as an important baseline, providing interpretability and giving a reference point for assessing the added value of more complicated models. XGBoost advances the analysis by effectively capturing intricate patterns in data using its gradient boosting approach. Its resistance to noise and capacity to tolerate missing data make it a dependable choice for datasets including real-world errors. Finally, the KNN Regressor offers a new perspective by focusing on proximity-based predictions, which effectively use local patterns or clusters in the dataset.

The performance of each model was evaluated using RMSE and R^2 scores. For an ideal model, RMSE should be as close to 0 as possible, indicating minimal prediction error, and the R^2 score should close to 1, signifying a strong fit to the data.

2) *Hyper Parameter Tuning:*

Hyperparameter tuning played a crucial role in optimizing model performance. To achieve the best possible results, we have employed both Grid Search and Randomized Search to identify the optimal hyperparameters for your models. While Grid Search methodically evaluates every possible combination of hyperparameters, it can be computationally expensive and time-consuming, especially when the hyperparameter space is large. But Randomized Search takes a more efficient approach by sampling random combinations from the hyperparameter space, making it significantly faster and more suitable for complex models with multiple hyperparameters. Randomized Search lowers the chance of overfitting to a particular grid because it encompasses more possibilities without requiring the same level of thorough testing as Grid Search. As a result, Randomized Search not only saved valuable time and resources but also provided better model performance by identifying robust parameter combinations.

F. *Model Evaluation and Best model Selection*

After model development and training, overfitting and underfitting were assessed by calculating RMSE and R^2 scores for both training and validation datasets. A model is balanced when both scores are nearly identical. These predictions were compared with a reference dataset that provides the actual remaining cycles, ensuring accurate model evaluation. The XGBoost Regressor was selected as the best model due to its robust performance, demonstrating neither overfitting nor underfitting. It achieved an acceptable RMSE (<25) and an R^2 score (65-70), metrics typically associated with deep learning models, thus aligning well with the project's objectives. This model is serialized into a pickle file and stored within the Flask application to facilitate RUL predictions based on input from the form data.

G. *Implementation of Explainable AI(XAI):*

Explainable Artificial Intelligence (XAI) techniques were used to assess the importance of each feature in predicting RUL. Feature importance scores were generated using the XGBoost model, which helped identify key factors contributing to RUL predictions. These insights improved model accuracy and informed feature selection. As shown in Figure 9, the feature importance scores clearly illustrate the relative significance of each attribute, highlighting the most influential features for RUL prediction.

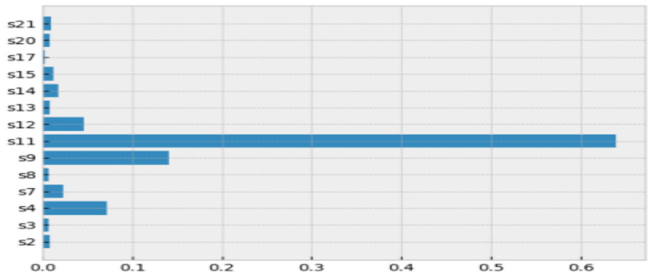

Fig. 9. Feature importance using XGBoost technique.

H. *Web page rendering:*

We have developed a Flask-based web application that dynamically renders the analysis of the models, adapting in real-time as the data within PostgreSQL evolves due to changes in the dataset or fluctuations in model performance. The application allows users to input sensor data for dynamic RUL prediction using the optimized model stored in a pickle file. This enhances user engagement, provides engine health insights, and supports informed maintenance decisions. The web app seamlessly integrates with the database, ensuring responsive updates with the latest predictions and analytics. Figure 10 shows the Flask-based web page.

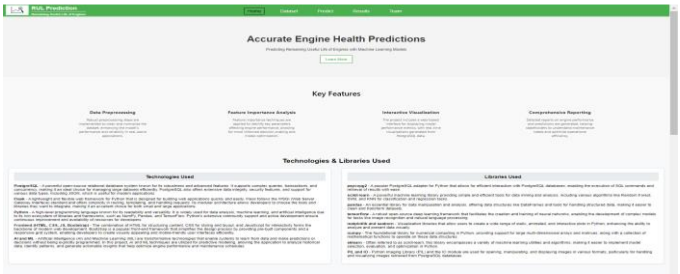

Fig. 10. Flask based web page images

V. RESULTS

A. *Model Performance*

1) *Model Development and Evaluation*

To forecast Remaining Useful Life (RUL) in aircraft engines, a range of machine learning models was rigorously evaluated, including Random Forest Regressor, XGBoost Regressor, Support Vector Regressor, Linear Regressor, and K-Nearest Neighbors. These models were meticulously trained and assessed using historical engine operational data, enabling the accurate differentiation between engines nearing failure and those operating optimally.

a) RandomForestRegressor:

- Training Set: RMSE: 15.80, R^2: 0.87
- Validation Set: RMSE: 23.24, R^2: 0.71
- Test Set: RMSE: 28.16, R^2: 0.54

These metrics indicate that while the model demonstrates strong performance on the training dataset, it exhibits a significant decline in predictive accuracy when applied to the testing dataset. his disparity indicates that the model is overfitting, capturing noise and specific patterns from training data is used instead of effectively generalizing to new data. The results of the Random Forest Regressor are visualized in Figure 11, which compares predicted versus actual Remaining Useful Life (RUL) values. While the model performs well on the training set, overfitting is apparent, as evidenced by significant deviations in test predictions, particularly for engines nearing the end of their life cycles. This inconsistency demonstrates how poorly the model generalizes unseen data.

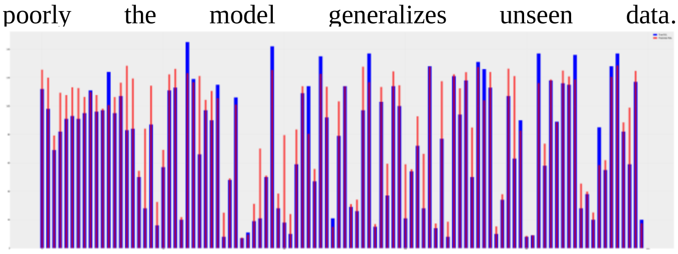

Fig. 11. Actual RUL (blue) vs Predicted RUL (red) of RandomForestRegressor

b) Support Vector Regressor

The performance metrics associated with the Support Vector Regressor (SVR) model are delineated as follows:

- Train set RMSE:19.768, R2:0.795
- Validation set RMSE:23.987, R2:0.688
- test set RMSE:23.784, R2:0.672

The Support Vector Regressor demonstrates an admirable capacity to fit the training data, its performance on unseen data underscores a need for enhancement in model generalization. In Figure 12, the predictions of the Support Vector Regressor (SVR) are compared against the actual RUL values. The SVR shows better generalization than the Random Forest Regressor, maintaining consistent performance across datasets. However, as seen in this figure, the model struggles with extreme RUL values, where prediction errors become more pronounced.

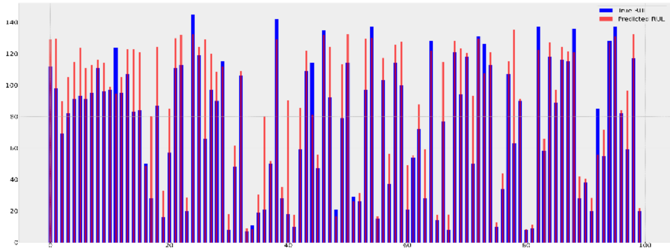

Fig. 12. Actual RUL (blue) vs Predicted RUL (red) of SupportVectorRegressor

c) Linear Regressor

- train set RMSE:22.628, R2:0.731
- validation set RMSE:24.961, R2:0.663
- test set RMSE:29.0749, R2:0.51

The Linear Regressor provides a baseline performance, its inability to maintain predictive accuracy across different datasets indicates a potential for underfitting. According to the results, the Linear Regressor does well on the training dataset. These results indicate that the model captures a considerable portion of the variance in the training dataset. However, performance measurements for both the validation and test datasets show a considerable reduction.

The increase in RMSE and the corresponding decrease in R^2 for the validation and test sets suggest that the Linear Regressor is struggling to generalize effectively to unseen data. The R^2 score of 0.51 on the test set implies that less than half of the variance in the data is being explained, which raises concerns about the model's robustness in predictive tasks. The Linear Regressor's performance is shown in Figure 13, emphasizing its limitations..

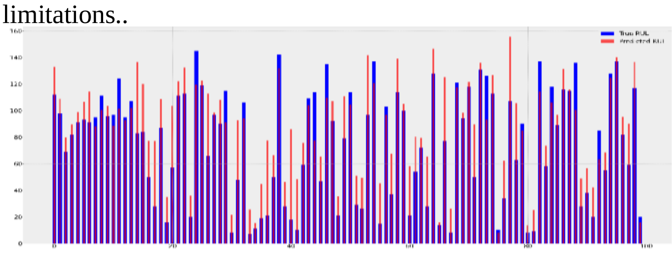

Fig. 13. Actual RUL (blue) vs Predicted RUL(red) of LinearRegressor

d) XGBoost Regressor:

- train set RMSE:20.581, R2:0.778
- validation set RMSE:22.501, R2:0.726
- test set RMSE:23.801, R2:0.672

The results confirm that the model accurately predicts engine RUL, capturing underlying patterns with high training accuracy. A low RMSE indicates minimal disparity between predicted and actual values, while an R^2 of 78% explains most of the training data variance. Validation results show the model generalizes well, with slightly higher but acceptable RMSE, avoiding overfitting. On the test set, the model demonstrates robustness, accurately predicting RUL and explaining 67% of the variance. Figure 14 illustrates XGBoost's superior performance, showing close alignment between predicted and actual RUL values. Its robustness and minimal deviations stem from effective handling of complex patterns and regularization techniques.

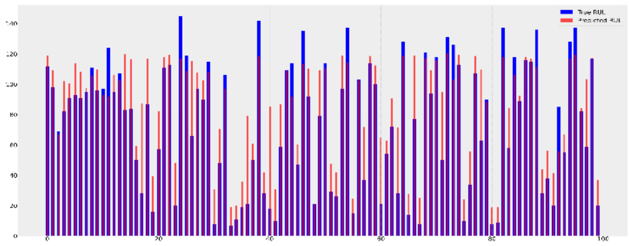

Fig. 14. Actual RUL (blue) vs Predicted RUL (red) of XGBoostRegressor

e) KNN

- train set RMSE:29.169, R2:0.55
- validation set RMSE:30.435, R2:0.499
- test set RMSE:30.735, R2:0.453

The K-Nearest Neighbors (KNN) model shows limited accuracy and predictive power across all datasets. Its declining performance on validation and test sets highlights its inability to generalize beyond the training data. The low R^2 scores suggest the model fails to explain a significant portion of the variance, confirming its inadequacy for Remaining Useful Life (RUL) prediction. Figure 15 further illustrates the KNN model's poor performance, with substantial deviations between predicted and actual RUL values. This underperformance highlights the model's unsuitability for predictive maintenance tasks.

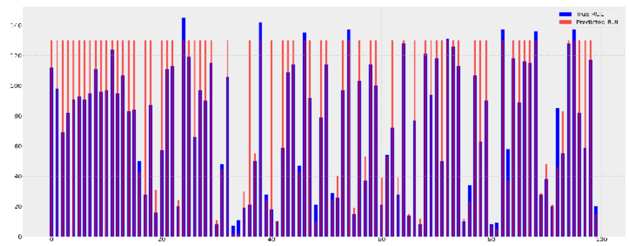

Fig. 15. Actual RUL (blue) vs Predicted RUL(red) of KNN

2) Importance of Key Features

Following the modelling phase, we conducted a feature importance analysis using the XGBoost framework. To comprehend how different attributes contributed to the model's predictions, feature importance mapping was used. Notably, the following features were identified as most influential: 's4', 's7', 's9', 's11', 's12'.

These attributes play a vital role in predicting RUL of each These attributes play a vital role in predicting RUL of each engine. This can be observed from figure 16.

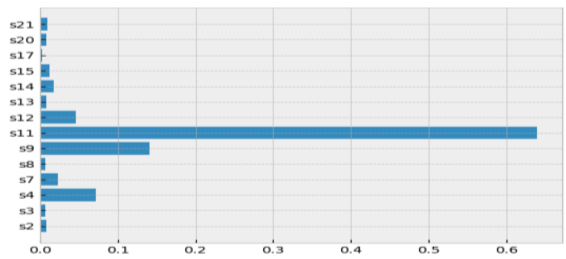

Fig. 16. Feature Importance graph for XGBoostRegressor

3) RUL prediction from User entered data from Form

A user input form has been integrated into the web interface, enabling manual sensor data submission as an alternative to engine data acquisition. This allows accurate RUL predictions based on user-provided data, comparable to those from the internal algorithm. The data seamlessly transfers to the Flask application for processing. Figure 17 displays the web page for user input and RUL prediction.

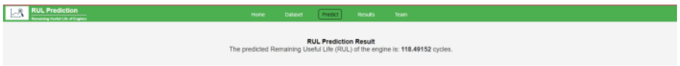

Fig. 17. Predicting RUL from User-Entered Data via Pickle File

4) Observations and Insights

The XGBoost Regressor has emerged as the leading model for estimating engines' Remaining Useful Life (RUL), displaying an admirable mix of accuracy and generality. The training set results show an RMSE of 20.58 and a R^2 score of 0.78, suggesting the model accurately captures the training data without overfitting. This is demonstrated by its relatively low RMSE on both the validation (22.50) and test sets (23.80), which remain strongly related to training performance. Table 1 shows the RMSE and R2 score of training and testing for different models.

TABLE 1: PERFORMANCE FOR EACH MODEL

Model Comparison				
Model	***Training RMSE***	***Train R^2 Score***	***Testing RMSE***	***Test R^2 Score***
Rand. Forest	15.79	0.86	28.15	0.54
SVR	19.76	0.79	23.7	0.67
Lin. Regressor	22.62	0.73	29.07	0.51
XGBoost	20.58	0.77	23.80	0.67
KNN	29.16	0.55	30.73	0.45

XGBoost outperforms models like Support Vector Regressor and Random Forest Regressor, demonstrating resilience against overfitting and underfitting. Random Forest showed overfitting, with a training RMSE of 15.80 and test RMSE of 28.16, while the Support Vector Regressor had an R^2 of 0.67, limiting its predictive accuracy. XGBoost's robust optimization, feature importance handling, and regularization effectively balance the bias-variance trade-off, ensuring high accuracy and generalization across datasets. Its consistent performance in training and testing establishes it as the optimal choice for RUL prediction in predictive maintenance.

5) Comparative study

TABLE 2: COMPARISON OF RESULTS WITH OTHER WORKS

Comparative Analysis		
Reference/ Author name	***Model***	***RMSE***
Proposed work	XGBoost	RMSE = 23.8
Ellefsen et al., 2018 [5]	Semi Supervised based architecture	RMSE = 26
Mathew et al., 2017 [12]	Random Forest	RMSE = 24.9
Fernando Sánchez Lasheras [14]	Hybrid model of SVM and ARIMA	RMSE = 39.7
Kang et al., 2021 [16]	ANN	RMSE = 25.8

To demonstrate the efficacy of the suggested XGBoost model, its performance is contrasted with that of a number of other models that are already available in the literature. As shown in the table 2, the XGBoost model achieved an RMSE of 23.8, outperforming other machine learning and hybrid models. These comparisons underscore the superior accuracy and efficiency of the XGBoost model in predicting RUL, highlighting its potential over both traditional machine learning and deep learning approaches.

VI.CONCLUSION

This research work shows substantial advancement in predictive maintenance, showcasing the practical advantages of machine learning over more complex deep learning models. By utilizing regression techniques such as XGBoost, Random Forest, and Support Vector Regressor, the system achieved accurate Remaining Useful Life (RUL) predictions while maintaining computational efficiency. XGBoost demonstrated exceptional performance with low RMSE and high R^2 scores, rivaling deep learning models like LSTM without their inherent complexity and resource demands. This underscores the practicality of using machine learning in scenarios where interpretability, efficiency, and ease of implementation are crucial. A defining strength of this system lies in its explainability, achieved through XAI techniques that quantify feature importance. By bridging the gap between unprocessed sensor data and useful insights, this interpretability helps stakeholders to understand the factors influencing RUL predictions. Unlike deep learning models, which frequently act as black boxes, this machine learning approach ensures transparency and confidence in the process of making decisions, making it particularly suited for industries like aviation, where safety and accountability are critical. In contrast to deep learning models, which require extensive computational resources and large datasets, this project demonstrates that machine learning can achieve comparable predictive accuracy with fewer resources. The lightweight yet powerful nature of these models highlights their suitability for practical deployment in predictive maintenance systems, where timely and accurate insights are vital. By reducing unplanned downtime, optimizing maintenance schedules, and ensuring operational reliability, this system addresses a critical need in the aviation sector while laying the groundwork for further innovation in predictive analytics.

VII. FUTURE ENHANCEMENT

As we continue to work on estimating the Remaining Useful Life (RUL) of aviation engines utilizing machine learning models and explainable AI strategies, identifying future advancements will be critical for boosting both model efficacy and interpretability. As the models continue to evolve, they will provide increased operational reliability and safety, with better informed decision-making processes and driving cost-efficient maintenance planning.

1) We have used 130 as threshold value. Experimenting with different threshold values, other than 130, can improve the RUL target column by better aligning with real-world engine degradation, enhancing model accuracy and performance.

2) We have implemented box plot technique for noise removal. Implementing advanced outlier detection algorithms will help identify and remove noisy data points, improving model stability and accuracy for time-series engine sensor data.

3) Using techniques like RFE or LASSO to refine feature selection will improve model performance and reduce computational complexity by focusing on relevant data.

4) Implementing other ML models, i.e., ensemble methods, improves forecast accuracy and identifies strengths and limitations of different approaches.

VIII. REFERENCES

[1] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech and N. Zerhouni, "Direct Remaining Useful Life Estimation Based on Support Vector Regression," in IEEE Transactions on Industrial Electronics, vol. 64, no. 3, pp. 2276-2285, March 2017, doi: 10.1109/TIE.2016.2623260.

[2] Ma, J.; Su, H.; Zhao, W.-L.; Liu, B. Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning. Complexity 2018, 2018, 3813029.

[3] Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. Mech. Syst. Signal Process. 2019, 115, 213–237.

[4] J. Xu and L. Xu, "Health management based on fusion prognostics for avionics systems," in Journal of Systems Engineering and Electronics, vol. 22, no. 3, pp. 428-436, June 2011, doi: 10.3969/j.issn.1004-4132.2011.03.010.

[5] Ellefsen, A.L.; Bjørlykhaug, E.; Æsøy, V.; Ushakov, S.; Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab. Eng. Syst. Saf. 2018, 183, 240–251.

[6] Ayodeji, A.; Wang, Z.; Wang, W.; Qin, W.; Yang, C.; Xu, S.; Liu, X. Causal augmented ConvNet: A temporal memory dilated convolution model for long-sequence time series prediction. ISA Trans. 2021, 123, 200–217.

[7] Sharanya, S., Venkataraman, Revathi and Murali, G.. "Predicting remaining useful life of turbofan engines using degradation signal based echo state network" International Journal of Turbo & Jet-Engines, vol. 40, no. s1, 2023, pp. s181-s194. https://doi.org/10.1515/tjj-2022-0007.

[8] Zhang, Y.; Xin, Y.; Liu, Z.-W.; Chi, M.; Ma, G. Health status assessment and remaining useful life prediction of aero-engine based on BiGRU and MMoE. Reliab. Eng. Syst. Saf. 2022, 220, 108263. https://doi.org/10.1016/j.ress.2021.108263

[9] A. Saxena, K. Goebel, D. Simon and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 2008, pp. 1-9, doi: 10.1109/PHM.2008.4711414.

[10] Wang, Y.; Zhao, Y. Multi-Scale Remaining Useful Life Prediction Using Long Short-Term Memory. Sustainability 2022, 14, 15667. https://doi.org/10.3390/su142315667.

[11] Wang, G.; Lyu, Z.; Li, X. An Optimized Random Forest Regression Model for Li-Ion Battery Prognostics and Health Management. Batteries 2023, 9, 332. https://doi.org/10.3390/batteries9060332

[12] V. Mathew, T. Toby, V. Singh, B. M. Rao and M. G. Kumar, "Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning," 2017 IEEE International Conference on Circuits and Systems (ICCS), Thiruvananthapuram, India, 2017, pp. 306-311, doi: 10.1109/ICCS1.2017.8326010.

[13] Deng, S., Zhou, J. Prediction of Remaining Useful Life of Aero-engines Based on CNN-LSTM-Attention. Int J Comput Intell Syst 17, 232 (2024). https://doi.org/10.1007/s44196-024-00639-w

[14] Ordóñez, C.; Lasheras, F.S.; Roca-Pardiñas, J.; Juez, F.J.D.C. A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. J. Comput. Appl. Math. 2018, 346, 184–191. https://doi.org/10.1016/j.cam.2018.07.008.

[15] Mutunga, Jackline Mwende et al. “Health-Index Based Prognostics for a Turbofan Engine using Ensemble of Machine Learning Algorithms.” (2019).

[16] Kang Z, Catal C, Tekinerdogan B. Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks. Sensors (Basel). 2021;21(3):932. Published 2021 Jan 30. doi:10.3390/s21030932.

[17] Youness G, Aalah A. An Explainable Artificial Intelligence Approach for Remaining Useful Life Prediction. Aerospace. 2023; 10(5):474. https://doi.org/10.3390/aerospace10050474.

[18] Protopapadakis, G, Apostolidis, A, & Kalfas, AI. "Explainable and Interpretable AI-Assisted Remaining Useful Life Estimation for Aeroengines." Proceedings of the ASME Turbo Expo 2022: Turbomachinery Technical Conference and Exposition. Volume 2: Coal, Biomass, Hydrogen, and Alternative Fuels; Controls, Diagnostics, and Instrumentation; Steam Turbine. Rotterdam, Netherlands. June 13–17, 2022. V002T05A002. ASME. https://doi.org/10.1115/GT2022-80777.