Predictive Maintenance and Risk Assessment of Turbofan Engines Using Classical ML

Dhruv Nandigam
SE23UCSE055

D.L.C Siddarth
SE23UCSE053

Anirudh Vuppala
SE23UCSE007

P.V. Karthik
SE23UCSE132

Rithvik Guddati
SE23UCSE199

***Abstract*—The maintenance of turbofan engines is a critical task in ensuring the reliability and safety of aerospace operations. This work presents a multi-phase approach to predict engine degradation stages and assess maintenance risks using the NASA CMAPSS dataset. In Phase 1, we develop a clustering-based method to segment engine health into distinct degradation stages, leveraging PCA and KMeans while addressing noise during stage transitions. Phase 2 involves classification of these degradation stages using Random Forest, SVM, and Ridge classifiers, evaluated through precision, recall, and F1-Score. In Phase 3, we predict the time remaining until the next stage transition through regression techniques, including Random Forest Regressor, Ridge Regression, and SVR, focusing on minimizing RMSE, MAE, and R^2. In Phase 4, we compute a risk score combining failure probability and time left to failure, utilizing classifier output and regression predictions. The risk score is normalized using Min-Max and urgency-based techniques, with thresholds tuned using Precision-Recall curves to optimize maintenance alerts. Our approach demonstrates the ability to predict engine degradation and quantify risk, providing a foundation for proactive maintenance planning.**

I. Introduction

Predictive maintenance plays a crucial role in the aerospace industry, where the reliability and safety of turbofan engines are paramount. Sudden engine failures can result in significant operational downtime, economic losses, and, more critically, safety risks. To mitigate these challenges, predictive maintenance strategies aim to detect potential failures before they occur, allowing for timely maintenance actions.

One of the major challenges in predictive maintenance is accurately predicting engine degradation stages. Turbofan engines exhibit complex operational dynamics, where sensor data may reflect gradual or abrupt changes in engine health. Moreover, distinguishing between normal variability and actual degradation patterns is a non-trivial task due to the inherent noise and variability in sensor measurements.

In this work, we address the problem of predicting degradation stages in turbofan engines using the NASA CMAPSS dataset. Our objective is to accurately classify engine health states and predict the time remaining until the next degradation stage, enabling proactive maintenance decisions. By segmenting engine health into distinct stages and predicting stage transitions, we aim to enhance maintenance planning and reduce the risk of unplanned engine failures.

Our approach consists of a multi-phase methodology designed to address different aspects of the problem. First, we cluster sensor data to identify degradation stages, leveraging dimensionality reduction techniques to handle high-dimensional data. Next, we employ classification algorithms to identify the current degradation stage based on real-time measurements. To predict the time remaining until the next stage transition, we utilize regression techniques that model the temporal progression of engine health. Finally, we calculate a risk score by combining failure probability with the estimated time to the next critical stage, which helps prioritize maintenance actions.

This comprehensive approach not only provides insights into the current state of engine health but also forecasts the remaining useful life until the next significant degradation stage. By integrating classification and regression models with risk assessment, our methodology contributes to proactive maintenance strategies, potentially reducing unexpected engine failures and maintenance costs.

II. Methodology

The proposed methodology for predicting engine degradation and assessing maintenance risk consists of four key phases: clustering for stage labeling, classification for degradation stage identification, regression for time-to-next-stage prediction, and risk score computation. Each phase addresses a specific challenge associated with the predictive maintenance of turbofan engines. Below, we detail each phase, including data preprocessing, model selection, and evaluation techniques.

A. Clustering-Based Stage Labeling

In this phase, we segment the engine health data into distinct degradation stages. Initially, we verified that the dataset did not contain any missing values, eliminating the need for imputing or dropping rows. We then examined the raw sensor data, focusing on the standard deviation of each sensor to identify those with minimal variance. Since our clustering approach focuses on engine health based on sensor readings, we excluded the operating conditions from this analysis. After calculating the standard deviation of each sensor, we discarded sensors that exhibited minimal change across the dataset, as they contributed little to identifying degradation stages.

Next, we standardized the selected sensors to ensure that they all had a mean of zero while preserving the variance within each feature. This step helped maintain uniformity across the data, making the clustering process more robust. To better understand the data, we visualized sensor readings from randomly selected engine samples. These plots helped identify patterns and variability among the sensors over time, giving us insights into which features might be most informative for clustering.

To address potential noise in the data, we applied mean-windowed smoothing with a window size of five, effectively reducing the impact of sporadic outliers while retaining the underlying trend of the sensor measurements. Following data smoothing, we employed dimensionality reduction techniques to facilitate clustering. Initially, we used both PCA and t-SNE to visualize the data distribution and identify possible clusters. While both methods highlighted variability in the data, manual inspection revealed that identifying clusters from these visualizations was challenging.

Upon further analysis, we determined that the first and second principal components (PC1 and PC2) captured approximately 95

After determining the importance of PC1, we experimented with clustering methods, specifically KMeans and Agglomerative Clustering, using both PC1 and the combined features PC1+PC2. Visual examination of the cluster separation showed that KMeans, when applied solely to PC1, produced the most distinct and meaningful clusters. Consequently, we selected this configuration for stage labeling.

To translate the clusters into meaningful degradation stages, we developed a simple mapping from cluster ID to stage name, allowing for consistent labeling and prediction of engine health states. An example visualization of predicted clusters along with the cycle count and PC1 for a specific engine is shown in Figure 2.

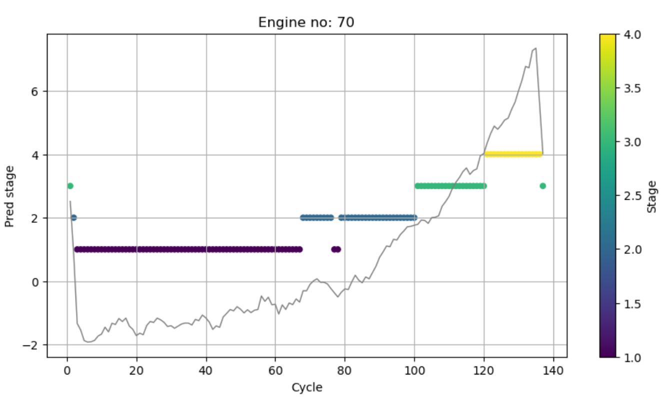

Fig. 1. Predicted clusters along with the cycle count and PC1 for engine no 70.

B. Classification of Degradation Stages

After establishing stage labels, the next phase involves training classifiers to predict the current degradation stage from real-time sensor measurements. To prevent data leakage, we split the data early based on engine numbers, ensuring that training and testing sets remained independent. We saved the test engine IDs to maintain consistency across experiments.

Outlier handling was performed using Z-score and IQR methods, supported by visual inspection through boxplots. Rather than dropping the detected outliers, we replaced them with rolling median or mean values, retaining the data structure while minimizing noise. Additionally, we logged the outlier rate for each sensor to monitor data quality.

During exploratory data analysis (EDA), we examined the class distribution across degradation stages (0 to 4) and visualized sensor value distributions to understand changes with degradation. We plotted pairplots and PCA projections to inspect data separability by stage, and calculated correlation matrices to identify the most informative sensors.

For feature engineering, we created rolling statistics (mean, standard deviation) over a window of five cycles and computed delta values between consecutive cycles. We also calculated the time since the last stage change to capture temporal patterns. After feature extraction, highly correlated or low-variance features were dropped to improve model robustness.

We trained multiple classifiers, including Logistic Regression with balanced class weights, SVM (linear and RBF), Random Forest, and XGBoost. To address class imbalance, we applied SMOTE to the training data and used class weight adjustments within the models where applicable. We evaluated model performance using precision, recall, and F1-score, focusing on stages 3 and 4, and visualized the results with confusion matrices. Feature importance was analyzed using SHAP values and permutation importance, providing insights into key predictive features.

The final model selection was based on accuracy and validation metrics, and the best-performing model was saved with a standardized loading function for future use.

This multi-step approach ensures that the classification model not only accurately predicts degradation stages but also generalizes well to unseen engine data, thereby supporting proactive maintenance decisions.

C. Regression for Time-to-Next-Stage Prediction

Following the classification of engine stages, we predict the remaining cycles until the next degradation transition. To achieve this, we generate a target column labeled ”Time to Next Stage” by calculating the difference between consecutive stage changes. This target variable is crucial for forecasting the degradation timeline.

We experiment with three regression models: Random Forest Regressor, Ridge Regression, and Support Vector Regressor (SVR). The Random Forest Regressor captures complex, non-linear relationships between the input features and the target variable. The Ridge Regression model, with L2 regularization, stabilizes the predictions by minimizing the influence of highly correlated features. On the other hand, SVR leverages the kernel trick to model non-linear dependencies effectively, making it suitable for predicting remaining useful life (RUL) in scenarios with high-dimensional data. The regression models are evaluated based on Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2), which collectively assess both the accuracy and consistency of predictions.

D. Risk Score Computation and Decision Logic

The final phase of the methodology involves computing a risk score to quantify the urgency of maintenance actions. The risk score is formulated as the product of the failure probability, obtained from the classifier, and the estimated time to the next degradation stage, derived from the regression model. This combined score provides a comprehensive measure of engine health risk.

To normalize the risk score and make it comparable across engines, we employ two techniques. The first technique, Min-Max Normalization, scales the raw score to a range between 0 and 1, providing a relative measure of risk. The second technique, Urgency-Based Inversion, computes the risk score as the ratio of the failure probability to the estimated time left, adding a small constant to avoid division by zero. These normalized scores are then compared against a predefined threshold to issue maintenance alerts, allowing for proactive intervention. We tune the alert threshold using Precision-Recall curves to find a balance between timely warnings and false alarms.

[INSERT DIAGRAM]

By systematically combining classification outputs with time-to-failure predictions, our methodology provides an integrated risk assessment framework. This approach not only forecasts the next degradation stage but also assesses the urgency of maintenance actions, thus enabling better decision-making and reducing the risk of unexpected engine failures.

III. Results

In this section, we present the results obtained from each phase of the proposed methodology. The results are organized according to the phases outlined earlier, focusing first on clustering for stage labeling (Phase 1) and classification of degradation stages (Phase 2). Each phase is evaluated based on its primary objective: accurately segmenting engine health into degradation stages using clustering techniques in Phase 1, and predicting these stages from real-time sensor data using classification algorithms in Phase 2.

We present visualizations and performance metrics to demonstrate the effectiveness of each approach, including detailed analysis of model accuracy, stage separation, and classification performance. The results highlight key insights into the reliability of clustering methods for stage labeling and the predictive power of classification models in identifying engine degradation states. Subsequent sections will address the results for the regression model to predict the time to the next degradation stage (Phase 3) and the risk score computation for maintenance alerting (Phase 4).

A. Phase 1: Clustering-Based Stage Labeling

The primary objective of Phase 1 was to identify distinct degradation stages in turbofan engine data using clustering techniques. After performing data preprocessing, we utilized Principal Component Analysis (PCA) to reduce the dimensionality of the sensor data. PCA revealed that the first principal component (PC1) captured a significant portion of the variance, with PC1 and PC2 collectively accounting for approximately 95% of the total variance. Among these, PC1 was identified as the most informative feature, showing a clear upward trend as the engine cycle count increased, indicating progressive engine degradation.

To determine the most effective clustering approach, we experimented with both KMeans and Agglomerative Clustering. Clustering was performed using PC1 alone as well as the combination of PC1 and PC2. After visual analysis of the clustering patterns and evaluating cluster separability, we concluded that KMeans clustering using only PC1 provided the most distinct and interpretable stages. The clusters aligned well with expected degradation patterns, where the cluster IDs could be mapped to stage labels (from 0 to 4), representing increasing levels of degradation.

The clustering results are visualized in Figure 2, where PC1 values are plotted against engine cycle count for a representative engine. The distinct clusters are shown with color-coded labels, highlighting the progression through degradation stages over time. This visual representation demonstrates that the clusters formed by KMeans on PC1 align with the gradual degradation observed in real engine cycles.

One of the key challenges during clustering was identifying a reliable feature for stage separation. Initial attempts using both PC1 and PC2 proved less effective, as PC2 introduced ambiguity without adding significant differentiation between stages. Through Kernel Density Estimation (KDE) plots, we confirmed that PC1 was the dominant feature, capturing the degradation trend consistently. Additionally, smoothing the sensor data with a mean window of size 5 helped reduce noise, which contributed to clearer stage boundaries.

Mapping the identified clusters to degradation stages was performed manually based on the cluster distribution over engine cycles. The final mapping aligned well with domain knowledge, where early clusters corresponded to healthy states and later clusters to advanced degradation. This stage labeling serves as the foundation for subsequent classification tasks.

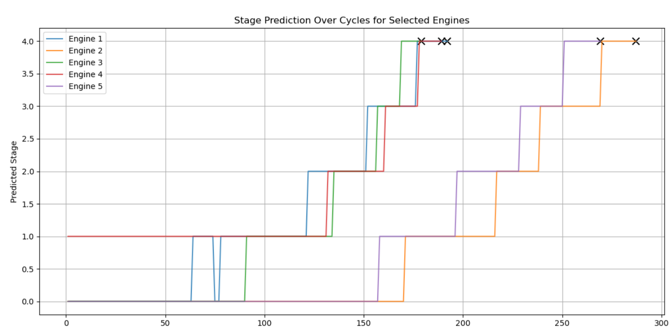

Fig. 2. Clustering results using PC1 over engine cycle count, showing identified degradation stages for a representative set of engines.

Overall, the clustering results demonstrate that using PC1 with KMeans provides a robust method for segmenting engine health states. This phase successfully establishes labeled degradation stages, forming the basis for the classification models in the subsequent phase.

B. Phase 2: Classification of Degradation Stages

The primary objective of Phase 2 was to accurately classify the degradation stages identified in Phase 1 based on real-time sensor measurements. After clustering the data to generate stage labels, we trained several machine learning classifiers to predict these stages from the processed sensor data. The models used included Logistic Regression with balanced class weights, Support Vector Machines (SVM) with an RBF kernel, Random Forest, and XGBoost. To handle class imbalance, we experimented with the Synthetic Minority Over-sampling Technique (SMOTE) during training.

During data preparation, we performed train-test splitting using engine numbers to maintain temporal consistency and prevent data leakage. We also addressed outlier handling by replacing detected outliers with rolling median values rather than removing them entirely, thereby preserving the temporal continuity of sensor data. Additionally, we engineered features such as rolling statistics (mean and standard deviation) over a fixed window to capture temporal trends.

The classification models were evaluated using accuracy, precision, recall, and F1-score, with a particular focus on accurately identifying stages 3 and 4, as these represent critical degradation states requiring maintenance intervention. Among the models tested, XGBoost consistently demonstrated the best performance, achieving high accuracy and F1-scores, particularly in identifying critical stages. Logistic Regression, the runner-up, was notably less effective compared to XGBoost. Random Forest and SVM showed moderate performance but were outperformed by both XGBoost and Logistic Regression.

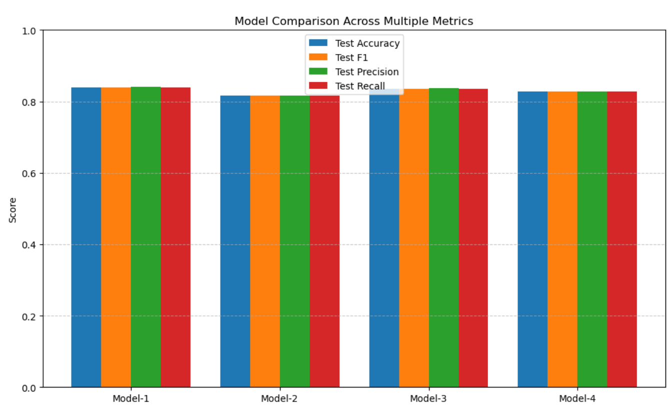

Fig. 3. Comparison of accuracy and F1-score for different classification models (1: Log-Reg, 2: SVM, 3: Random Forest, 4: XGBoost).

Further analysis using ROC-AUC curves demonstrated that XGBoost outperformed Logistic Regression, highlighting its superior capability in distinguishing between degradation stages. As shown in Figure 4, the ROC-AUC curve clearly indicates the enhanced performance of XGBoost compared to the other classifiers.

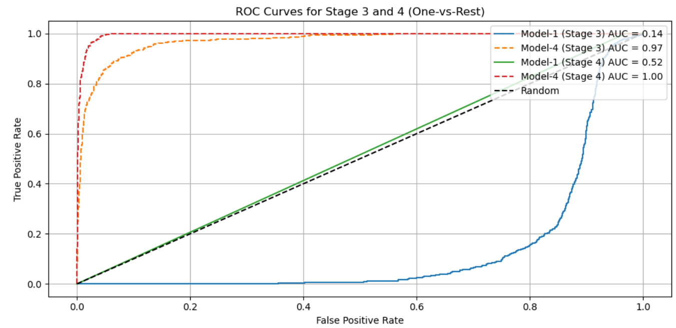

Fig. 4. ROC-AUC curves for Logistic Regression (Model-1) and XGBoost (Model-4).

Interestingly, the application of SMOTE had little to no impact on model performance. The recall increased by 0.001, indicating that the models were inherently robust to class imbalance, or that the SMOTE technique did not significantly augment the minority class data in this context. This suggests that XGBoost's internal handling of class imbalance was sufficient without synthetic data augmentation.

These results demonstrate that the proposed classification pipeline effectively distinguishes degradation stages, with XGBoost providing the most reliable performance. The model's ability to accurately predict critical stages supports its integration into the risk assessment framework. A detailed numerical comparison between the models will be presented in the subsequent analysis table.

C. Regression for Time-to-Next-Stage Prediction

D. Risk Score Computation and Decision Logic