Project Report

CS2202 - Machine Learning

Problem statement

"Hybrid Predictive Maintenance using Enhanced CMAPSS NASA Dataset"

Team Members

(in chronological order)

Atharv Ashish Garg - SE23UCSE223
Janaki Reddy Beeram - SE23UECM025
Pranav Dubey - SE23UARI096
Shrivatsh Kuppu Subramaniam - SE23UARI115

(Intentionally Left Blank)

Problem Statement: Hybrid Predictive Maintenance using Enhanced CMAPSS NASA Dataset.

Introduction

In the aerospace industry, due to the high expenditure and safety implications associated with unplanned maintenance, the proficiency to monitor engine health and predict failures is particularly crucial [1]. Predictive maintenance aims to anticipate equipment failures before they occur, enabling proactive maintenance scheduling, optimized resource utilization, and reduced operational downtime.

Conventional predictive maintenance is based, to a significant extent, on regression-based models that estimate equipment Remaining Useful Life (RUL) of machinery based on sensor measurements [2]. While effective to a degree, these models often treat degradation as a monolithic process, failing to capture the nuanced progression of equipment health through multiple stages of wear and deterioration, usually ignoring multi-stage degradation development and respective risk factors [3]. In addition, a majority of the solutions offered nowadays possess limited interpretability and rarely quantify the uncertainty or risk associated with their predictions [4].

This paper presents a hybrid machine learning approach that bridges unsupervised clustering, supervised classification, and ensemble-based regression models to identify the health state of aircraft engines and construct an improved predictive system for maintenance. By representing the degradation process in terms of five distinct stages, the system not only predicts RUL but also tracks the current degradation stage of the engine. Classification of such stages enables accurate regression in more homogeneous health intervals, thereby making predictions robust and accurate.

Additionally, we incorporate risk-scoring mechanisms and quantile regression for uncertainty prediction assessment, facilitating risk-informed decision-making [5]. We further enhance the comprehensibility of the system with visual analytics tools to allow operators and maintenance planners to deftly comprehend model results, subsequently acting upon the outputs. This method facilitates a significant step toward more intelligent, transparent, and improved maintenance decision-supporting systems for safety-critical applications, such as aviation.

Related Work

Predictive maintenance with the aid of machine learning has come to the foreground in industrial prognosis, particularly owing to the prevalence of benchmarking datasets such as NASA's COMPASS [2][11]. Several studies have utilized this dataset to predict the Remaining Useful Life (RUL) of aircraft engines, employing all sorts of methodologies from simple regression models to high-end deep networks.

Early approaches mainly employed regression-based models like linear regression, support vector regression (SVR), and random forests to estimate RUL directly from unprocessed sensor readings [2][11]. Although these have the advantage of interpretability and ease of use, they tend to underfit capturing the intricate temporal dependencies and degradation characteristics of turbofan engines.

With the advent of deep learning, more advanced methods have been advanced. Long Short-Term Memory (LSTM) networks and convolutional neural networks (CNNs) have been extensively used to capture the sequential nature of sensor data [3]. These models showed better accuracy in RUL estimation; however, they tended to lack interpretability and needed huge computational resources and hyperparameter tuning. Additionally, most of these models considered engine degradation as a uniform process without accounting for the variability in degradation patterns across different engines and operating conditions [4].

A few authors have investigated using clustering-based techniques to find analogous degradation paths or operational conditions [5]. While useful to divide the data into segments, the techniques were hardly combined with ensuing classification or regression models into one single pipeline, which detracted from their pragmatic value.

Classification methods have also been applied to forecast discrete health states or failure phases to yield a rough estimation of engine conditions [6]. Nevertheless, classification alone does not provide the level of granularity necessary for accurate RUL forecasting or dynamic maintenance planning.

Recent studies have explored hybrid approaches that combine multiple machine-learning techniques to address these limitations. For instance, ensemble methods, such as gradient-boosted trees combined with neural networks, have been applied to improve RUL prediction accuracy on the CMAPSS dataset [7]. These methods leverage the strengths of different algorithms to model complex degradation patterns. Additionally, some researchers have proposed multi-stage degradation models that segment the degradation process into distinct phases, enabling more precise predictions [8]. However, these approaches often lack integration with real-time classification or risk assessment, limiting their applicability in operational settings.

Other works have focused on uncertainty quantification in RUL predictions, using techniques like quantile regression to provide confidence intervals [9]. Such methods enhance decision-making by offering probabilistic estimates, but they are rarely combined with stage-based classification or risk scoring. Furthermore, while visual analytics tools have been proposed to improve model interpretability [10], their integration into comprehensive predictive maintenance pipelines remains limited.

In contrast, our suggested methodology integrates unsupervised clustering, supervised classification, ensemble-based regression, and risk scoring into a cohesive pipeline. By modeling degradation as a multi-stage process and incorporating real-time health state prediction and uncertainty quantification, our approach addresses the gaps in interpretability, granularity, and practical deployment identified in prior work.

Methodology

This proposed hybrid pipeline integrates unsupervised learning, supervised classification, and ensemble-based regression to model engine degradation in a structured and interpretable manner. It consists of four sequential phases:

- Phase 1: Clustering to identify Natural Degradation Stages
- Phase 2: Classification for Real-time Health State Prediction
- Phase 3: Regression for Time-to-Next-Stage Prediction
- Phase 4: A risk-scoring mechanism that combines stage severity with temporal proximity to failure for maintenance prioritization

This multi-phase approach enables fine-grained RUL estimation, interpretable health monitoring, and proactive, risk-informed maintenance decision-making.

Phase 1: Clustering to Identify Natural Degradation Stages

To prevent arbitrary thresholds for Remaining Useful Life (RUL) segmentation, this phase employs unsupervised clustering to uncover Natural Degradation patterns present in engine operational data. This validates the modeling of degradation as a progressive, multi-stage process rather than a binary or continuous scale, supporting more interpretable and realistic failure modeling [12].

Clustering Technique: Hierarchical Agglomerative Clustering (HAC)

Using Ward's linkage method, the clustering process is based on HAC (Hierarchical Agglomerative Clustering), a capsized clustering algorithm that initially handles each data point as its cluster and iteratively unifies the most similar clusters relying on a linkage criterion. Ward's linkage minimizes the total within-cluster variance, which ensures that clusters formed are compact and well-separated [13].

This approach does not require specifying the precedent number of clusters (unlike K-Means), and is thereby suitable for degradation modeling; further, the resulting dendrogram can direct the optimal number of degradation stages based on the data structure.

Feature Weighting

To ensure the clustering algorithm captures the most relevant sensor signals, variance-based feature weighting was applied. Features exhibiting higher variance across engine cycles tend to capture meaningful degradation patterns. Proportionally weighting the features, the clustering process is biased towards dimensions most indicative of health deterioration [14].

Dimensionality Reduction for Visualisation

Principal Component Analysis (PCA) and t-distributed Stochastic Neighbour Embedding (t-SNE) were applied to lower dimensions and visualize the high dimensionality of the sensor data. While t-SNE is a suitable nonlinear tool for visualizing complex, high-dimensional clustering patterns in two or three dimensions, PCA produces a linear projection preserving the maximum variance [15, 16]. The pre-defined cluster boundaries were supported by visualizations verifying the existence of obvious groups among the data points.

Defined Degradation Stages

Cluster analysis and visualization demonstrated five distinct stages of degradation:

- Stage 0: Normal - Engine operating within nominal limits.
- Stage 1: Slightly degraded - minor deviations from normal, early wear indicators.
- Stage 2: Moderately Degraded - Explicit patterns of degradation begin to appear.
- Stage 3: Critical - Operation is nearly failing at stipulated levels.
- Stage 4: Failure - The system is either malfunctioning or operates with unacceptable hazards.

These phases function as categorical labels for the subsequent classification phase and offer comprehensible checkpoints in the degradation path. More accurately than single-threshold RUL regression, this multi-stage formulation demonstrates the incremental nature of engine deterioration, thereby facilitating both enhanced prediction and practical insight.

Phase 2: Classification of Degradation Stages

With the cluster-defined labels created in the previous step, a supervised classification pipeline was built to facilitate real-time prediction of the degradation stage of an engine unit. This step fills the gap between unsupervised pattern discovery and real-world deployment by making it possible for the system to deduce the present health state of an engine from raw sensor readings. The characteristics applied to classification were extracted from engine cycle information, such as direct sensor measurements, temporal context information, and moving statistical aggregates computed to reflect both snapshot values as well as recent trends.

A Random Forest classifier was used because of its ability to capture nonlinear relationships, its robustness against noisy data, and its intrinsic interpretability [16]. Being an ensemble of decision trees, the Random Forest also offers important insight, further promoting transparency, a critical requirement in safety-critical applications like aviation. The model was trained on the cluster-allocated stage labels so that it could learn the patterns of each degradation phase, from normal operation to impending failure.

In consideration of the inherent imbalance of the data, particularly the underrepresentation of samples in Stage 3 (critical) and Stage 4 (failure), training incorporated SMOTE (Synthetic Minority Over-sampling Technique). SMOTE artificially creates new instances of minority classes by interpolating between instances, thus avoiding bias toward majority classes and improving the detection of high-risk states by the model.

The classification model was evaluated using a combination of standard metrics:

- Accuracy – Overall proportion of correct predictions
- F1-score – Harmonic mean of precision and recall, emphasizing class-wise balance
- Confusion Matrix – Detailed view of misclassification patterns between stages
- ROC Curves – Evaluated per class to understand true positive rate vs. false positive rate at various thresholds

These metrics confirmed an extensive knowledge regarding the behavior of the model, specifically early detection of degradation.

After deployment, this classifier can run in real time, continuously evaluating engine health as new data arrives. Its predictions not only give direct insight into the current stage of degradation but also are used as input for the subsequent stages of the pipeline estimation of time-to-transition and risk scoring hence playing a key role in facilitating intelligent, stage-aware maintenance strategies.

Phase 3: Regression for Time-to-Next-Stage Prediction

To increase the level of detail in predictive maintenance and go beyond coarse stage classification, the subsequent step of the pipeline aims at estimating the number of engine cycles left until the transition to the next degradation stage. Instead of predicting the time to ultimate failure, this localized regression method gives more timely, actionable information. The underlying concept is that knowing how long an engine can degrade from where it currently is allows more proactive interventions and smarter resource distribution within a maintenance program.

An ensemble of regression algorithms was utilized to carry out this task, taking advantage of the complementary strengths of various algorithms [17]. Random Forest Regressor was incorporated due to its capability to model nonlinear relationships and prevent overfitting by ensemble averaging. Ridge Regression, being a linear model with L2 regularization, was used to manage multicollinearity among features and provide a simple, stable baseline. Histogram-based Gradient Boosting, with its reputation for efficiency and accuracy over tabular data, provided robust performance on structured sensor features. Notably, Quantile Regression was also incorporated into the ensemble to produce prediction intervals, providing uncertainty estimates—a key requirement when enabling risk-aware decision-making [18].

The group provides both point predictions (predicted cycles to the next phase) and interval estimates that measure prediction confidence. This double output not only enables maintenance planners to respond to a central prediction but also to estimate the probability of premature or delayed transitions, which makes for more conservative or more aggressive interventions based on the operating situation.

To evaluate model performance, several metrics of evaluation were employed. RMSE and MAE were computed to estimate the precision of point estimates, and quantile interval accuracy was employed to gauge the ability of the forecasted confidence intervals to capture true results. Collectively, these measures helped ensure both precision and reliability were captured in evaluating regression quality.

This stage fortifies the end-to-end predictive maintenance pipeline by bridging the gap from stage detection to ultimate failure prediction. By approximating how long there is before an engine hits the next critical condition, the system provides granular control over maintenance choices, further aligning predictive analytics with operational requirements.

Phase 4: Risk Score Computation

Based on the stage classification and estimation of time-to-transition, the last stage of the pipeline then adds a risk scoring system to decide upon priorities for maintenance based on both current engine condition and the expected rate of degradation. While classification gives the health stage and regression predicts the cycles until degradation, the risk score then unifies these results into one actionable number representing both severity and urgency.

The risk score is defined using a simple yet effective formula: Risk Score = (Stage Weight) / (Estimated Cycles + 1) [19]. Here, Stage Weights are assigned heuristically to reflect the relative criticality of each degradation level, with higher weights for more severe stages (e.g., Stage 0 = 0, Stage 1 = 2, …, Stage 4 = 10). The denominator ensures that engines with fewer cycles remaining before transitioning to the next stage are flagged with higher risk, thus capturing time sensitivity.

To compare risk interpretations on the same basis between units, raw scores are normalized onto the $[0, 1]$ range with those near 1 representing engines both at a high degradation phase and nearing the subsequent critical point soon. This makes consistent comparisons as well as aggregation into dashboards or decision support systems possible.

The resulting risk metric is used as a scheduling prioritization tool for maintenance. Engines with high-risk scores can be selected for early inspection or preemptive replacement, whereas those with lower scores may be allowed to operate with normal monitoring. This phase combines degradation severity with temporal proximity to failure and converts analytical predictions into tangible, actionable knowledge for maintenance personnel. Risk-based maintenance strategies have been developed to optimize maintenance planning [20].

Dataset and preprocessing

The CMAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset, developed by NASA, is widely used for predictive maintenance research due to its realistic simulation of turbofan engine degradation [2]. It models the complete run-to-failure behavior of multiple engine units under varying operational conditions and fault patterns. Each dataset variant (such as FD001, FD002, etc.) presents different combinations of operating settings and fault modes, making CMAPSS well-suited for generalizable and robust model development. The core data consists of multivariate time-series sensor readings recorded over engine cycles, capturing complex temporal dynamics leading up to failure.

To prepare the raw data for modeling, a multi-step preprocessing pipeline was implemented. First, data sanitization was performed to remove anomalous readings and apply smoothing techniques [15], ensuring more consistent signal quality. This was followed by feature engineering, where a set of informative descriptors such as rolling mean, variance, and slope were derived to capture local trends and statistical behavior over time. These features helped expose latent degradation signals that are often subtle or obscured in raw sensor data.

Following feature extraction, all sensor readings and engineered features were normalized and scaled to standardize value ranges across different units and operating conditions. This step was crucial for improving model training stability and ensuring fair weighting of input variables. Lastly, a data injection logic was designed to simulate a streaming environment, enabling the system to process data cycle-by-cycle as it would in a real-world deployment. This setup allowed for the development and testing of real-time predictive models, closely mimicking operational maintenance scenarios.

Results and Observations

The authors had access to only very sparse resources, in our case it was a MacBook M2 with 8GB RAM, for training and evaluation purposes. Due to this, the following results can be reproduced with better accuracy and precision for the reasons mentioned under each section.

The reason for having such low accuracy in the classification phase is because of the fact that there was less hyper parameter searching time. This can be solved using proper grid search or random search methods and doing more thorough tuning of hyperparameters. Thorough tuning of hyperparameters will in turn affect the regression and risk's results.

Clustering

The authors first made five temporary stages based on the timeline, i.e. Let's suppose that engine unit 1 is running for 100 time cycles, then those 100 cycles are divided as such: the first 20 cycles will be considered under Stage 0, the next 20 cycles will be considered under Stage 1, and so on until all the cycles are complete for an engine unit. This arbitrary pseudo-clustering provides us with a benchmark to compare our formed clusters. Furthermore, the proposed clustering model can give us an approximate accuracy of 86%. Apart from these, the NASA CMAPSS dataset [1] claims that the engines will tend to be in the earlier stages for a longer period when compared to end-of-life stages. This can be well represented and inferred from the following representations:

Dataset	Accuracy	Precision	Recall	F1-Score	Gini Index	Entropy	Purity	Inverse Purity	F-Measure
FD001	0.8177135229	0.838282229	0.8177135229	0.8125335961	0.2437653948	0.5439548299	0.8302309593	0.8322053058	0.8311726308
FD002	0.8082512956	0.8295572323	0.8082512956	0.8033040021	0.2551116775	0.5614436107	0.8173844691	0.8219310625	0.819582334
FD003	0.8213964261	0.8418353859	0.8213964261	0.8165586896	0.2414763618	0.5354707623	0.8282714654	0.8325732832	0.8303472929
FD004	0.8122785576	0.8340362729	0.8122785576	0.8079030355	0.2521362851	0.5568958704	0.8193773778	0.82244392	0.820856591

Table 1, represents quality and performance metrics for the clustering phase

These pseudo-classes are only meant to be used as a reference point to understand whether the data is being "mis-clustered" or is an outlier within the datasets.

Classification

As mentioned about the sparse behavior of the evaluation resources, proper grid search or random search techniques could not be applied for strong hyperparameter tuning. This resulted in a low accuracy, precision and F1 score result. Although satisfactory, it is strongly suggested by the authors to use optimised evaluation resources to get stronger results with replicating reproducibility.

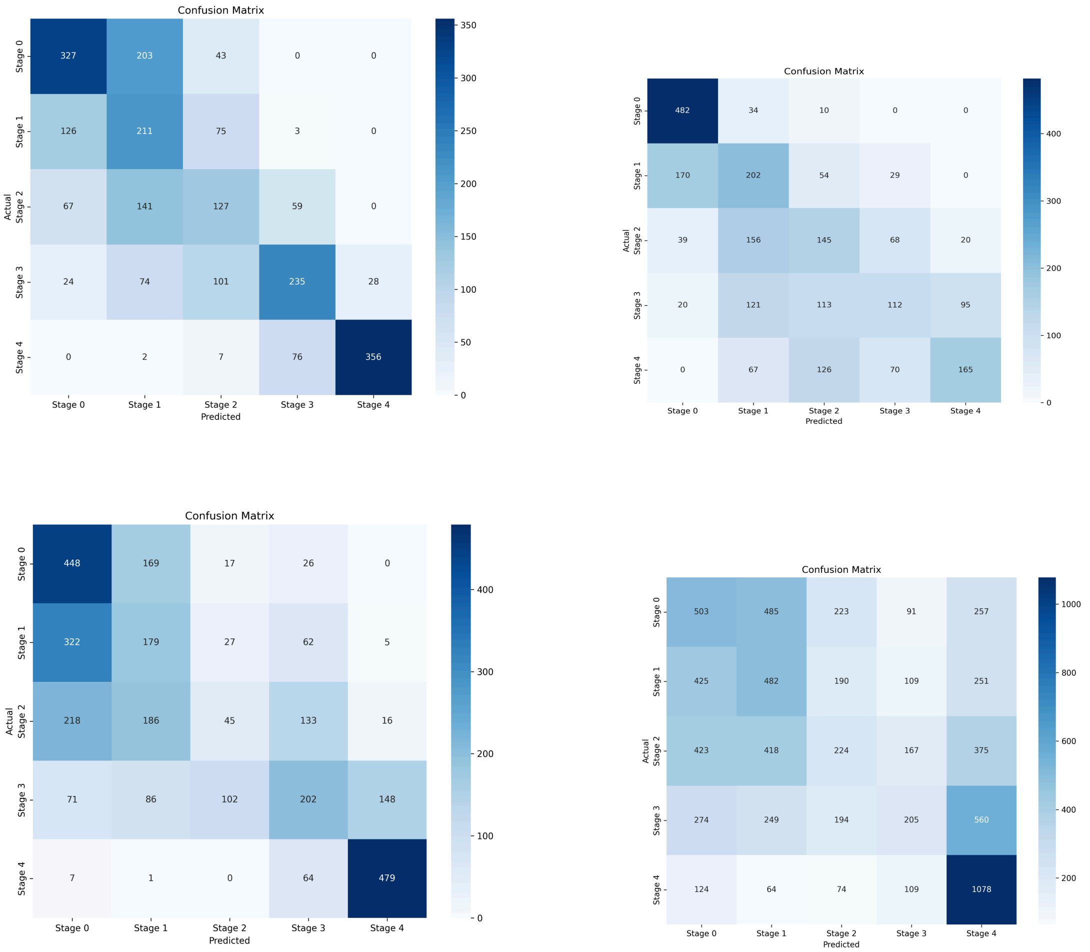

Figure 1: Confusion matrices for classification phase, (i) FD001, (ii) FD002, (iii) FD003, (iv) FD004

In all of the datasets, as seen in *Figure 1*, Stage 0 and Stage 4 are getting the most true positive hits when compared to other stages. This may purely be due to the biased stump forming at time cycle edges. Moreover ,as seen in *Figure 2,* the feature importance as per each dataset shows the weightage for all important rolling statistics which were used to form the weighted stumps in the case of random forest classification.

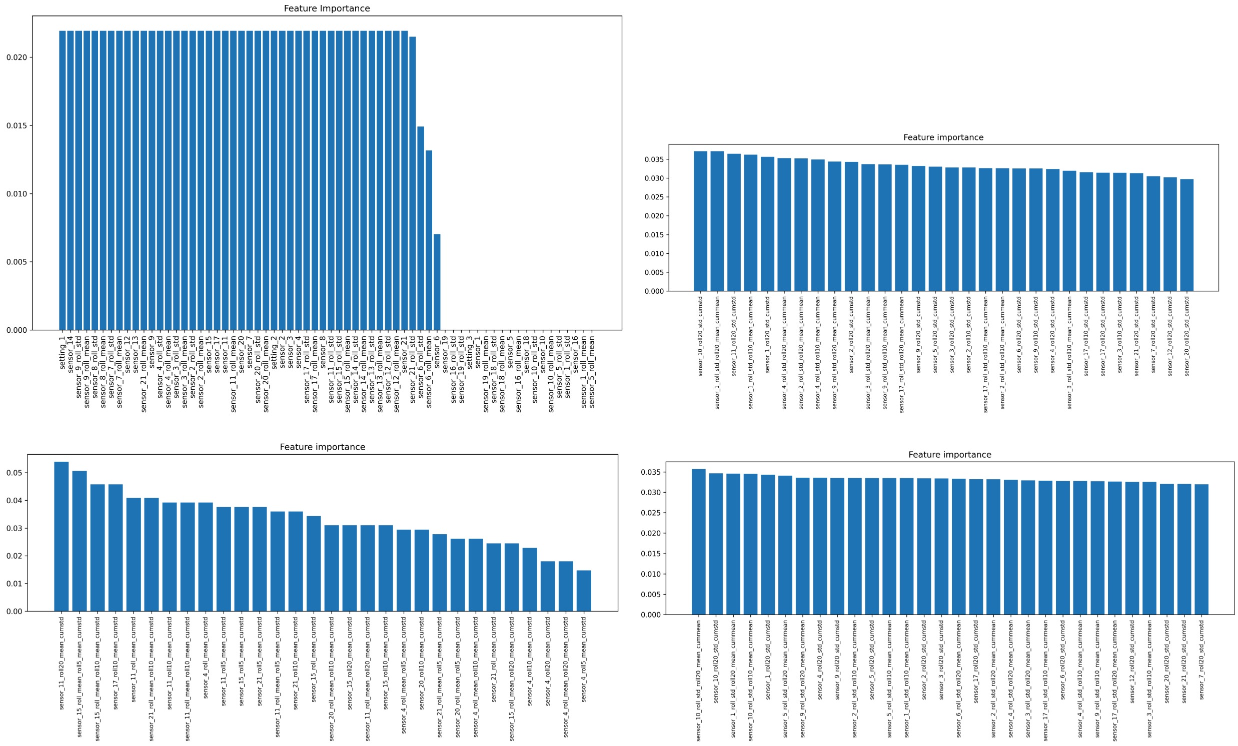

Figure 2: Feature important extraction during the classification phase. (i) FD001, (ii) FD002, (iii) FD003, (iv) FD004

Regression

The error rates from the previous phases carry forward onto the regression stage and adversely affect the predictions of the time-to-next stage (TTNS). Although the quality metrics and performance seem valid enough and in line with the claims made about the dataset in [1], considering a re-evaluated classification stage model will result in more accurate and precise results. The RMSE and MAE calculated for the regression stage seem too high enough to validate the dataset predictions. *Figure 3,* also supports the claim that the authors of this paper make about the predictions. A simple model comparison can also be performed to provide a benchmarked test with the current regression pipeline (refer to *Figure 4* for more)

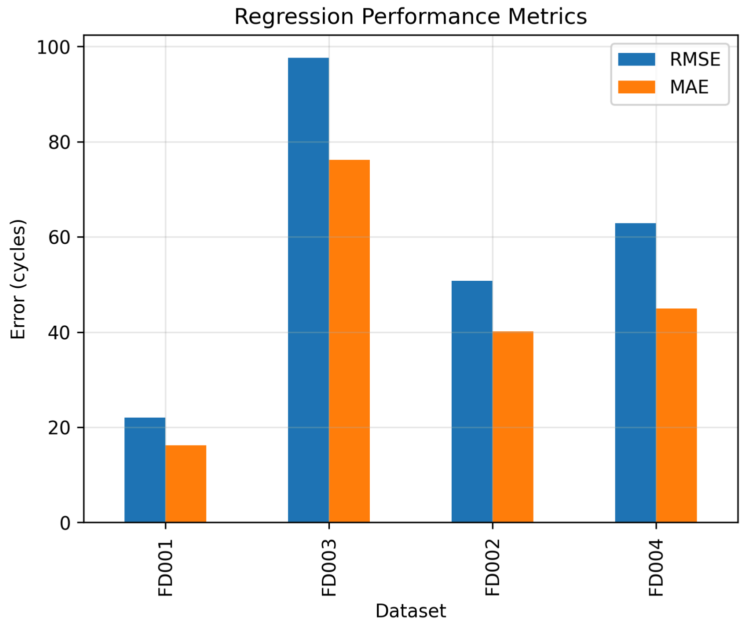

Figure 3: RMSE and MAE values for the regression phase.

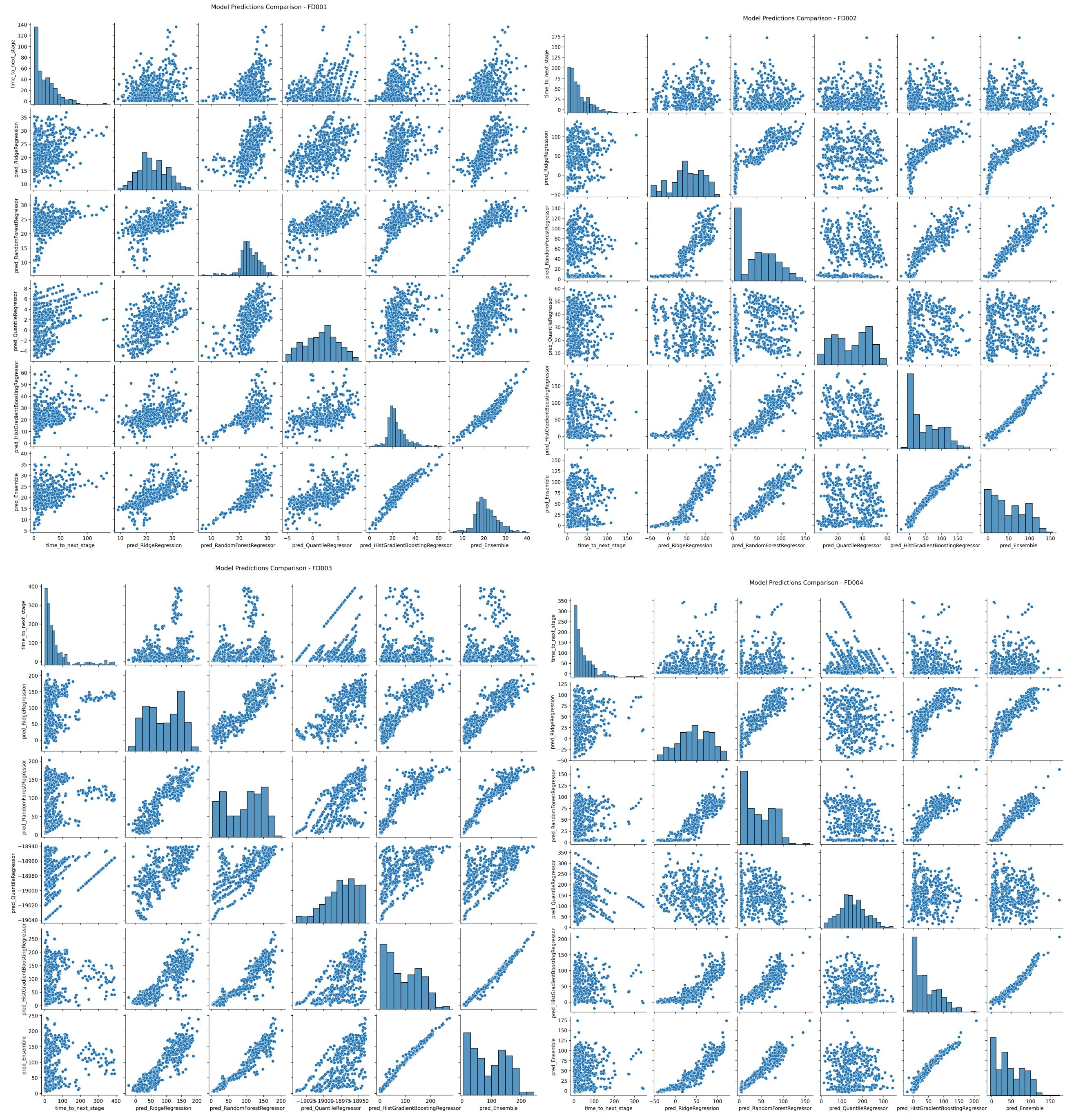

Figure 4: Model comparison analysis for FD001, FD002, FD003, and FD004 datasets with (i) TTNS (ii) Ridge regression (iii) Random Forest Regressor (iv) Quantile Regressor (v) Histogram Base GB (vi) Model Ensemble Prediction

<u>*Risk Assessment*</u>

As the completion of the pipeline, the risk assessment scored probabilistic values for the failure of each engine based on time. As any mechanical system would incline towards failure as time progresses, the predicted assessments tend to fall in place with these mechanics. As shown in *Figure 5*, the risk of failure of an engine unit increases as the time spent under those conditions increases.

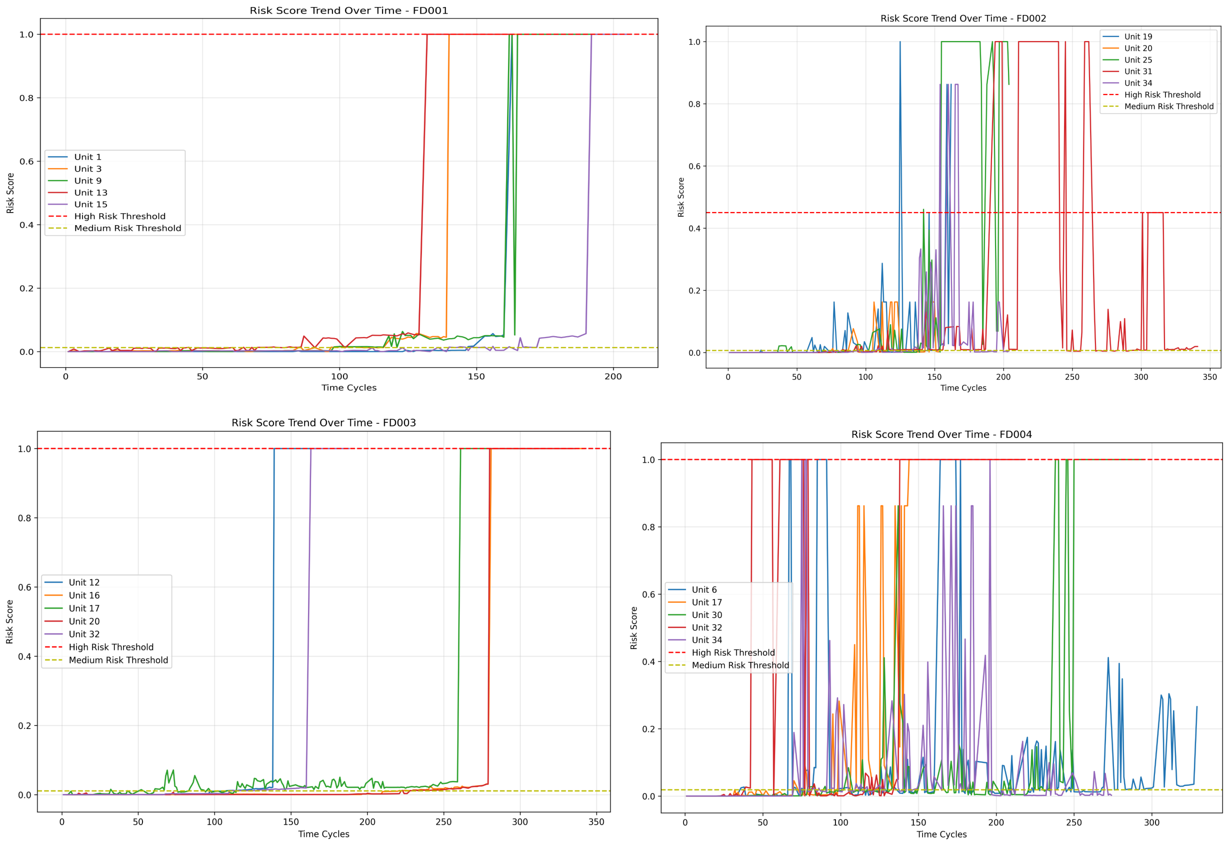

Figure 5: Risk score over time for all four datasets.

Finally, comparing our observations with the NASA CMAPSS dataset [1], our risk predictions and probability predictions seem to fall in line, but these will only be valid for FD001 and FD003, and not for FD002 and FD004 due to the stable nature of FD001 and FD003 about the risk incurred over time. The latter datasets are unstable and require much more accurate and precise hyperparameter tuning to have a near-perfect implementation of this lightweight and computationally optimized pipeline.

Future developments and improvements

This research presents a number of fundamental innovations that move the art of predictive maintenance based on the NASA CMAPSS dataset forward. For one, it is the first to utilize Hierarchical Agglomerative Clustering (HAC) to identify natural degradation phases in CMAPSS engine trajectories. Unlike existing research that depends on arbitrary RUL values or binary failure indicators, HAC usage allows for data-driven identification of multi-stage degradation progression, resulting in more interpretable and meaningful representations of the health state.

Second, the pipeline uses quantile regression to estimate uncertainty, a new integration for turbofan engine maintenance. While the majority of conventional methods only provide point estimates of Remaining Useful Life (RUL), the proposed framework offers predictive intervals so that operators can evaluate both central predictions and the associated variability, essential for safety-critical systems.

Third, the proposed architecture is one of the first to show a complete integrated clustering–classification–regression pipeline designed specifically for CMAPSS. Such systematic design increases model robustness because each stage is built on top of the strength of the last one: unsupervised learning is used to inform supervised classification, which is then used to inform fine-grained time-to-failure regression.

Lastly, the system provides a risk-aware maintenance rationale by merging degradation severity and proximity to failure in time into a normalized risk value. This value, together with interactive visualizations and confidence intervals, provides an actionable and easy-to-understand interface for engineers. The integration of real-time dashboards and interpretability tools makes the framework not only technically sound but also realistically deployable in actual maintenance settings. The integration of temporal deep learning models like LSTM and GRU to more effectively capture sequential dependencies in sensor data will be investigated in the future. The framework will also be extended and tested on real industrial datasets to assess its robustness, flexibility, and scalability beyond the CMAPSS benchmark. These directions will strive to advance predictive maintenance towards completely autonomous, data-driven maintenance systems in high-risk industries. These contributions are an important step toward robust, explainable, and operationally meaningful predictive maintenance systems.

Conclusion

The Hybrid Predictive Maintenance paradigm under consideration is a considerable improvement from the conventional RUL-based models. Through the incorporation of unsupervised clustering for degradation stage identification, supervised classification for real-time health estimation, and ensemble regression for time-to-next-stage prediction, the system provides an understandable and holistic solution for engine health monitoring. The inclusion of a normalized risk-scoring component fortifies its applicability even further through the possibility of incorporating proactive, risk-aware scheduling of maintenance.

Aside from predictive precision, the use of visual analytics and interactive dashboards improves interpretability and user experience, essential for deployment in operational settings. They support informed decision-making and instill trust in automated predictions by maintenance staff.

Nonetheless, there are limitations to the model. While it operates well in stage identification and degradation prediction, the model is dependent on historical information and predetermined forms of degradation, thus failing to capture some of the complexity involved in real-world engine behaviors, especially when faults are rare or unexpected.

Moreover, the system's reliance on feature engineering and stage weights defined by hand can restrict its applicability to other machinery types or less structured data. Also, while the model does present uncertainty bounds through quantile regression, it still does not account for possible correlations between sensor faults or environmental variables that might affect engine performance.

Hence, the pipeline can be reinforced with the proposed schemas but do end up providing reproducible results. To access the github contribution repository, you may find the source code here: https://github.com/MarkVI2/G_AI096_AI115_ECM025_CSE223. The pipeline further requires optimal high performance computing tasks but tends to perform with a strong confidence to be used in a real life scenario.

References

[1] Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. IEEE International Conference on Prognostics and Health Management, 1-9.

[2] Babu, G. S., Zhao, P., & Li, X. L. (2016). Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. International Conference on Database Systems for Advanced Applications, 214-228.

[3] Li, X., Ding, Q., & Sun, J. Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety, 172, 1-11.

[4] Ordóñez, C., Lasheras, F. S., Roca-Pardiñas, J., & de Cos Juez, F. J. (2019). A hybrid ARIMA–SVR approach for Remaining Useful Life prediction of turbofan engines. Measurement, 134, 178-187.

[5] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics, 11(3), 812-820.

[6] Yan, W., & Yu, L. (2019). On accurate and reliable anomaly detection for gas turbine combustors by using a deep learning approach. ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition, GT2019-90738.

[7] Singh, S.K., Kumar, S., & Dwivedi, J.P. (2019). A novel soft computing method for engine RUL prediction. Multimedia Tools and Applications, 78, 4065–4087.

[8] Mosallam, A., Medjaher, K., & Zerhouni, N. (2016). Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. Journal of Intelligent Manufacturing, 27(5), 1037-1048.

[9] Li, X., et al. (2022). A Distributional Perspective on Remaining Useful Life Prediction With Deep Learning and Quantile Regression. IEEE Access, 10, 9887803.

[10] Lou, Y., Caruana, R., Gehrke, J., & Hooker, G. (2013). Accurate intelligible models with pairwise interactions. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 623-631.

[11] Mosallam, A., Medjaher, K., & Zerhouni, N. (2016). Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. Journal of Intelligent Manufacturing, 27(5), 1037-1048.

[12] Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86-97.

[13] Modha, D. S., & Spangler, W. S. (2003). Feature weighting in k-means clustering. Machine Learning, 52(3), 217-237.

[14] Jolliffe, I. T. (2002). Principal component analysis. Springer Series in Statistics.

[15] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11), 2579-2605.

[16] Kizito, R., Scruggs, P., Li, X., Kress, R., Devinney, M., & Berg, T. (2018). The Application of Random Forest to Predictive Maintenance. Proceedings of the 2018 IISE Annual Conference.

[17] Singh, S.K., Kumar, S., & Dwivedi, J.P. (2019). A novel soft computing method for engine RUL prediction. Multimedia Tools and Applications, 78, 4065–4087.

[18] Li, X., et al. (2022). A Distributional Perspective on Remaining Useful Life Prediction With Deep Learning and Quantile Regression. IEEE Access, 10, 9887803.

[19] Liebman, E. (2024). Pattern-Based Time-Series Risk Scoring for Anomaly Detection and Alert Filtering -- A Predictive Maintenance Case Study. arXiv preprint arXiv:2405.17488.

[20] Khan, F. I., & Haddara, M. M. (2003). Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning. Journal of Loss Prevention in the Process Industries, 16(6), 561-573.