Enabling Accurate and Timely Prognostics for Aircraft Turbofan Engines

Philippa Scroggins and Sidi Lu
Department of Computer Science, William & Mary, Williamsburg, VA 23185, USA
prscroggins@wm.edu, sidi@wm.edu

***Abstract*—The aviation industry relies on scheduled maintenance performed on aircraft engines, which ensures safety but incurs significant costs during routine inspections. Traditional preventative maintenance may introduce issues during inspections, leading to delays and cancellations. Predictive maintenance, powered by edge computing, offers a more efficient solution to predict engine failures, optimize schedules, and enhance safety.**

This paper explores the development and evaluation of a predictive model based on the Gradient-Boosting Regression Tree (GBRT) algorithm for turbofan engine prognostics. Our study uses a synthetic dataset to evaluate the performance of the proposed model through various external conditions and internal configurations. Through this analysis, we compare our model's performance to existing solutions and propose a new benchmarking metric, Margin-Adjusted Reliability Score (MARS), to better assess the applicability and effectiveness of predictive maintenance models in real-world scenarios.

***Index Terms*—Aircraft, turbofan engine, predictive maintenance, edge computing, fault forecasting, time series data**

I. Introduction

Regularly scheduled engine maintenance is a standard practice in the aviation industry due to the high cost of failure and low error tolerance. However, it is expensive as it occurs regardless of the engine's actual condition. Preventative maintenance, while reducing failure risks, can introduce new mechanical issues during inspections, leading to costly flight delays, diversions, cancellations, and accidents. For instance, a single flight cancellation can cost an airline about $140K, while each hour of delay costs around $17K [1].

Advancements in edge computing and the Industrial Internet of Things (IIoT) enable a shift from preventative to predictive maintenance, enhancing uptime, cost efficiency, and safety [2]. Predictive maintenance starts with continuous engine monitoring using sensors [3] which collect data on parameters like temperature and vibration. The data, combined with off-board operational information, is analyzed by machine learning models to detect anomalies and predict failures [4]. The resulting insights guide maintenance scheduling to ensure safe and continuous aircraft operation. For instance, predictive maintenance has helped Boeing 787 reduce flight delays and cancellations by 30% and unscheduled removals by 20% [5].

However, engine prognostics remains challenging due to data ownership disputes between engine manufacturers and commercial consumers, stringent regulatory requirements by the Federal Aviation Administration (FAA), and the technical limitations of predictive models [5]. Crucially, effective models need data from failed components, but obtaining such data requires allowing engines to fail [6]. Even models with high accuracy in controlled settings may struggle in real-world applications due to these constraints [7].

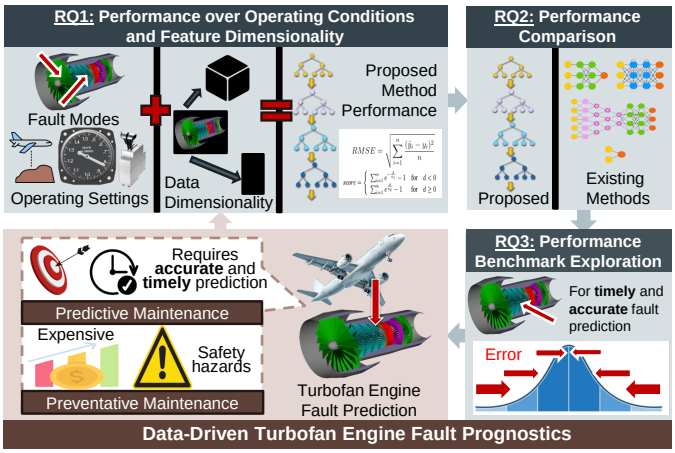


Fig. 1. A succinct overview of our structured methodology for developing a prediction framework to inform turbofan engine fault prognostics. It includes three main research questions (RQs) related to performance behavior (top-left), performance comparisons (top-right), and performance benchmark exploration (bottom-right). RQ1 examines how the performance of our proposed method varies relative to the different operating conditions in the data and under different feature engineering conditions. RQ2 compares the performance of our prediction framework to other methods used for this task. RQ3 examines current performance benchmarks and explores a new benchmarking framework that better addresses the needs of the problem.

To address the scarcity of real-world failure data, researchers rely on simulations to generate synthetic data for developing and testing prognostic models. This study uses a novel dataset from the National Aeronautics and Space Administration's (NASA) Ames Prognostics Center of Excellence [8]–[10], which simulates run-to-failure trajectories for a fleet of aircraft engines under realistic flight conditions [11]. This dataset is a valuable resource for studying engine degradation and failure prediction for developing predictive models.

The above dataset has inspired numerous publications proposing various turbofan engine prognostics algorithms, many using neural networks. However, assessing progress is challenging due to the lack of uniform benchmarking metrics, inconsistent sub-datasets, and varying evaluation environments across studies [12]. A standardized performance framework for benchmarking these approaches is still needed.

Research Questions. Our research is guided by three primary questions: (i) How does the performance of the proposed method vary under different feature engineering conditions

and operating conditions within the sub-datasets (**RQ1**)? (ii) How does the proposed algorithm compare to other existing solutions (**RQ2**)? And finally, (iii) how can we assess the applicability of these solutions for turbofan engine prognostics and predictive maintenance (**RQ3**)?

Contributions. In this study, we develop a method using the Gradient-Boosting Regression Tree (GBRT) algorithm for turbofan engine prognostics, analyze its performance, compare it with existing algorithms, and use the insights gained to inform the creation of a new benchmarking framework that evaluates suitability based on performance accuracy and fault detection timeliness. Our specific contributions are as follows:

- We propose a predictive model for turbofan engine prognostics that utilizes the GBRT algorithm to estimate the Remaining Useful Life (RUL) of aircraft turbofan engines (Section IV).
- For **RQ1**, we test the prediction accuracy and timeliness of our method across different feature dimensionalities and loss functions to highlight how to achieve the best prediction accuracy and timeliness under different operating conditions and fault scenarios (Section V-A).
- For **RQ2**, we compare the prediction accuracy and timeliness of our method with other benchmark solutions, including ConvGAT [13], FCDAE-CNN-LSTM [14], Support Vector Machine (SVM) [15], Multi-Layer Perceptron (MLP) [15], Deep Belief Network (DBN) [15], Long Short-Term Memory (LSTM) [16], and Deep Convolutional Neural Network (DCNN) [17] (Section V-B).
- For **RQ3**, we develop an evaluation framework that captures the real-world applicability and effectiveness of future solutions (Section V-C).

The rest of this paper is organized as follows: Sec. II reviews related work. The experiment datasets and hardware configuration are detailed in Sec. III. Extensive experimental results are shown in Sec. IV. We finally present our detailed discussions in Sec. V, and Sec. VI concludes the entire paper.

II. Related Work

The field of IIoT engine prognostics is hindered by the lack of common datasets, which limits researchers' ability to compare solutions effectively. To address this, Saxena and Goebel established a prognostics data repository in 2008 [12]. Since then, several prognostics datasets have been published and widely used globally. Five of these datasets were generated using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) tool, which simulates realistic turbofan engine data, allowing for fault injection and varying degrees of degradation to support engine prognostic algorithms [11]. The first C-MAPSS dataset was created for a 2008 PHM Society conference data challenge, and subsequent datasets with varying complexity have since been released and utilized in publications [12]. Here, we discuss three primary categories of prognostic models identified in the literature [12].

(1) **Neural Network-Based Methods.** These methods transform engine trajectory data into a multidimensional feature space, using corresponding RUL values to label feature vectors. Supervised learning is then applied to map feature vectors to RUL. Recent advancements include a multi-dimensional attention mechanism combined with a feature-sequence dimensional convolution network, which captures interactions in feature dimensions and temporal sequences, improving RUL prediction accuracy on datasets like NASA's turbofan engine data and XJTU-SY [18]. Hybrid deep learning models, such as Convolutional Long Short-Term Memory (CNN-LSTM) [19] and FCDAE-CNN-LSTM [14], have also gained popularity.

(2) **Health Index-Based Methods.** These methods map sensor measurements to a health index for each training unit, which is then linked to RUL. Recent advancements focus on detecting system degradation using deep learning models. For example, [20] combines deep belief networks with self-organizing map neural networks to build a health index that captures correlations between multi-component systems, significantly improving RUL prediction.

(3) **Similarity-Based Matching.** These methods create a library of system instances with known failure times. For a test instance, similarity with library instances is evaluated to estimate and aggregate RUL. Recent advancements include integrating autoencoder architectures and failure mode-specific metrics to enhance accuracy. For example, [21] uses a classifier to identify the failure mode and guide RUL prediction.

The Gap in Previous Work. Despite the extensive body of literature on turbofan engine prognostics, significant gaps remain. One major issue is the inconsistency in performance benchmarking which makes it difficult to effectively compare results across different studies. Additionally, there are often differences in the datasets chosen for building these prognostic models. Researchers have advised that future work should focus on establishing standardized performance benchmarking and datasets to align the field's progress [12].

III. Experiment Setup

A. Experiment Tools

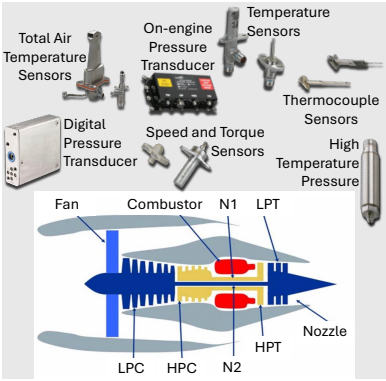
C-MAPSS is a MATLAB and Simulink-based tool designed to simulate a large commercial turbofan engine in the 90K lb thrust class. It operates in diverse scenarios, including altitudes from sea level to 40K ft, Mach numbers from 0 to 0.90, and sea-level temperatures from -60 to 103 degrees Fahrenheit. C-MAPSS also features a power management system for simulating engine operation across various thrust levels under different flight conditions [11].

B. Dataset Description

The studied dataset, the Turbofan Engine Degradation Simulation dataset, generated using C-MAPSS and detailed in [11], includes sensor measurements from multiple engines throughout their usage history. It is designed to support the development of algorithms for predicting engine RUL [11].

This dataset is organized into $n \times 26$ matrices, where n represents the number of engine cycles per trajectory. Each row captures the parameters measured during that cycle: the first column indicates the engine/trajectory number, the second

the cycle number, columns three through five the operational settings, and columns six through twenty-six the engine sensor measurements, as shown in Fig. 2.



Sensor Measurement
Fan Inlet Temperature (◦R)
LPC Outlet Temperature (◦R)
HPC Outlet Temperature (◦R)
LPT Outlet Temperature (◦R)
Fan Inlet Pressure (psia)
Bypass-Duct Pressure (psia)
HPC Outlet Pressure (psia)
Physical Fan Speed (rpm)
Physical Core Speed (rpm)
Engine Pressure Ratio (P50/P2)
HPC Outlet Static Pressure (psia)
Ratio of Fuel Flow to Ps30 (pps/psia)
Corrected Fan Speed (rpm)
Corrected Core Speed (rpm)
Bypass Ratio
Burner Fuel-Air Ratio
Bleed Enthalpy
Required Fan Speed
Required Fan Conversion Speed
High-Pressure Turbines Cool Air Flow
Low-Pressure Turbines Cool Air Flow

Fig. 2. The overview of the simulated engine and sensor measurements. On the bottom left is a high-level diagram of the engine simulated by C-MAPSS, and to the right is the list of sensor measurements included in the C-MAPSS generated dataset. Note that LPC stands for low-pressure compressor, HPC stands for high-pressure compressor, and LPT stands for low-pressure turbine. Different types of engine sensors are pictured on the top left.

Specifically, the full dataset consists of four sub-datasets, each differing in fault modes and operating conditions, as shown in Table I. FD001 sub-dataset simulates High Pressure Compressor (HPC) degradation under one operating condition (sea level). FD002 sub-dataset also simulates HPC degradation but across six operating conditions. FD003 sub-dataset simulates both HPC and fan degradation with one operating condition, while FD004 is the most complex, simulating both fault modes across six operating conditions. Each sub-dataset includes a training set, a testing set, and corresponding ground truth RUL values for the testing data.

	FD001	FD002	FD003	FD004
Training Trajectories	100	260	100	249
Testing Trajectories	100	259	100	248
Operating Conditions	1	6	1	6
Fault Modes	HPC	HPC	HPC & Fan	HPC & Fan

Table I. Dataset details. Number of training and testing trajectories, operating conditions, and fault modes in each of the FD001, FD002, FD003, and FD004 sub-datasets.

The training and testing datasets consist of operational data from multiple engines over their life cycles. In the training set, each engine's data ends when the health index reaches 0. In the testing set, data is truncated before engine failure, aiming to predict the RUL for each engine trajectory.

C. Hardware

The experiment hardware includes a variety of sensors mandated for engine installation by FAA [5], some of which are depicted in Fig. 2. However, C-MAPSS generated data assumes sensors and actuators to be "ideal," meaning they have no dynamics, computational time delays, errors, or biases [22].

IV. Experimental Design and Results

A. Proposed Methodology

While deep learning models dominate this field, exploring GBRT presents an intriguing alternative that could provide new insights into a less-studied class of algorithms.

GBRT is an ensemble learning technique that sequentially adds decision trees, each learning from the residuals of the previous ones to minimize error. The process starts with a base tree and adds trees that correct prior errors, scaled by a learning rate to prevent overfitting. The final model, a weighted combination of all trees, captures complex nonlinear relationships between engine sensor data and RUL values [23].

B. Data Preparation

As shown in Fig. 3, considering the impact of diverse operating conditions on turbofan engine degradation in aircraft, we use K-means clustering [24] to explore sensor measurement variations under different conditions. It is applied to each sub-dataset using three operational parameters, including Altitude (OPS1), Mach Number (OPS2), and Throttle Revolver Angle (OPS3). This process segments the data into distinct groups based on operating parameter combinations. Sub-datasets FD001 and FD003, with a single operating parameter, produce one cluster each, while FD002 and FD004, with three different parameters, yield six clusters each. These combinations are added as re-engineered features in the data.

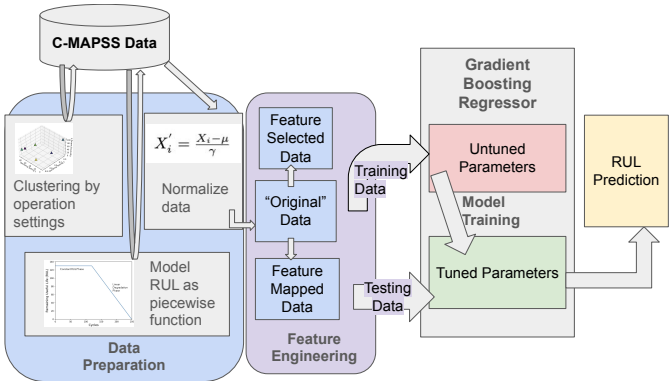


Fig. 3. Prediction framework for the proposed GBRT-based algorithm. Data is first clustered by operational settings to identify operating conditions, which are then incorporated as new features. Ground truth RUL values are modeled for non-target rows using a piecewise function and included as an additional feature. The data is normalized to mitigate noise. Subsequently, the dataset undergoes feature selection and polynomial feature mapping to create two additional versions of the dataset. For each version, training data is used to optimize the Gradient Boosting Regression parameters. After parameter tuning, the model predicts RUL on the test data.

Piecewise RUL Target Function. Ground truth RUL values for the testing set are available only for the final engine cycle, while the training set has no RUL values. To improve prediction accuracy without a physics-based model, we use an approximate degradation model. A piece-wise linear degradation function, as recommended in previous studies [25]–[29], is employed to cap RUL values and capture nonlinear degradation after specific usage thresholds (Fig. 4). This method prevents RUL overestimation and more accurately reflects actual degradation patterns.

Data Normalization. The diversity of operating conditions leads to varying sensor values, making data normalization essential. We normalize the feature data using Eq. 1.

$$X_i' = \frac{X_i - \mu}{\gamma} \tag{1}$$

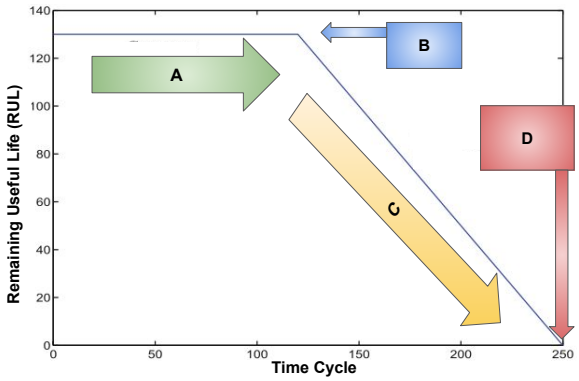


Fig. 4. Piecewise engine degradation function. In segment A, the Remaining Useful Life (RUL) remains constant, indicating that the health index stays stable. At point B, a fault develops, marking the start of degradation. As the fault evolves in segment C, the health index declines and RUL decreases accordingly. Finally, at point D, the health index reaches zero, indicating complete degradation where RUL is zero.

Feature Engineering. We apply Principal Component Analysis (PCA) to create a feature-reduced version of the data, preserving 95% correlation. To explore subtle relationships between features, we also use polynomial feature mapping for feature expansion. Given the variance in the number of sensors across different engines, it is crucial to understand how the algorithm responds to changes in dimensionality.

C. Evaluation Metrics

The variability in aviation industry needs and evolving regulatory standards complicates benchmark establishment, making it difficult to compare results across studies and slowing the field's progress [12]. Despite these challenges, we prioritize three key considerations in selecting our metrics. We adopt the scoring method recommended by the dataset developers in [11], referred to as "Score", defined below.

$$Score = \begin{cases} \sum_{i=1}^{n} e^{-\frac{d}{13}} - 1 & \text{for} \quad d < 0 \\ \sum_{i=1}^{n} e^{\frac{d}{10}} - 1 & \text{for} \quad d \geq 0 \end{cases} \tag{2}$$

where n refers to the number of trajectories and d is the difference between the estimated RUL and the true RUL. "Score" penalizes late failure predictions more heavily than early ones, aligning with the risk-averse nature of the aerospace industry.

However, it has limitations, such as sensitivity to outliers and bias towards algorithms that underestimate RUL. To address this, we explore additional prognostic evaluation methods, guided by the framework in [30], [31], which recommends assessing prediction accuracy, timeframe, maintenance of performance levels relative to RUL, and accuracy at different times. These considerations helped to select from commonly used metrics in related literature for easier comparison.

Root Mean Squared Error (RMSE), shown in Eq. 3, complements "Score" by equally penalizing early and late predictions and measuring the accuracy of both target and step-wise RUL predictions [13]–[17], [25], [32]. We incorporate RMSE to assess performance.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{d^2}{n}} \tag{3}$$

D. Experiment Results

We produce three variations of the GBRT algorithm: GBRT I uses the data without feature engineering, GBRT II uses feature-selected data, and GBRT III uses polynomial feature-mapped data. Each model is tuned on the training data from each sub-dataset to optimize the learning rate, number of boosting stages, maximum depth of estimators, minimum samples required to split a node, minimum samples per leaf, number of features for the best split, and the fraction of samples for fitting base learners. After tuning, the models are tested on the corresponding testing data.

Loss Functions. Each model is trained and tested four times using different loss functions: squared error (E^2), huber (h), and two versions of quantile (Q) loss with $\alpha = 0.45$ and $\alpha = 0.43$. Quantile loss is used to improve scoring by better aligning the loss function with scoring penalties. In some cases, changing the loss function slightly increases RMSE but significantly improves the score. Therefore, the loss functions are selected primarily based on their resulting scores rather than RMSEs. Table II shows the loss functions used to achieve these results.

Table II. The value of loss functions.

	FD001	FD002	FD003	FD004
GBRT I	$Q_{\alpha=0.43}$	$Q_{\alpha=0.45}$	$Q_{\alpha=0.43}$	E^2
GBRT II	E^2	E^2	E^2	$Q_{\alpha=0.45}$
GBRT III	h	$Q_{\alpha=0.45}$	h	h

As shown in Fig. 5, GBRT I produced the best RMSE values on all sub-datasets except FD002, where GBRT III slightly outperformed it with an RMSE of 13.35 compared to 13.64. GBRT II had slightly higher RMSEs than GBRT I and GBRT III. In terms of "Score", GBRT I had the best score on FD001, while GBRT III outperformed on FD002, FD003, and FD004, indicating that feature mapping may improve prediction timeliness in more complex operating conditions.

V. Observations and Discussions

In this section, we present and summarize our answers to **RQ1**, **RQ2**, and **RQ3**, and discuss the key observations.

A. Observations and Discussions for RQ1

In analyzing performance across different sub-datasets (Fig. 5), it is evident that the complexity of operating conditions significantly impacts the algorithm's performance. On sub-dataset FD001, which simulates a single operating condition and fault mode (HPC degradation), GBRT I, trained on the original set of features, yields the best RMSE and score. GBRT II, trained on the feature-selected dataset, produced the highest RMSE of 15.0, but scored better than GBRT III. This suggests that in simpler data like FD001, the relationships between sensor measurements and RUL are represented effectively without the need for additional feature mapping to enhance the model's understanding of degradation patterns.

FD002 simulates a more complex scenario with six operating conditions, leading to higher RMSE for GBRT I and GBRT II, indicating more complex degradation patterns. GBRT III has a slightly lower RMSE than GBRT I (13.35 vs. 13.64)

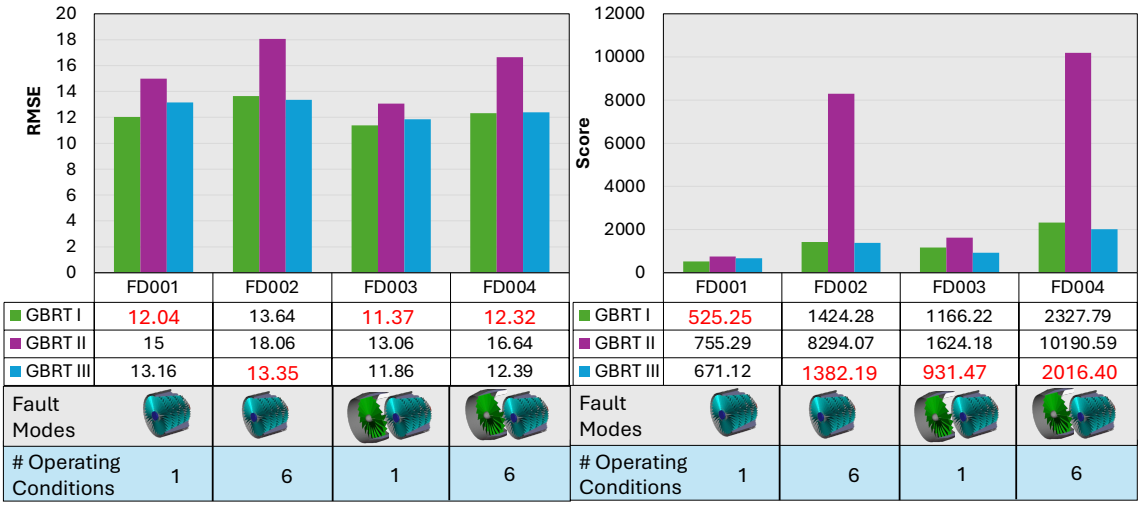


Fig. 5. The RMSE for each version of the algorithm (GBRT I: original features, GBRT II: feature selection, GBRT III: feature mapping) are displayed to the left, with the corresponding "Score" values (see Eq. 2) displayed to the right. Fault modes (HPC degradation, HPC and fan degradation) and the number of operating conditions for each sub-dataset are indicated between (a) and (b). The best RMSE and "Score" values are highlighted in red.

and achieves a better score (1382.19 vs. 1424.28). GBRT III's superior performance suggests that with multiple operating conditions, feature expansion better captures complex sensor-RUL correlations, improving RUL predictions.

FD003, simulating one operating condition and two fault modes (HPC and fan degradation), is less complex. Similar to FD002, GBRT I has a slightly better RMSE (11.35 vs. 11.71 for GBRT III), but GBRT III achieves a significantly better score (297.79 vs. 689.05). These results suggest that with multiple fault modes, feature expansion improves the timeliness of RUL predictions while maintaining high accuracy.

FD004, the most complex sub-dataset with six operating conditions and two fault modes, shows GBRT I with a slightly better RMSE (12.12 vs. 12.30 for GBRT III). However, GBRT III achieves a better score (2360.96 vs. 2502.38). These results suggest that in complex scenarios with multiple operating conditions and fault modes, feature expansion helps capture more intricate degradation patterns, improving scoring while maintaining competitive accuracy.

B. Observations and Discussions for RQ2

We then compare the performance of our methods with those from existing literature (Fig. 6). We selected several recently published methods, including ConvGAT [13], FCDAE-CNN-LSTM [14], as well as SVM-based, MLP-based, DBN-based [15], LSTM-based [16], and DCNN-based methods [17].

Comparing the value of RMSE, both GBRT I and GBRT III demonstrate competitive performance: GBRT III achieves the lowest RMSE of 13.4 on FD002, and GBRT I achieves the lowest RMSE of 12.3 on FD004. On FD001 and FD003, the ConvGAT method produces the lowest RMSEs of 11.3 and 11.0, respectively [13], but GBRT I performs comparably with RMSEs of 12.0 for FD001 and 11.4 for FD003. This indicates that GBRT I is as effective as some neural network-based methods in prediction accuracy. Additionally, while other methods show significant accuracy disparities between FD001/FD003 and FD002/FD004, both GBRT I and GBRT III maintain consistent accuracy across all datasets, demonstrating their adaptability to varying degradation patterns.

Comparing the "Scores," we observe that while neither proposed model achieves the best scores, GBRT III performs competitively on FD004 and FD002. On FD004, GBRT III's score of 2016.4 is second only to ConvGAT's 1231.17 [13]. On FD002, both GBRT I and GBRT III score competitively, with scores of 1424.28 and 1382.19, similar to the FCDAE-CNN-LSTM method's 1466.03, performing better than the other methods. However, ConvGAT still achieves a significantly better score of 771.61 on FD002.

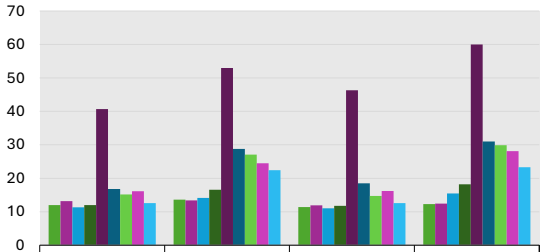
Overall, GBRT I and GBRT III achieve competitive accuracy across all four sub-datasets but only comparable scoring on FD002 and FD004. Since FD002 and FD004 simulate six operating conditions, this suggests that GBRT III is well-suited for predicting degradation patterns in complex conditions but struggles with timely predictions on the simpler dataset.

C. Observations and Discussions for RQ3

The commonly used scoring function in Eq. 2 compares turbofan engine prognostic solutions but lacks insight into critical performance aspects. Since it sums each trajectory's prediction error, scores on datasets with more trajectories, like FD002 and FD004, are inherently worse than those on smaller datasets. Besides, while this score accounts for accuracy weighted by timeliness, it doesn't explicitly show how these factors interact. Both accuracy and timeliness are crucial for selecting an optimal solution, yet a comprehensive benchmark addressing both remains undeveloped.

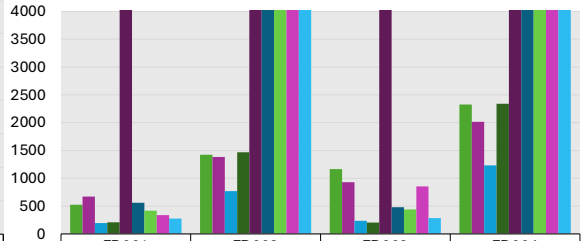
To address this, we introduce the **Margin-Adjusted Reliability Score (MARS)**, defined in Eq. 4. MARS offers a framework for evaluating how effectively an algorithm performs within a specified margin of maintenance anticipation.

$$MARS : s(b_1, b_2) = \frac{1}{n} \sum_{i=1}^{n} I(b_1 \leq d_i \leq b_2) \tag{4}$$



	FD001	FD002	FD003	FD004
GBRT I (Proposed)	12	13.6	11.4	12.3
GBRT III (Proposed)	13.2	13.4	11.9	12.4
ConvGAT	11.3	14.1	11.0	15.5
FCDAE-CNN-LSTM	12	16.6	11.8	18.2
SVM	40.7	53	46.3	60
MLP	16.8	28.8	18.5	31
DBN	15.2	27.1	14.7	29.9
LSTM	16.1	24.5	16.2	28.1
DCNN	12.6	22.4	12.6	23.3

a. RMSE Comparison



	FD001	FD002	FD003	FD004
GBRT I (Proposed)	525.25	1424.28	1166.22	2327.79
GBRT III (Proposed)	671.12	1382.19	931.47	2016.4
ConvGAT	197.43	771.61	235.26	1231.17
FCDAE-CNN-LSTM	209.44	1466.03	205.07	2338.9
SVM	7703	316483	22541	141122
MLP	561	14027	480	10444
DBN	418	9032	442	7954
LSTM	338	4450	852	5550
DCNN	274	10412	284	12466

b. Score Comparison

Fig. 6. Comparison of the performance of the GBRT I and GBRT III methods with various methods in previous literature. The values of RMSE for each algorithm are displayed in (a), with corresponding score values (see Eq. 2) displayed in (b). The best RMSE and "Score" values are highlighted in red.

where d_i is the difference between the predicted RUL and the true RUL for the i^{th} trajectory, b_1 is the lower bound, and b_2 is the upper bound of the margin. MARS evaluates the reliability of the algorithm in predicting RUL within a specified margin, with scores closer to 1 indicating higher reliability, and scores closer to 0 indicating lower reliability.

MARS defines a margin for maintenance anticipation and evaluates algorithm performance within that margin, penalizing late predictions while providing a clear, standardized measure of timeliness and accuracy. For instance, a margin might allow the true RUL to be five below or ten above the predicted RUL ($b_1 = -5, b_2 = 10$). Ideally, maintenance is deferred until the predicted RUL reaches 15 for economic efficiency but is performed before it drops to 5 for safety. The margin can be adjusted to allow more error in early predictions than in late ones, balancing safety with economic efficiency.

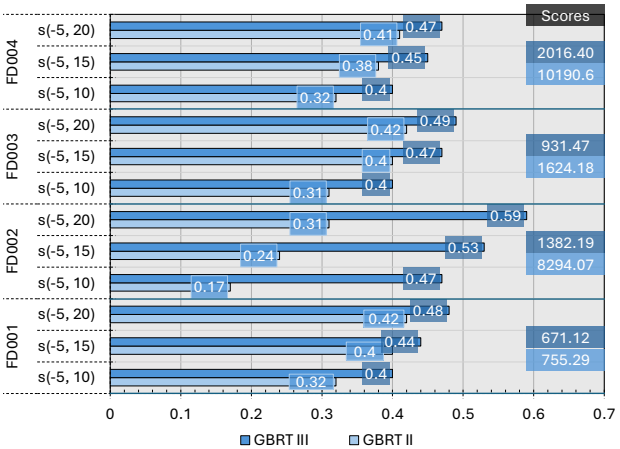


Fig. 7. Comparison of MARS values of GBRT II and GBRT III across sub-datasets FD001 through FD004 using margins (-5, 10), (-5, 15), and (-5, 20). The corresponding score values on each sub-dataset are shown to the right.

Figure 7 shows the MARS results for GBRT II and GBRT III, evaluated with margin settings of b_1=-5 and b_2=10, 15, and 20. Comparing MARS values instead of "Score" clarifies whether GBRT III's superior performance is due to better accuracy or improved timeliness. While "Scores" suggest better performance on FD001 and FD003, MARS provides a more nuanced view. Even within the narrowest margin, GBRT III consistently scores 0.4 or higher, with its best performance on FD002—a distinction less clear with conventional scoring. Conversely, GBRT II performs its worst on FD002, a detail not evident from score values alone. MARS thus offers a more effective assessment of timeliness and accuracy across scenarios for consistent comparisons.

VI. Concluding Remarks

In this study, we propose a GBRT-based model and the MARS evaluation metric for turbofan engine fault prognostics in aircraft, demonstrating the effectiveness of enhancing predictive maintenance. Our analysis shows how training the model on original, feature-selected, and feature-mapped sensor data impacts predictive accuracy across both simple and complex operating conditions. We found that GBRT I (original features) and GBRT III (feature mapping) models achieve competitive accuracy across all scenarios, excelling in complex cases (FD002 and FD004). However, their "Score" indicate that while accuracy is strong, prediction timeliness under simpler conditions (FD001 and FD003) needs improvement. Besides, our proposed MARS metric reveals critical differences that traditional scoring methods may overlook, offering a more industry-relevant metric for evaluating predictive models in real-world applications.

Acknowledgement

This work is supported in part by the National Science Foundation (NSF) grant CNS-2348151 and Commonwealth Cyber Initiative grant HC-3Q24-048.

References

[1] J. T. Bernardo, "Cognitive and functional frameworks for hard/soft fusion for the condition monitoring of aircraft," in *2015 18th International Conference on Information Fusion (Fusion)*. IEEE, 2015.

[2] R. Jafarpourmarzouni, S. Lu, Z. Dong *et al.*, "Enhancing real-time inference performance for time-critical software-defined vehicles," in *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*. IEEE, 2024, pp. 101–113.

[3] Y. Luo, D. Xu, G. Zhou, Y. Sun, and S. Lu, "Impact of raindrops on camera-based detection in software-defined vehicles," in *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*. IEEE, 2024, pp. 193–205.

[4] S. Lu, Y. Yao, and W. Shi, "CLONE: Collaborative learning on the edges," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 222–10 236, 2020.

[5] R. Walthall and R. Rajamani, *The Role of PHM at Commercial Airlines*. John Wiley & Sons, Ltd, 2018, ch. 18, pp. 503–534. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119515326.ch18

[6] J. Dalzochio, R. Kunst, J. L. V. Barbosa, P. C. d. S. Neto, E. Pignaton, C. S. ten Caten, and A. d. L. T. da Penha, "Predictive maintenance in the military domain: A systematic review of the literature," *ACM Comput. Surv.*, 2023.

[7] J. Chen and S. Lu, "An advanced driving agent with the multimodal large language model for autonomous vehicles," in *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST)*. IEEE, 2024, pp. 1–11.

[8] Z. Mian, X. Deng, X. Dong, Y. Tian, T. Cao, K. Chen, and T. A. Jaber, "A literature review of fault diagnosis based on ensemble learning," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107357, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623015415

[9] R. Siraskar, S. Kumar, S. Patil, A. Bongale, and K. Kotecha, "Reinforcement learning for predictive maintenance: A systematic technical review," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 885–12 947, 2023.

[10] A. Ucar, M. Karakose, and N. Kırımça, "Artificial intelligence for predictive maintenance applications: key components, trustworthiness, and future trends," *Applied Sciences*, vol. 14, no. 2, p. 898, 2024.

[11] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*, 2008, pp. 1–9.

[12] E. Ramasso and A. Saxena, "Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset," in *Annual Conference of the Prognostics and Health Management Society 2014.*, Fort Worth, TX, USA., United States, Sep. 2014. [Online]. Available: https://hal.science/hal-01145003

[13] X. Chen and M. Zeng, "Convolution-graph attention network with sensor embeddings for remaining useful life prediction of turbofan engines," *IEEE Sensors Journal*, vol. 23, no. 14, pp. 15 786–15 794, 2023.

[14] Y. Wang and Y. Wang, "A denoising semi-supervised deep learning model for remaining useful life prediction of turbofan engine degradation," *Applied Intelligence*, vol. 53, no. 19, pp. 22 682–22 699, 2023.

[15] C. Zhang, L. Pin, A. Qin, and K. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–13, 07 2016.

[16] C. Chen, J. Shi, N. Lu, Z. H. Zhu, and B. Jiang, "Data-driven predictive maintenance strategy considering the uncertainty in remaining useful life prediction," *Neurocomputing*, vol. 494, 04 2022.

[17] Q. Zhang, L. Yang, W. Guo, J. Qiang, C. Peng, Q. Li, and Z. Deng, "A deep learning method for lithium-ion battery remaining useful life prediction based on sparse segment data via cloud computing system," *Energy*, vol. 241, p. 122716, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544221029650

[18] Z. Cen, S. Hu, Y. Hou, Z. Chen, and Y. Ke, "Remaining useful life prediction of machinery based on improved sample convolution and interaction network," *Engineering Applications of Artificial Intelligence*, vol. 135, p. 108813, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197624009710

[19] P. Khumprom, A. Davila-Frias, D. Grewell, and D. Buakum, "A hybrid evolutionary cnn-lstm model for prognostics of c-mapss aircraft dataset," in *2023 Annual Reliability and Maintainability Symposium (RAMS)*, 2023, pp. 1–8.

[20] X. Cao, K. Peng, and R. Jiao, "Degradation modeling and remaining life prediction for a multi-component system under triple uncertainties," *Computers & Industrial Engineering*, p. 110432, 2024.

[21] S. Onofri, A. Marchioni, G. Setti, M. Mangia, and R. Rovatti, "Multi-class similarity-based approach for remaining useful life estimation," in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2024, pp. 01–06.

[22] D. K. Frederick, J. A. DeCastro, and J. S. Litt, "User's guide for the commercial modular aero-propulsion system simulation (c-mapss)," Tech. Rep., 2007.

[23] R. Munagala, "Gradient boost for regression explained," 2021, available: https://www.numpyninja.com/post/gradient-boost-for-regression-explained.

[24] Z. Huang, H. Zheng, C. Li, and C. Che, "Application of machine learning-based k-means clustering for financial fraud detection," *Academic Journal of Science and Technology*, vol. 10, no. 1, pp. 33–39, 2024.

[25] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications*, S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, Eds. Cham: Springer International Publishing, 2016, pp. 214–228.

[26] F. Heimes, "Recurrent neural networks for remaining useful life estimation," 11 2008, pp. 1 – 6.

[27] L. Peel, "Data driven prognostics using a kalman filter ensemble of neural network models," Oct. 2008, pp. 1–6, 2008 International Conference on Prognostics and Health Management (PHM) ; Conference date: 06-10-2008 Through 09-10-2008.

[28] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," 11 2008, pp. 1 – 6.

[29] S. Behera, A. Choubey, C. S. Kanani, Y. S. Patel, R. Misra, and A. Sillitti, "Ensemble trees learning based improved predictive maintenance using iiot for turbofan engines," *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:142503498

[30] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," in *2008 International Conference on Prognostics and Health Management*, 2008, pp. 1–17.

[31] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and health management*, vol. 1, no. 1, pp. 4–23, 2010.

[32] H. Wang, D. Li, D. Li, C. Liu, X. Yang, and G. Zhu, "Remaining useful life prediction of aircraft turbofan engine based on random forest feature selection and multi-layer perceptron," *Applied Sciences*, vol. 13, no. 12, p. 7186, 2023.