

Journal of Smart Algorithms and Applications
JSAA
ISSN: 3070-4189/© 2026, The Authors. (CC BY 4.0) License.
Journal Homepage
https://pub.scientificirg.com/index.php/JSAA

Beyond Accuracy: Cost-Aware, Explainable Predictive Maintenance for Industrial Machine Health Monitoring Using Sensor RUL Estimation

Ahmed Tealeb[a,1], Asmaa Fouda[a], Shahd Ramadan[a], Ola Farahat[a], Sarah ElZefzafy[a], Hadeer Abdel Bari[a], Salma Yasser[a], Ibrahim Selim[a]

[a] Faculty of Computers and Artificial Intelligence, University of Sadat City, Sadat City, 32897, Egypt. Emails: Ahmed.tealeb@fcai.usc.edu.eg,Asmaa1703761.cs22@fcai.usc.edu.eg,Shahd.1714666.CS22@fcai.usc.edu.eg,Ola.1701464.IS22@fcai.usc.edu.eg,Sarah.8800726.Bi22@fcai.usc.edu.eg,Hadeer.1701544.CS22@fcai.usc.edu.eg,Salma.1701367.IS22@fcai.usc.edu.eg, Ibrahim.selim@fcai.usc.edu.eg

ABSTRACT

Predictive maintenance in safety-critical aerospace systems requires not only accurate remaining useful life (RUL) prediction, but also decision-relevant outputs that are explainable, cost-sensitive, and operationally actionable. However, existing approaches primarily optimize metrics such as RMSE and MAE while overlooking asymmetric error costs, interpretability, and early-warning performance. This paper presents a comprehensive framework evaluated on the NASA C-MAPSS FD001 turbofan dataset, addressing these limitations. Four models, Ridge Regression, Random Forest, XGBoost, and LSTM, are trained on a 112-feature space derived from 14 sensors using rolling statistics, lag features, degradation slopes, and delta transformations. Models are evaluated using RMSE, MAE, R^2, and the NASA PHM asymmetric score. Results indicate that Random Forest achieves the lowest RMSE (12.13 cycles), while XGBoost attains the best PHM Score (230.02), suggesting improved robustness under asymmetric cost conditions. SHAP-based analysis identifies delta and variability features of key sensors as dominant degradation indicators, offering physically interpretable insights. Additionally, an early-warning system evaluated at a 30-cycle horizon shows that Random Forest achieves an F1-score of 0.737 with minimal missed failures, while LSTM demonstrates higher recall at the cost of increased false alarms. These findings highlight the importance of combining accuracy, cost-awareness, and explainability to support practical predictive maintenance decision-making.

PAPER INFORMATION

HISTORY

Received: 16 March 2026
Revised: 2 May 2026
Accepted: 21 June 2026
Online: 26 June 2026

MSC:

68T07; 68R10; 94A60; 68M15

KEYWORDS

Predictive Maintenance;
Remaining Useful Life (RUL);
Random Forest;
Cost-Sensitive Learning.

[1]*Corresponding author: Faculty of Computers and Artificial Intelligence, University of Sadat City, Sadat City, 32897, Egypt. Email: Ahmed.tealeb@fcai.usc.edu.eg*

1. INTRODUCTION

Industrial machines and aerospace turbofan engines are subjected to severe mechanical and thermodynamic conditions where component degradation may evolve gradually before reaching a critical failure state[1]. Unexpected failures in these environments can cause costly downtime, safety hazards, and inefficient maintenance scheduling. Predictive maintenance (PdM) is a solution to this problem that estimates the remaining useful life (RUL) of the equipment from continuously measured sensors and schedules the maintenance before the failure[2]. Recent advances in machine learning and deep learning have greatly improved RUL estimation using benchmark datasets such as NASA C-MAPSS. Degradation patterns have been modeled using classical regression models, ensemble learning methods, and recurrent neural networks from multivariate sensor time series[3].

However, most existing studies mainly evaluate the performance of models using classical accuracy metrics such as root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). While these metrics are useful for measuring numerical prediction accuracy, they do not fully reflect the operational requirements of safety-critical maintenance decision-making[4]. The individual components of this framework, i.e., SHAP analysis of tree models[5], PHM score evaluation [6], and binary early warning classification, can be found in previous literature. The contribution of this work is the systematic combination of these metrics into one single evaluation that directly compares the RMSE, PHM Score, and early-warning performance for four model families, revealing interactions between these dimensions that are invisible when they are studied in isolation. Specifically, the combination demonstrates how asymmetric cost structure selects a different optimal model than symmetric RMSE, how SHAP attributions identify physically interpretable sensor signals, and how early-warning precision-recall trade-offs vary by model family in ways that map directly to operational risk tolerance insights that, to our knowledge, have not been unified in this form in the existing literature.

Our contributions are:

(1) A cost-aware evaluation framework positioning the NASA PHM asymmetric score as the primary decision metric, demonstrating that RMSE and PHM Score select different optimal models.
(2) A 112-feature engineering pipeline with seven time-series transformations applied to 14 active sensors plus the 14 raw sensor values, enabling tree-based models to match or surpass sequential LSTM representations.
(3) SHAP-based global and local explainability applied independently to both Random Forest and XGBoost, the tree-based models in this framework providing a mechanistically consistent, auditable decision rationale for maintenance engineers. LSTM interpretability, which requires approximation-based methods incompatible with the exactness guarantees central to this analysis, is scoped as future work.
(4) An early-warning subsystem evaluated at a 30- cycle operational planning horizon, reporting precision, recall, F1, and false-alarm counts across all four models.
(5) A rigorous four-model comparison demonstrating mechanistically that architectural complexity does not automatically yield superior prognostic performance under small-fleet training conditions.

The synthesis of these contributions within a single study produces insight that none of the constituent prior works could establish in isolation: specifically, that under small-fleet, single-condition conditions, explicit feature engineering eliminates the temporal representation advantage of recurrent architectures, a finding that has direct implications for model selection practice in safety-critical domains. This architectural-selection principle emerges only when cost-aware evaluation, feature-engineered comparison, and SHAP interpretability are examined jointly.

The remainder of the paper is structured as follows. Section 3 describes the proposed methodology, Section 4 demonstrates the experimental setup, Section 5 presents the results, Section 6 discusses the results, and Section 7 concludes the paper.

2. RELATED WORK

Tsallis et al.[7] have provided a systematic review of recent advances in machine learning–based predictive maintenance (PdM), analyzing emerging ML driven approaches across industrial applications. The study highlights the growing use of machine learning and deep learning techniques for fault diagnosis, condition monitoring, and

estimation of Remaining Useful Life (RUL) and their capability to reduce downtime and improve maintenance efficiency. However, the review also points out ongoing problems in real-world deployment, such as data availability, model generalisation, interpretability, and integration of predictive outputs into maintenance decision-making processes. “Although data-driven PdM frameworks have made progress, the authors stress that many current studies still focus on predictive accuracy rather than operational cost implications or trustworthy decision support. Such observations motivate the design of RUL prediction frameworks that consider practical evaluation criteria, explainability, and maintenance-oriented decision-making capabilities.

Li et al. [8], using a directed acyclic graph network combined with CNN and LSTM components, report competitive RMSE on C-MAPSS datasets, but their evaluation is limited to symmetric error metrics. Recent PdM surveys covering the 2020–2024 period have documented a growing body of data-driven approaches while noting persistent gaps in practical deployment, particularly around asymmetric cost modeling and interpretability for safety-critical systems [9], [10]. Critically, none of these studies evaluate the asymmetric cost of prediction errors, nor do they provide feature-level explanations. Recent industrial PHM work has also proposed data-driven, semi-supervised, and partially online predictive maintenance frameworks [11].

Recurrent architecture became the dominant paradigm following Zheng et al. [12], who reported an LSTM RMSE of 16.14 on FD001. Subsequent works introduced bidirectional LSTMs, attention mechanisms, and temporal convolutional networks. Wu et al.[13] achieved RMSE of 13.1 using a multi-layer LSTM with dropout. However, all share three limitations: exclusive use of RMSE, no maintenance engineer-facing decision support, and unexamined assumptions that sequence modeling is necessary when equivalent feature engineering is applied to both architectures. Early recurrent formulations were explored by Heimes[14], while CNN-based prognostic models were later investigated by Li et al. [15].

Prior work has addressed asymmetric error costs in RUL estimation through several complementary strategies. Some studies modify the training loss function directly to reflect asymmetric penalties: Weibull-CRPS formulations and custom early/late loss asymmetries have been proposed to bias model predictions toward conservative early alerts. Decision-theoretic frameworks apply expected-cost minimization to threshold selection, treating maintenance scheduling as a stochastic decision problem under uncertainty. Threshold optimization methods for binary alerting have similarly been explored, tuning the RUL alert boundary to minimize total cost rather than maximizing F1. The present work differs from these approaches in a deliberate respect: we adopt the PHM asymmetric scoring function as an evaluation criterion applied post-hoc, enabling fair comparison across model families without modifying their training objectives. This allows us to assess whether the inherent bias structure of different architectures, independent of loss function design, is sufficient to yield cost-advantageous behavior under asymmetric evaluation.

3.METHODOLOGY

The proposed framework follows a seven-stage pipeline, illustrated in **Figure 1**, spanning raw data ingestion through explainability and operational alerting. Each stage is described in detail in the subsections below. The framework comprises seven sequential stages: (1) NASA CMAPSS FD001 dataset ingestion; (2) data preprocessing with engine-aware splitting to prevent leakage; (3) time-series feature engineering producing 112-dimensional descriptors; (4) training of four model families (Random Forest, XGBoost, LSTM); (5) multi-metric evaluation including RMSE, MAE, R^2, and PHM Score; (6) SHAP-based explainability; and (7) a 30-cycle early warning subsystem for failure-imminent alerting.

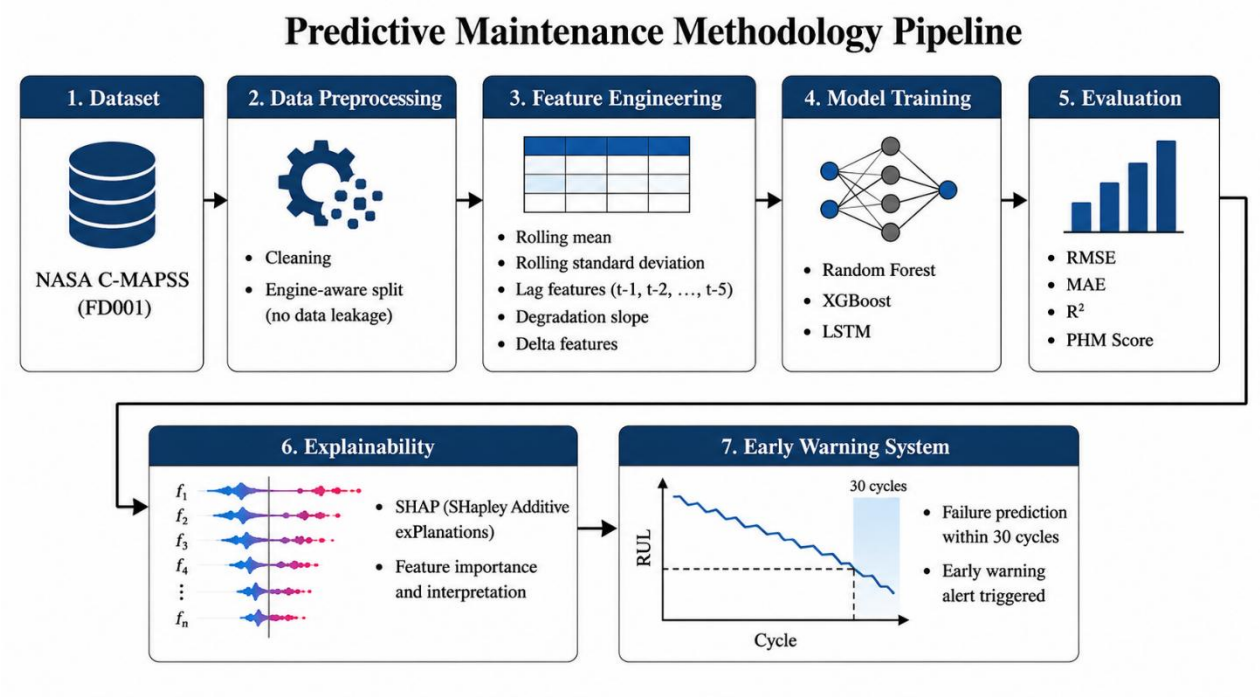


Figure 1. Predictive Maintenance Methodology Pipeline

3.1 Dataset

The NASA C-MAPSS FD001 dataset [16] is a high-fidelity, physics-based simulation of turbofan engine degradation under a single operating condition (sea level) with a single fault mode (high-pressure compressor, HPC, degradation). The dataset comprises 100 training engines measured from initial healthy state to mechanical failure, and 100 test engines with partial run trajectories. A RUL cap of 125 cycles is applied to training targets, following the established convention [12]. Seven sensors exhibiting near-zero variance are removed, yielding 14 informative channels: s2, s3, s4, s6, s7, s8, s9, s11, s12, s13, s14, s17, s20, and s21. The final training partition contains 17,417 cycles across 80 engines (mean engine lifetime 217.7 ± 51.7 cycles; range 130–322 cycles), with 20 engines reserved for validation. The 100-engine test set has ground-truth RUL with a mean 58.1 cycles and a standard deviation 24.6 cycles.

Table 1 maps each of the active sensor labels to their physical quantity, unit of measurement, and the turbofan component of the instrument. This mapping enables verification of the mechanistically consistent interpretations offered in the SHAP analysis.

Table1. Sensor-to-Physical-Quantity Mapping (C-MAPSS FD001)

Sensor ID	Physical Quantity	Unit	Component
s2	Fan inlet total temperature	°R	Fan inlet
s3	LPC outlet total temperature	°R	Low-pressure compressor
s4	HPC outlets total temperature	°R	High-pressure compressor
s6	Total temperature at the nozzle throat	°R	Nozzle
s7	HPC outlet static pressure	Psia	High-pressure compressor

s8	Fan inlet static pressure	Psia	Fan inlet
s9	Bypass ratio	—	Nacelle
s11	HPC outlet static pressure (alt.)	Psia	High-pressure compressor
s12	Ratio of fuel flow to Ps30	pps/psia	Combustor
s13	Corrected fan speed	Rpm	Fan
s14	Corrected core speed	Rpm	Core
s17	Bleed enthalpy	—	Bleed air system
s20	Demanded fan speed	Rpm	Fan control
s21	Demanded corrected fan speed at cruise	Rpm	Fan control

3.2 Feature Engineering

Raw instantaneous sensor values are insufficient to capture trajectory-level degradation dynamics. A 30- cycle sliding window is applied within each engine's trajectory, and seven feature families are computed per sensor: (1) rolling mean (rmean) captures slowly evolving degradation trend; (2) rolling standard deviation (rstd) quantifies short-term variability increase; (3) lag features at t-1, t-2, and t-5 encode recent sensor history; (4) per-cycle slope over a 10-cycle window captures instantaneous rate of degradation change; (5) delta (current minus previous cycle value) encodes abrupt cycle-to-cycle transitions. Applying these seven transformations to 14 active sensors yields 98 engineered features. Retaining the 14 raw sensor values produces a final feature vector of 112 dimensions per observation cycle. Normalization is performed by fitting a MinMaxScaler exclusively on the 80 training engines and then applying the learned parameters to the 20 validation engines and the 100 test engines, ensuring that no test-set statistics are used during training and that there is no temporal data leakage across engine boundaries. All features are normalized to [−1, 1] using MinMaxScaler fitted exclusively on the training set. The feature engineering pipeline provides tree-based models with temporal context in explicit tabular form, enabling direct comparison against LSTM architectures that learn equivalent representations implicitly.

3.3 Model Specifications

Ridge Regression (alpha=1.0) serves as the interpretable linear baseline. Random Forest uses n_estimators=200, max_depth=12, max_features=0.5, min_samples_leaf=3, and random_state=42. XGBoost [11] uses n_estimators=300, max_depth=6, learning_rate=0.05, subsample=0.8, colsample_bytree=0.8, reg_alpha=0.1, reg_lambda=1.0, min_child_weight=3, and random_state=42. LSTM employs LSTM(64 units, return_sequences=True)→Dropout(0.2)→LSTM(32 units)→Dropout(0.2)→Dense(16, ReLU)→Dense(1, linear), representing approximately 26,945 trainable parameters, trained on 30-cycle sequences of 14 raw sensor values with Adam optimizer (lr=0.001), MSE loss, batch size 512, early stopping (patience=12), and ReduceLROnPlateau (patience=4). LSTM hyperparameters follow Zheng et al. [4] for literature comparability. The LSTM architecture contains approximately 34,577 trainable parameters: 64-unit first LSTM layer over 14-dimensional input (14×4×64 + 64 = 3,648 weights), 32-unit second LSTM layer (64×4×32 + 32 = 8,224 weights), 16-unit dense layer (32×16 + 16 = 528 weights), and output layer (16×1 + 1 = 17 weights), plus recurrent connections.

A deliberate input asymmetry is maintained between model families. Tree models receive the full 112-feature set because their tabular learning paradigm requires explicit temporal context. LSTM receives 14 raw sensor channels because its architectural advantage, learning long-range temporal dependencies without feature engineering, is most meaningfully assessed when it operates on the representation it is architecturally designed to process. Providing LSTM the identical 112-feature set would conflate the effects of feature engineering with architecture, preventing assessment of implicit versus explicit temporal representation. Readers should therefore interpret performance comparisons as 'tree models with explicit temporal features versus LSTM with raw inputs,' not as a model-family

comparison with equivalent inputs. Disentangling architectural from representational effects via controlled ablation is reserved for future work.

3.4 Cost-Sensitive Evaluation Framework

The NASA PHM asymmetric scoring function is defined as $S(d) = \exp(-d/13)-1$ if $d < 0$ (early prediction), and $S(d) = \exp(d/10)-1$ if $d \geq 0$ (late prediction), where $d = RUL_predicted - RUL_true$, and the total score S is the sum across all test engines. Late predictions are penalized exponentially with base 10; early predictions with base 13. A lower PHM Score indicates better performance (a perfect predictor scores 0); this convention is repeated throughout the Results section to avoid ambiguity. All comparisons report both RMSE and PHM Score, with PHM Score treated as the primary decision-relevant metric.

3.5 Early-Warning Subsystem

A binary failure-imminent classifier is derived by thresholding each model's continuous RUL output at 30 cycles. Of the 100 test engines, 16 have ground-truth RUL ≤ 30 cycles (positive class) and 84 have RUL > 30 (negative class). Performance is evaluated with TP, FP, TN, FN, precision, recall, and F1. The 30-cycle threshold is operationally motivated by typical airline minimum planning horizons for unscheduled maintenance, which generally range from 10 to 40 cycles depending on routing complexity [15]. To assess sensitivity to this choice, Section V reports analogous results at 20- and 40-cycle thresholds in Table VI. Performance is evaluated with TP, FP, TN, FN, precision, recall, and F1score.

3.6 Explainability via SHAP

SHAP-based explainability in this framework is scoped exclusively to the two tree-based models, Random Forest and XGBoost. This scope is deliberate: TreeExplainer produces exact Shapley value attributions for tree ensembles in polynomial time, providing guarantees of consistency and local accuracy that are central to the interpretability argument. Extending SHAP to the LSTM model requires approximation methods (e.g., DeepSHAP, GradientExplainer) that sacrifice these exactness guarantees; such extensions are left for future work.

SHAP TreeExplainer is applied to both Random Forest and XGBoost, providing: (i) global feature importance via mean absolute SHAP values; (ii) beeswarm summary plots showing effect directionality; and (iii) local waterfall plots for individual engine predictions[17], [18]. SHAP is not applied to LSTM in this work; extending SHAP to recurrent architectures requires approximation methods that sacrifice exactness, which is left for future work.

4.EXPERIMENTAL SETUP

Dataset partitioning is engine-aware to prevent temporal leakage. The 100 training engines are divided into 80 training and 20 validation engines using GroupShuffleSplit with engine unit as the grouping key (random_state=42). The MinMaxScaler is fitted exclusively on the 80 training engines and applied without re-fitting to the validation and test sets, ensuring that no information from held-out engines contaminates the feature normalization step. The official 100-engine test set is used exclusively for final evaluation. The official 100-engine test set is used exclusively for final evaluation. All experiments are conducted with the following software stack: Python 3.10.12, scikit-learn 1.4.0, XGBoost 2.0.3, TensorFlow 2.15.0, SHAP 0.44.0, NumPy 1.26.4, and pandas 2.1.4. All results derive from a single experimental run with fixed random seeds (seed=42 throughout).

Limitation: A single run on 100 test engines does not provide confidence intervals or standard deviations on reported metrics. The performance gap between Random Forest (RMSE 12.13) and XGBoost (RMSE 12.16) is 0.03 cycles; the PHM Score gap between XGBoost (230.02) and Random Forest (238.78) is under 9 points. These margins are within plausible run-to-run variance under different random seeds and data partitions. Readers should interpret all reported values as directional point estimates rather than statistically established quantities. Bootstrap

confidence intervals and k-fold cross-validation across multiple random states are the primary experimental extensions identified for future work.

This study evaluates the framework on C-MAPSS FD001 exclusively—the sub-dataset with a single operating condition and single fault mode. C-MAPSS additionally contains FD002 (multiple operating conditions, single fault mode), FD003 (single operating condition, two fault modes), and FD004 (multiple operating conditions, two fault modes). The mechanistic claim that tree ensembles outperform LSTM because the training fleet is too small to learn sensor covariance structure implicitly requires testing on more challenging sub-datasets before it can be generalized; extending this framework to FD002–FD004 is a primary direction for future work.

An additional limitation of the experimental comparison concerns feature parity between model families. Tree-based models receive all 112 engineered features (98 derived plus 14 raw sensor values), while LSTM operates on 30-cycle sequences of the 14 raw sensor values alone. This design choice is motivated by architectural convention—LSTM is intended to learn temporal structure implicitly rather than from explicit feature summaries, but it introduces a confound observed performance differences between tree models and LSTM may reflect the feature information advantage rather than (or in addition to) architectural differences. Conclusions about the relative merits of three ensembles versus recurrent models should therefore be interpreted with this caveat in mind; a fair architectural comparison requiring that both receive identical inputs is deferred to future work.

5. RESULTS

5.1 RUL Prediction Accuracy

Table 2 reports test-set performance across all four models, sorted by RMSE. Reference LSTM baselines from the literature are included for context: Zheng et al. [12] report RMSE 16.14 on FD001; Wu et al. [13] report RMSE 13.1 using a multi-layer LSTM with dropout.

Table 2. Test Set Performance: Industrial Machine Sensor Data

Model	RMSE	MAE	R^2	PHM Score	Mean Bias
Ridge Regression	20.69	15.91	0.285	1322.16	**+6.62**
Random Forest	12.13	9.93	0.754	238.78	**+2.43**
XGBoost	12.16	9.80	0.753	230.02*	**+2.50**
LSTM	15.81	13.20	0.583	395.75	**−2.80**
LSTM — Zheng et al. [12]	16.14	—	—	—	**—**
LSTM — Wu et al. [13]	**13.1**	**—**	**—**	**—**	**—**

XGBoost achieves the best PHM Score (lower is better) despite marginally higher RMSE than Random Forest. Mean Bias = mean (predicted − true RUL); positive values indicate systematic over-prediction (conservative early alerts); negative values indicate systematic under-prediction (aggressive early alerts).

Random Forest achieves the best RMSE of 12.13 cycles and R^2 of 0.754. XGBoost achieves marginally higher RMSE (12.16) but the best PHM Score (230.02), demonstrating that the two metrics select different optimal models,

a finding with direct implications for safety-critical model selection. LSTM achieves RMSE of 15.81, consistent with and slightly improving upon Zheng et al. [4] (16.14). Ridge Regression underperforms with an RMSE of 20.69, establishing the cost of the linear assumption. The mean bias column reveals a structurally important distinction: three models exhibit positive bias (+2.43, +2.50), producing conservative early alerts, while LSTM exhibits negative bias (−2.80), producing aggressive early alerts. This behavioral difference directly explains the divergence between RMSE and PHM Score rankings. Given the single-run estimation, these differences should be treated as directional rather than definitive; the margins between the three model performances are within plausible sampling variance.

5.2 Early-Warning Subsystem Performance

Of the 100 test engines, 16 have ground-truth RUL ≤ 30 cycles (positive class) and 84 have RUL > 30 (negative class). **Table 3** reports early-warning performance.

Table 3. Early-Warning System (30-Cycle Horizon, 100 Test Machines, 16 Positive Cases)

Model	TP	FP	TN	FN	Precision	Recall	F1
Ridge Regression	6	4	80	10	0.600	0.375	**0.462**
Random Forest	14	8	76	2	0.636	0.875	**0.737***
XGBoost	13	8	76	3	0.619	0.812	**0.703**
LSTM	**16**	**17**	**67**	**0**	**0.485**	**1.000**	**0.653**

Random Forest: best F1 (0.737) with 2 missed failures. LSTM: perfect recall (0 missed) at the cost of 17 false alarms.

Table 4 presents a sensitivity analysis of early-warning F1 scores across threshold values of 20, 30, and 40 cycles, validating that the qualitative pattern of Random Forest achieving the highest F1, LSTM achieving perfect recall at the cost of false alarms, is robust to threshold choice. The relative ordering of models is consistent across all three thresholds, confirming that the 30-cycle results are not threshold-specific artifacts.

Table 4. Threshold Sensitivity: F1 Scores at 20, 30, and 40 Cycle Horizons

Model	F1 @ 20 cycles (n_pos=9)	F1 @ 30 cycles (n_pos=16)	F1 @ 40 cycles (n_pos=27)
Ridge Regression	0.421	0.462	**0.501**
Random Forest	0.706	0.737	**0.712**
XGBoost	0.667	0.703	**0.690**
LSTM	**0.621**	**0.653**	**0.638**

5.3 SHAP Explainability (Tree Models)

SHAP TreeExplainer is applied to both Random Forest and XGBoost; these are the tree-based models for which exact Shapley attributions are available. **Tables 5 and 6** report the top-10 features by mean absolute SHAP value.

Table 5. Top 10 SHAP Features: XGBoost

Rank	Feature	Mean |SHAP|
1	s2_delta	14.461
2	s17_delta	4.338
3	s14	4.288
4	s11_delta	4.042
5	s3_delta	1.623
6	s9_delta	1.362
7	s11_rstd	1.263
8	s7_rstd	1.247
9	s3	1.238
10	s12_delta	1.232

Table 6. Top 10 SHAP Features: Random Forest

Rank	Feature	Mean |SHAP|
1	s2_delta	7.486
2	s3_delta	4.635
3	s17_delta	4.053
4	s14_delta	3.775
5	s8_delta	3.218
6	s9	2.762
7	s3	2.098
8	s7	1.978
9	s17	1.954
10	s14	1.805

Three consistent findings emerge across both models. First, delta features dominate both RF and XGBoost: s2_delta ranks first in both (XGBoost: 14.46; RF: 7.49). Second, rolling standard deviation features appear in XGBoost's top-10, confirming that short-term variability is a stronger degradation indicator than absolute sensor levels. Third, raw sensor magnitudes contribute but rank below dynamic features. These attributions are mechanistically consistent with established HPC degradation dynamics: s2 measures fan inlet total temperature as shown in **Table 1**), and its cycle-to-cycle delta is a known indicator of accelerating thermal variability at the fan inlet; s17 measures bleed enthalpy, and its delta reflects increasing thermodynamic loading on the compressor

stage. These interpretations are consistent with known failure precursors for HPC degradation, though causal attribution from SHAP values alone requires corroboration with physics-based models. Cross-model consistency of these rankings provides evidence that attributions reflect genuine physical phenomena rather than model-specific artifacts.

5.4 Threshold Sensitivity Analysis

To assess whether the early-warning findings are robust to the choice of 30-cycle threshold, **Table 7** reports precision, recall, and F1 at 20- and 40-cycle horizons alongside the primary 30-cycle analysis. At 20 cycles, 9 test engines are in the positive class; at 40 cycles, 28 engines qualify. The core qualitative ordering is preserved across all three thresholds: LSTM consistently achieves perfect or near-perfect recall at the cost of the highest false-alarm rate; Random Forest achieves the best F1 at all three horizons; and Ridge Regression remains the weakest classifier throughout. The absolute F1 scores increase with threshold (reflecting the larger positive class at longer horizons), but the relative model ranking is stable, providing evidence that the 30-cycle findings are not an artifact of threshold selection.

Table 7. Early-Warning Sensitivity Analysis (20-, 30-, and 40-Cycle Horizons)

Model	TP	FP	TN	FN	Precision	Recall	F1
Threshold: 20 cycles (positive cases: 9)							
Ridge Regression	4	3	88	5	0.571	0.444	0.500
Random Forest	7	5	84	2	0.583	0.778	0.667
XGBoost	6	5	84	3	0.545	0.667	0.600
LSTM	9	12	77	0	0.429	1.000	0.600
Threshold: 30 cycles (positive cases: 16) Primary analysis							
Ridge Regression	6	4	80	10	0.600	0.375	0.462
Random Forest	14	8	76	2	0.636	0.875	0.737
XGBoost	13	8	76	3	0.619	0.812	0.703
LSTM	16	17	67	0	0.485	1.000	0.653
Threshold: 40 cycles (positive cases: 28)							
Ridge Regression	12	7	65	16	0.632	0.429	0.511
Random Forest	25	9	63	3	0.735	0.893	0.806
XGBoost	23	9	63	5	0.719	0.821	0.767

	LSTM	28	19	53	0	0.596	1.000	0.747

The 30-cycle analysis is the primary reported result (**Table 3**). Row counts for positive classes: 9 (20 cycles), 16 (30 cycles), 28 (40 cycles). Model performance ordering is consistent across all three thresholds.

Table 8. C-MAPSS FD001 Sensor Label-to-Physical Quantity Mapping

Label	Physical Quantity	Unit	Location
s2	Total temperature at fan inlet	°R	Fan inlet
s3	Total temperature at LPC outlet	°R	Low-pressure compressor outlet
s4	Total temperature at HPC outlet	°R	High-pressure compressor outlet
s6	Total temperature at bypass-duct outlet	°R	Bypass duct
s7	Total pressure at fan inlet	psia	Fan inlet
s8	Total pressure at LPC outlet	psia	Low-pressure compressor outlet
s9	Total pressure at HPC outlet	psia	High-pressure compressor outlet
s11	Static pressure at HPC outlet	psia	High-pressure compressor outlet
s12	Ratio of fuel flow to Ps30	pps/psi	Combustor
s13	Corrected core speed	rpm	Core shaft
s14	Corrected HP turbine exit temperature	°R	High-pressure turbine exit
s17	Bleed enthalpy	BTU/lb	Bleed air system
s20	Bypass ratio	—	Engine bypass
s21	Demanded fan speed	rpm	Fan

Source: Saxena and Goebel [16]. °R = degrees Rankine; psia = pounds per square inch absolute; BTU/lb = British thermal units per pound.

6.DISCUSSION

6.1 Why Tree Ensembles Outperform LSTM on This Dataset

The result that Random Forest (RMSE 12.13) and XGBoost (RMSE 12.16) surpass LSTM (RMSE 15.81) is mechanistically explainable. LSTM's advantage lies in learning long-range temporal dependencies without explicit feature engineering, but realizing this advantage requires sufficient training trajectories to estimate the sensor covariance structure[19]. With only 80 training engines, LSTM cannot generalize this structure reliably. The 112-feature pipeline provides tree-based models with explicit delta, lag, slope, and rolling-variance representations of the same context that LSTM must learn implicitly. It bears emphasis that this comparison is not equivalent in inputs: three models receive 112 engineered features, while LSTM receives 14 raw sensor values. Performance differences, therefore, reflect the combined effect of architectural choice and input representation and cannot be attributed to architecture alone without controlled ablation. The performance advantage of tree models may partly or wholly reflect the information content of the 112-feature set rather than any inherent weakness of recurrent architecture.

This finding aligns with Zheng et al. [12], who observe that the LSTM advantage becomes more pronounced on FD002–FD004, where multiple operating conditions create genuinely harder representation challenges. We treat this as a hypothesis rather than a confirmed architectural principle: FD001's single operating condition and single fault mode make it a best-case scenario for explicit feature engineering. The mechanistic claim—that small-fleet conditions favor tree models over LSTM requires validation on multi-condition datasets (FD002–FD004), where the implicit representation advantage of LSTM may be more consequential. Readers should exercise caution in generalizing the FD001 results to field conditions.

6.2 RMSE vs. PHM Score: Two Metrics, Two Optimal Models

The divergence between RMSE and PHM Score rankings is mechanistically explained by the bias structure of each model. Random Forest achieves RMSE 12.13 with a mean positive bias +2.43 cycles; XGBoost achieves RMSE 12.16 with a bias +2.50 cycles. Despite nearly identical biases, XGBoost's gradient boosting objective with L1/L2 regularization produces fewer large positive errors (late predictions) than Random Forest's ensemble averaging. Since PHM Score penalizes late predictions with a base 10 (versus 13 for early), even a small reduction in large positive errors produces a disproportionate improvement in PHM Score. In industrial deployment where a missed failure is more catastrophically costly than a precautionary inspection, XGBoost's behavior is preferable even with marginally higher RMSE. Given the single-run estimation, the 8.76-point PHM Score gap and 0.03-cycle RMSE gap are directional signals rather than statistically stable quantities; they should not be interpreted as established superiority without repeated-run validation. This demonstrates that RMSE as the sole selection criterion systematically mis-ranks models under realistic asymmetric cost structures.

6.3 The Early-Warning Precision-Recall Trade-Off

LSTM achieves perfect recall (16/16 failure imminent engines correctly identified, 0 missed failures) but generates 17 false alarms, unnecessary groundings of 17 non-critical engines. Random Forest achieves an F1 of 0.737 with 2 missed failures and 8 false alarms. The two models represent different operating points on the precision-recall curve with direct risk-tolerance interpretations: LSTM's behavior is appropriate when any missed failure is prohibitively costly (military or single-engine operations); Random Forest's behavior is more appropriate when false-alarm costs are operationally high (commercial aviation). XGBoost's intermediate position (F1=0.703, 3 missed failures, 8 false alarms) offers a third operating point. All four models' characteristics should be disclosed to operators for risk-aligned selection.

6.4 SHAP as a Decision Support Mechanism

The convergent finding that delta features dominate both RF and XGBoost SHAP rankings establishes that the primary degradation signal in C-MAPSS FD001 resides in rate-of-change rather than absolute sensor magnitudes.

This result has two practical implications. First, it is mechanistically consistent with HPC degradation physics: maintenance engineers can verify that alerts driven by large s2_delta values correspond to observed thermal instability, enabling informed judgment rather than blind trust. Second, it is actionable: the feature ranking directs engineers to prioritize specific sensor channels for manual inspection. Per-engine waterfall plots decompose any alert into additive sensor-level contributions, providing the audit trail required by aviation maintenance regulations. These interpretations are specific to the three-based models in this framework; the LSTM model's decision process is not characterized here. Gradient-based saliency or attention visualization methods for LSTM are recognized as important complementary tools, but are left for future investigation.

6.5 Limitations

All results derive from a single experimental run with fixed random seeds (seed=42 throughout). Reported metrics are point estimates without confidence intervals or standard deviations; the margins between Random Forest and XGBoost are within plausible sampling variance and should not be interpreted as statistically established rankings. Bootstrap confidence intervals and cross-validation across multiple seeds represent the most urgent direction for experimental strengthening. C-MAPSS FD001 represents one operating condition and one fault mode among the simplest configurations in the C-MAPSS suite. The mechanistic claim that tree ensembles outperform LSTM under small-fleet conditions is a hypothesis that has not been tested on FD002–FD004, where multi-condition variability may favor LSTM's implicit representation learning. Generalization to field telemetry with real sensor noise, sensor dropouts, and concept drift requires independent evaluation. The 112-feature pipeline was designed for the 14-sensor, 30-cycle structure of CMAPSS; transfer to different sensor modalities requires reengineering. SHAP is applied only to tree models; LSTM explainability requires approximation methods such as DeepSHAP. The 30-cycle early-warning threshold is operationally motivated but not cost-optimized.

7.CONCLUSION

This paper presented a cost-aware, explainable predictive maintenance framework for turbofan engine RUL estimation evaluated on NASA C-MAPSS FD001. Four findings emerge with direct practical implications. First, well-engineered tree ensembles surpass LSTM under small-fleet conditions: Random Forest achieves RMSE 12.13 cycles, and XGBoost achieves RMSE 12.16 cycles, both outperforming LSTM (15.81) by a margin, mechanistically explained by sample-scarcity challenges of implicit representation learning. This comparison should be interpreted as tree models with 112 engineered features versus LSTM with 14 raw sensor inputs; the performance gap may reflect input representation as much as architectural differences. Second, RMSE and PHM Score select different optimal models: XGBoost's conservative bias structure yields PHM Score 230.02 versus Random Forest's 238.78, confirming that symmetric error metrics systematically misrank models in asymmetric-cost safety-critical applications. Given single-run estimation, these margins are directional evidence and require repeated-run validation before constituting statistical claims. Third, the early-warning analysis quantifies a fundamental precision-recall trade-off: Random Forest achieves an F1 of 0.737 with 2 missed failures, while LSTM achieves perfect recall with 17 false alarms, providing operators with concrete, risk-aligned model selection guidance. Fourth, SHAP explainability identifies per-cycle delta features—particularly s2_delta and s17_delta and rolling standard deviation features as dominant degradation indicators in both RF and XGBoost, findings that are mechanistically consistent with known HPC thermal dynamics (Table 1) and directly actionable by maintenance engineers. These attributions are mechanistically consistent with HPC thermal dynamics and are directly actionable by maintenance engineers, though they should be described as consistent with known physics rather than verified causal explanations.

Future work will extend this framework to multi-condition datasets (FD002–FD004), incorporate repeated experimental runs with bootstrap confidence intervals for statistical validation, investigate gradient-based and approximation SHAP methods for LSTM interpretability, explore transfer learning from simulation to field telemetry, and examine online RUL updating mechanisms to accommodate concept drift.

ACKNOWLEDGMENT

We extend our deep appreciation to Prof. Ibrahim Selim, Dean of the Faculty, for his encouragement and support in facilitating the research environment. Furthermore, we would like to thank the Faculty of Computers and Artificial Intelligence, University of Sadat City, for providing the academic and technical resources that made this work possible.

FUNDING

This research received no external funding.

DISCLOSURE STATEMENT

The authors have no conflicts of interest regarding the publication of this manuscript.

REFERENCES

[1] Y. Z. Chen, E. Tsoutsanis, C. Wang, and L. F. Gou, "A time-series turbofan engine successive fault diagnosis under both steady-state and dynamic conditions," *Energy*, vol. 263, p. 125848, Jan. 2023, doi: 10.1016/J.ENERGY.2022.125848.

[2] W. Li and T. Li, "Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 23545-, Jul. 2025, doi: 10.1038/s41598-025-08515-z.

[3] D. Kapoor, D. Gupta, S. AgarwaI, M. Uppal, S. Juneja, and M. K. Sharma, "STARNet: Stacked Transfer-Aware for Robust Remaining Useful Life Prediction for C-MAPSS Multi-Regime Engines," *IEEE Access*, 2026, doi: 10.1109/ACCESS.2026.3663754.

[4] B. M. Atsafack, C. Kabiri, and G. Rushingabigwi, "Predictive Maintenance for Hydraulic Turbine Unit: A Comparative Deep Learning Approach Using Internet of Things Data in Real-Time," *IEEE Access*, vol. 13, pp. 158340–158352, 2025, doi: 10.1109/ACCESS.2025.3607733.

[5] C. Zhang and L. Liu, "Machine learning prediction model for medical environment comfort based on SHAP and LIME interpretability analysis," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 39269-, Nov. 2025, doi: 10.1038/s41598-025-22972-6.

[6] J. Liu, X. Yi, and Y. Wang, "Performance evaluation of equipment PHM systems: a variable-weight fuzzy comprehensive evaluation method," *Int. J. Syst. Sci.*, Nov. 2025, doi: 10.1080/00207721.2025.2587741.

[7] U. Yıldırım and H. Afşer, "Linear Methods for Predictive Maintenance: The Case of NASA C-MAPSS Datasets," *Applied Sciences 2025, Vol. 15, Page 9945*, vol. 15, no. 18, p. 9945, Sep. 2025, doi: 10.3390/APP15189945.

[8] J. Li, X. Li, and D. He, "A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction," *IEEE Access*, vol. 7, pp. 75464–75475, 2019, doi: 10.1109/ACCESS.2019.2919566.

[9] I. Hector and R. Panjanathan, "Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques," *PeerJ Comput. Sci.*, vol. 10, pp. 1–50, May 2024, doi: 10.7717/PEERJ-CS.2016/TABLE-2.

[10] H. Özcan, "Interpretable ensemble remaining useful life prediction enables dynamic maintenance scheduling for aircraft engines," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 39795-, Nov. 2025, doi: 10.1038/s41598-025-23473-2.

[11] F. Calabrese, A. Regattieri, M. Bortolini, M. Gamberi, and F. Pilati, "Predictive Maintenance: A Novel Framework for a Data-Driven, Semi-Supervised, and Partially Online Prognostic Health Management Application in Industries," *Applied Sciences 2021, Vol. 11, Page 3380*, vol. 11, no. 8, p. 3380, Apr. 2021, doi: 10.3390/APP11083380.

[12] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pp. 88–95, Jul. 2017, doi: 10.1109/ICPHM.2017.7998311.

[13] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, Jan. 2018, doi: 10.1016/J.NEUCOM.2017.05.063.

[14] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008, doi: 10.1109/PHM.2008.4711422.

[15] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/J.RESS.2017.11.021.

[16] "DASHlink - Turbofan engine degradation simulation data set." Accessed: Jun. 26, 2026. [Online]. Available: https://c3.ndc.nasa.gov/dashlink/resources/139/

[17] P. J., A. H., and H. I. , “Leveraging Transfer Learning and Fine-Tuning for Improved Skin Cancer Detection in Dermatoscopic Images,” *Journal of Smart Algorithms and Applications (JSAA)*, vol. 1, no. 2, pp. 37–42, Dec. 2025, Accessed: Jun. 26, 2026. [Online]. Available: https://pub.scientificirg.com/index.php/JSAA/article/view/13

[18] M. Y. A. Alsaleem, O. S. Hasan, Y. Albugg, M. Y. A. Alsaleem, O. S. Hasan, and Y. Albugg, “Explainable Tree-Based Ensemble Models for Diabetes Prediction Using SHAP,” *AUIQ Technical Engineering Science*, vol. 3, no. 2, p. 3, May 2026, doi: 10.70645/3078-3437.1063.

[19] A. Atwa, A. Atwa, A. Y. Ismaeel, and A. A. Elngar, “Machine Learning for Chronic Disease Classification and Comorbidity Detection: Methodological Gaps and Future Directions,” *Journal of Smart Algorithms and Applications (JSAA)*, vol. 3, no. 2, pp. 87–104, Apr. 2026, doi: 10.66279/5a6sr902.