Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Ranking Stability in Predictive-Maintenance Benchmarks: Engine-Grouped Validation, Evaluation Populations, and Historical Features

Waseem Alwan Zaboon
Wasit University
Tahsien Al-Quraishi
Melbourne Institute of Technology
Omar Al-Boridi
omar.alboridi@rmit.edu.au

RMIT University
Naseer Ali Hussien
Al-Ayen Iraqi University (AUIQ)

Research Article

Keywords:

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11721113/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Ranking Stability in Predictive-Maintenance Benchmarks: Engine-Grouped Validation, Evaluation Populations, and Historical Features

Waseem Alwan Zaboon[1], Tahsien Al-Quraishi[2], Omar Al-Boridi[3*], Naseer Ali Hussien[4]

[1]Mechanical Engineering Department, College of Engineering, Wasit University, Wasit, 52001, Iraq.
[2]School of Business, Melbourne Institute of Technology, Melbourne, Victoria, Australia.
[3]School of Computing Technologies, RMIT University, Melbourne, Victoria, Australia.
[4]Department of Computer Engineering Technology, College of Technical Engineering, Al-Ayen Iraqi University (AUIQ), Nasiriyah, Iraq.

*Corresponding author(s). E-mail(s): omar.alboridi@rmit.edu.au; oboridi@mit.edu.au;
Contributing authors: waseem_alwan@uowasit.edu.iq; tquraishi@mit.edu.au; naseerali@alayen.edu.iq;

Abstract

Model rankings in predictive-maintenance benchmarks can depend on the validation protocol, the prediction timestamps being scored, and the history represented in the inputs. This study investigates these dependencies using conventional and deep-learning classifiers on NASA C-MAPSS datasets. Models are developed using engine-grouped validation, fitting-only preprocessing, and common-timestamp evaluation. Controlled experiments examine alternative validation protocols, evaluation populations, and historical feature representations. A descriptive ranking-stability analysis further investigates how evaluation choices influence comparative model performance. The findings demonstrate that row-wise validation can substantially increase development performance without corresponding improvements in independent test results. Model rankings also vary across evaluation populations, highlighting the importance of consistent assessment protocols. Historical feature engineering delivers strong

predictive performance and demonstrates that appropriately selected temporal representations can substantially improve conventional machine-learning models. Furthermore, paired engine-level statistical inference and explanation-stability analyses provide additional evidence regarding the reliability and interpretability of model comparisons. The study contributes an integrated empirical assessment of evaluation sensitivity, ranking stability, and historical representation in explainable predictive maintenance. These findings provide methodological insights for developing more rigorous, transparent, and reproducible predictive-maintenance benchmarks, while recognising the need for further validation before operational deployment.

Keywords: Predictive maintenance, Engine-grouped validation, Benchmark evaluation, Feature engineering, Explainable artificial intelligence, C-MAPSS

1 Introduction

Predictive maintenance uses sensor histories to identify equipment degradation and support inspection planning. Sensor noise, rare near-failure observations, and dependence between successive measurements complicate model development. Temporal networks and conventional classifiers with engineered history provide different representations of these observations; their usefulness depends on the prediction task and evaluation design (Serradilla et al. 2022; Zhao et al. 2019).

A comparison can change when models are evaluated at different prediction times. A temporal network typically consumes sliding windows, whereas a conventional classifier produces a prediction for each measurement. Their native evaluation populations can therefore differ in label prevalence and degradation stage. Assessing every model at identical engine–cycle timestamps removes this particular source of mismatch while preserving each model's input representation. It does not make model complexity, effective history, or optimisation identical.

Validation boundaries matter for a different reason. Measurements from the same asset share operating characteristics and degradation histories. Engine-separated validation addresses prediction for unseen assets more directly than random row splitting. Data-dependent feature selection and scaling must also remain inside the fitting partition (Roberts et al. 2017; Cawley and Talbot 2010). These principles are established; the size and direction of their empirical effects should be measured rather than assumed.

Existing C-MAPSS classification benchmarks already compare conventional and neural models using tuning and engine-level validation (Özcan 2026). Consequently, neither adding familiar classifiers nor adopting grouped cross-validation, rolling features, or SHAP is in itself a methodological innovation. A more informative question is how robust the resulting *comparative conclusions* are to defensible changes in the experimental protocol. Two reports can obtain different model orderings simply because they score different cycles, represent different amounts of sensor history, or use different validation boundaries. These effects must be separated from evidence for architectural superiority.

This study therefore treats benchmark-conclusion stability as its primary object of analysis. Besides paired engine-level comparisons, it uses controlled validation diagnostics and matched evaluation populations, and derives a descriptive model-pair reversal measure from the reported performance tables. The resulting quantitative synthesis is exploratory and does not establish a novel estimator, statistically significant rank differences, or generalisation beyond the evaluated simulations.

1.1 Objectives and Research Questions

The objective is to determine which conclusions remain supported when model development, evaluation populations, and statistical dependence are made explicit. The study addresses four questions:

RQ1: How frequently do numerical model-pair orderings reverse when the evaluation population, sampling grid, or engine weighting changes on FD001 and FD002, and how do native-population scores differ in interpretation?
RQ2: How do row-wise validation and preprocessing fitted before splitting affect development scores and the configurations selected by LR and gradient boosting?
RQ3: Which LR-reference differences are supported by paired engine-level analysis across the two subsets?
RQ4: How sensitive are the FD001 LR results to rolling history, raw-only representation, oversampling, prediction horizon, and explanation sampling across training seeds?

1.2 Contributions and Scope

The contribution is a protocol-sensitive empirical assessment of predictive-maintenance model comparisons, rather than a new learning algorithm. The work is organised around four linked pieces of evidence:

(i) **Validation-design effects.** A crossed 2×2 diagnostic distinguishes engine-grouped from row-wise validation and fitting-only from deliberately global preprocessing for LR and histogram gradient boosting (HGB) on two C-MAPSS subsets. This tests whether larger development scores coincide with official-test gains.
(ii) **Quantified ranking sensitivity.** All seven model families are scored on identical timestamps under the main protocol. We additionally tabulate the fraction of model pairs whose mean-F1 ordering reverses under final-cycle evaluation, a coarser shared sampling grid, or equal-engine weighting. Native-instance results are reported separately because their model-specific observation populations are not matched.
(iii) **History-representation sensitivity.** Independently developed raw-input and rolling-feature LR pipelines assess how historical feature design changes near-failure classification on FD001. The rolling-window comparison is exploratory

and is not used to replace the pre-specified main model after observing test results.

(iv) **Uncertainty and explanation checks.** Paired engine-level bootstrap and randomisation procedures, multiplicity adjustment for the defined LR-reference family, repeated fits, optimisation diagnostics, and signed explanation checks indicate which interpretations are supported and which remain descriptive.

These components synthesise established procedures into a focused account of when a benchmark conclusion changes. The ranking-reversal measure is a descriptive summary of already evaluated populations, not a proposed novel statistical method. Candidate-count matching is a transparent development constraint, not proof of equal search quality or computational effort.

FD001 provides a single-condition benchmark, while FD002 adds multiple operating conditions. Both are simulated degradation datasets. The task is retrospective RUL-derived near-failure classification; maintenance decisions, costs, downtime, and field outcomes are outside the evaluated scope. Section 2 reviews related work, Section 3 defines the experiments, and Section 4 presents the results. Sections 5 and 6 discuss the findings and conclude the study.

2 Related Work

Predictive maintenance has been investigated using conventional machine-learning methods, deep neural networks, and temporal representations of sensor measurements. Reviews by Zhao et al. (2019) and Serradilla et al. (2022) describe developments in machinery health monitoring and deep-learning-based predictive maintenance, respectively. Fink et al. (2020) further highlight the challenges involved in translating prognostic models into industrial practice. Together, these studies motivate the use of temporal information while underscoring the need to distinguish benchmark predictive performance from operational utility.

Studies using NASA C-MAPSS have also established that classification-based prognostics can be evaluated with multiple model families and asset-aware development procedures. In particular, Özcan (2026) compare conventional and neural classifiers for a near-failure horizon and report engine-level validation and fitting-fold preprocessing. Consequently, neither near-failure classification nor engine-grouped validation is presented here as a new technique. The question addressed by the present study is instead how the conclusions of a model comparison vary when its validation boundaries, scored prediction times, and historical feature representations are made explicit. Table 1 positions this question relative to the cited literature.

The methodological foundations of grouped validation and leakage-aware preprocessing are therefore well established. Roberts et al. (2017) discuss dependence in hierarchical and temporally structured data, while Cawley and Talbot (2010) examine biases introduced by model selection. Building on these principles, the present study uses a controlled factorial diagnostic to separate changes in the validation boundary from changes in where feature selection and scaling are fitted. These experiments

Table 1 Selected related work and its methodological relevance to the present study. The comparison identifies the focus of the cited references; it does not imply that an unlisted procedure was absent from a particular study.

Reference(s)	Established focus	Connection to the present study
Zhao et al. (2019); Serradilla et al. (2022)	Reviews of learning methods and deep architectures for machinery monitoring and predictive maintenance.	Motivate the inclusion of conventional and temporal models; the present analysis focuses on evaluation sensitivity rather than proposing another architecture.
Fink et al. (2020)	Challenges and opportunities in applying prognostics to industrial systems.	Motivates cautious interpretation of simulated benchmark results and separates classification quality from maintenance benefit.
Özcan (2026)	Comparison of conventional and neural C-MAPSS classifiers, including engine-level validation and fitting-fold preprocessing.	Provides a closely related benchmark context; the present study concentrates on controlled protocol contrasts, aligned timestamps, and descriptive ranking stability.
Roberts et al. (2017); Cawley and Talbot (2010)	Validation of structured observations and bias arising from model selection.	Motivate engine-separated folds, fitting-only preprocessing, and diagnostics of row-wise versus grouped validation.
Hosmer et al. (2013); Cortes and Vapnik (1995); Breiman et al. (1984)	Foundations of logistic regression, support vector machines, and classification trees.	Support interpretable and conventional baselines used in the controlled comparisons.
Hochreiter and Schmidhuber (1997); Heimes (2008)	Recurrent learning and sequence-based approaches to prognostics.	Motivate the LSTM comparator and the need to align temporal and tabular models at common prediction timestamps.
(scikit-learn developers 2026)	Documented implementation of histogram gradient boosting.	Supports a nonlinear tabular comparator under the specified development budget.
Lundberg and Lee (2017); Angelov et al. (2021)	Additive feature attribution and broader issues in explainable artificial intelligence.	Motivate signed SHAP summaries and stability checks, without treating feature attributions as causal explanations.

measure effects under the tested configurations; they do not estimate a universal magnitude of data-leakage bias.

The predictive models themselves are also established. Logistic regression, support vector machines, and decision trees provide conventional reference models (Hosmer et al. 2013; Cortes and Vapnik 1995; Breiman et al. 1984); recurrent networks provide a temporal alternative (Hochreiter and Schmidhuber 1997; Heimes 2008); and histogram gradient boosting provides a nonlinear tabular comparator (scikit-learn developers 2026). Their inclusion allows the study to examine whether numerical model orderings remain stable under matched prediction timestamps and alternative evaluation populations. Although each model family is assigned the same number of candidate configurations, this constraint does not equalise search-space coverage, training instances, or computational effort.

Explainability is considered as a supporting analysis rather than a separate algorithmic contribution. SHAP produces additive feature attributions (Lundberg and Lee 2017), while broader explainable-AI research emphasises challenges in their

interpretation (Angelov et al. 2021). Here, signed logistic-regression coefficients and attribution-stability checks help characterise fitted predictions. Correlated historical features and the choice of explanation background limit mechanistic interpretation, and neither feature attribution nor classification performance establishes a causal degradation mechanism or a beneficial maintenance policy.

Overall, the research gap addressed is not a lack of predictive-maintenance classifiers or validation techniques. It is the need for a clearly documented, within-benchmark assessment of how model-comparison conclusions depend on validation design, prediction population, and historical representation. The pairwise rank-reversal analysis summarises changes in numerical ordering across reported evaluation rules; it is descriptive and does not establish statistical significance or a causal mechanism for those changes. Likewise, comparisons with scores reported in other publications are avoided when targets, feature representations, or evaluation populations are not aligned.

3 Methods

The protocol separates fitting, model selection, and benchmark testing. Main comparisons use engine-grouped development; deliberately relaxed validation protocols are analysed separately as diagnostics. All preprocessing, configuration searches, prediction files, and analysis scripts are included in the supplementary reproducibility archive.

3.1 Data and Prediction Target

FD001 contains 100 training and 100 test engines; FD002 contains 260 training and 259 test engines (Saxena and Goebel 2008; Saxena et al. 2008). Training trajectories run to failure, while test trajectories are truncated and accompanied by final-cycle RUL values. For engine i at cycle t, training RUL is $T_i - t$; test RUL is $r_i^{\text{final}} + T_i^{\text{obs}} - t$. RUL is capped at 125 cycles. The main binary target is $y_{it} = \mathbb{1}(\text{RUL}_{it} \leq 30)$. The engine identifier, cycle number, and RUL are not predictor inputs.

FD002 is fitted and tuned separately from FD001. It is an additional operating-condition evaluation within C-MAPSS, not a test of transfer from one fitted subset to another or independent field validation. Test labels do not select configurations or training epochs in these runs. Because this is an expanded revision following earlier benchmark analyses, results are interpreted as exploratory benchmark evidence rather than a prospectively blinded confirmation.

3.2 Features and Fitting-Only Preprocessing

The candidate pool contains three operating settings, 21 sensors, and each sensor's rolling mean and sample standard deviation: 66 features. The main rolling window is 20 cycles, uses the current and preceding observations within an engine, and never uses future measurements. Available history is used at the beginning of a trajectory; the standard deviation of a single observation is zero.

Within each fitting fold, the top 40 features are selected by absolute Pearson correlation with capped RUL. Constant-feature correlations are set to zero. Standardisation is fitted on those fitting observations. This selection criterion does not remove feature redundancy and differs from optimisation of the binary classification objective. The fitted feature list and scaler transform the corresponding validation observations without modifying their number or labels.

Random oversampling balances fitting labels, after which independent Gaussian noise with standard deviation 0.05 is added in standardised feature space. For temporal models, complete sequences are oversampled and noise is added to their elements. Validation and test instances remain unaugmented. Final models recompute the feature ranking and scaler using the complete training partition. FD002 uses global fitting-partition standardisation, without operating-condition-specific normalisation; this is a limitation of the tested representation.

3.3 Shared Development Budget and Model Configurations

All seven model families use the same three `StratifiedGroupKFold` partitions, grouped by engine, shuffled with seed 42. Each family receives four candidate configurations. The selection criterion is mean validation F1 at the same 50-cycle window endpoints used by the temporal representation. Tabular models fit all fitting rows, whereas LSTM fits sequences. Thus development populations are aligned at scoring but training instance counts and effective histories remain different.

The four candidate sets are: LR and linear SVM, $C \in \{0.01, 0.1, 1, 10\}$; KNN, $(k, \text{weighting}) \in \{(3, \text{uniform}), (7, \text{uniform}), (3, \text{distance}), (7, \text{distance})\}$; DT, $(\text{depth}, \text{minimum leaf}) \in \{(5, 5), (10, 1), (15, 5), (\text{None}, 10)\}$. MLP crosses L_2 penalty $\{10^{-4}, 10^{-2}\}$ with initial learning rate $\{0.001, 0.003\}$. Histogram gradient boosting (HGB) crosses maximum leaf count $\{7, 15\}$ with L_2 regularisation $\{0, 1\}$. LSTM crosses recurrent dropout $\{0, 0.3\}$ with Adam learning rate $\{0.001, 0.003\}$.

LR uses the `lbfgs` solver, L_2 regularisation, and at most 2,000 iterations. SVM uses squared-hinge loss and at most 100,000 iterations. MLP uses rectified-linear hidden layers of 64 and 32 units, Adam, and at most 300 iterations. Its internal row-based early stopping is disabled. HGB uses logistic loss, 200 boosting iterations, learning rate 0.1, and no internal early stopping. Other settings follow scikit-learn 1.8.0 defaults; selected configurations, iteration counts, training losses where available, and convergence warnings are recorded. Matching candidate counts does not make search-space coverage or wall-clock cost equivalent.

3.4 Temporal Representation and Epoch Selection

LSTM contains layers of 64 and 32 recurrent units, followed by a 16-unit rectified-linear dense layer and a sigmoid output. Binary cross-entropy is optimised with Adam in batches of 32. Sequences contain 50 observations, advance by 10 cycles, and include an extra final sequence when needed to reach an engine's last observation. Short trajectories are front-padded with zero in standardised feature space. No sequence crosses engine boundaries.

For each candidate and grouped fold, feature selection and scaling use the fitting engines only. Training lasts at most 100 epochs, with patience 10 on validation loss and restoration of the minimum-loss weights. Endpoint F1 from that restored model supplies the candidate's fold score. After selecting the candidate with highest mean fold F1, the median of its three selected epoch counts defines final training duration. Fresh models are then fitted to all training engines for seeds 42–46. The other model families are also refitted for these five seeds. Fold scores are development criteria, not unbiased estimates after configuration selection.

Tabular classifiers use their default class decision; LSTM uses a 0.5 sigmoid threshold. SVM's zero decision boundary supplies its classes and its decision function supplies ranking scores. These scores are not treated as calibrated failure probabilities.

3.5 Controlled Validation Diagnostics

A separate 2×2 diagnostic crosses validation boundaries (engine-grouped versus stratified rows) with preprocessing placement (fitting-fold-only versus complete-training-partition feature ranking and scaling before splitting). LR and HGB each use their four candidate configurations. The same split seed, number of folds, augmentation, and diagnostic scoring population of all validation rows are retained. Oversampling and noise remain fitting-only in every cell; the global-preprocessing cells deliberately expose validation information through feature ranking and scaling.

The diagnostic all-row scores are distinct from the main endpoint-based development scores. Each diagnostic selects its own configuration from mean fold F1. That configuration is refitted on all training observations and assessed on the official test rows across five seeds. No diagnostic uses official test labels during selection. Validation-engine overlap is recorded: grouped protocols have none, whereas row protocols can contain engines represented in fitting. These descriptive contrasts change the validation generalisation task and do not measure every possible form of leakage or imply that all contamination must increase every metric.

3.6 Test Populations and Paired Inference

The primary test population contains identical LSTM window endpoints for every model. Predictions are joined by engine and cycle with one-to-one and label-equality checks. Secondary analyses report each model's native population and one final-cycle prediction per engine. Final-cycle results average all five seeds. Primary F1 pools endpoint counts, so engines with more endpoints contribute more observations; using engines as the resampling unit does not itself equalise their contributions to that pooled statistic. Counts and positive prevalence are reported for each population. F1, precision, recall, ROC-AUC, and average precision (AP) are calculated per seed; AP summarises the precision–recall ranking without tuning a new threshold.

With resampling seed 20261018, pointwise 95% intervals use 2,000 paired bootstrap resamples of complete test engines (Efron and Tibshirani 1993). Each resample retains all selected timestamps of its sampled engines. F1 is calculated per seed and then averaged. Let C_{ms} denote the pooled vector $(\mathrm{TP}, \mathrm{FP}, \mathrm{FN})$ for model m and seed s.

For each comparator j, the statistic is

$$T_j = \frac{1}{5}\sum_{s=1}^{5}\left[f(C_{\mathrm{LR},s}) - f(C_{j,s})\right], \qquad f(C) = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{1}$$

For each of 9,999 randomisations, each engine's complete LR/comparator count vectors are exchanged with probability one half; the same exchange is applied across seeds. The two-sided Monte Carlo value uses $(1 + \#\{|T_j^{(b)}| \geq |T_j|\})/10{,}000$ (Phipson and Smyth 2010). Holm adjustment is applied jointly to the 12 LR-reference comparisons: six comparators in each of two subsets (Holm 1979). The comparison threshold is 0.05.

These procedures assume engine-level exchangeability and condition on fitted models and fixed benchmark training partitions. Seed repetition does not resample training fleets, and identical seed labels do not imply identical stochastic mechanisms across models. Bootstrap intervals are unadjusted pointwise intervals; their exclusion of zero need not agree with a multiplicity-adjusted randomisation decision. Non-significance does not establish equivalence.

3.7 Descriptive Model-Ranking Stability

To quantify the numerical stability of a benchmark ordering, we compare model-level mean F1 values under the main common-timestamp rule with three alternative rules: one final-cycle prediction per engine, a common stride-20 subset of endpoints, and equal-engine weighting of the main endpoints. Within each rule all seven models are assessed on the same selected engine–cycle observations; the rules themselves need not contain the same number or prevalence of observations. The already fitted main models are reused, so these comparisons isolate evaluation choices conditional on those fits, not retraining or training-protocol effects.

For a reference evaluation a and alternative b, let $F_m^{(a)}$ and $F_m^{(b)}$ be model m's mean F1 across the five fitted seeds. For all unordered pairs of models (m, n), define the descriptive reversal fraction

$$D(a,b) = \frac{\sum_{m<n} \mathbb{1}\left[(F_m^{(a)} - F_n^{(a)})(F_m^{(b)} - F_n^{(b)}) < 0\right]}{\sum_{m<n} \mathbb{1}\left[(F_m^{(a)} - F_n^{(a)})(F_m^{(b)} - F_n^{(b)}) \neq 0\right]}. \tag{2}$$

Thus a reversal requires opposite strict orderings; pairs tied in either rule are excluded from the denominator. With seven model families there are at most 21 eligible pairs. The values are derived from the reported, rounded mean F1 tables and should be interpreted as descriptive summaries; differences close to reporting precision may affect a rank. No permutation tests, confidence intervals, or significance claims are attached to D. Native-instance rankings are deliberately excluded from this calculation because tabular and temporal models have different native observation populations, making their pairwise scores unsuitable for a matched-comparison measure.

3.8 Sensitivity, Ablation, and Explanation

FD001 LR sensitivity checks independently retune C using the same grouped endpoint criterion. They compare rolling windows of 10 and 40 cycles with the main 20-cycle setting; near-failure horizons of 20 and 40 cycles with the main 30-cycle target; a raw-only pipeline using the 24 sensor/operating inputs; and a pipeline without oversampling but retaining noise. The 20-cycle window is retained as the fixed reference from the initial benchmark protocol, not as a demonstrated optimum. The sensitivity results do not retrospectively select a replacement main window. Raw-only development independently ranks its available inputs and refits its scaler. It therefore compares usable pipelines rather than deleting columns from an already selected engineered representation. Dimensionality still differs, so the contrast does not isolate history independently of feature count.

Paired engine-bootstrap intervals describe F1 differences for variants with the same target. Horizon changes alter the labels and are reported descriptively without a component-benefit interpretation. Sampling checks evaluate a stride of 20 on endpoints contained in the main population and compute pooled F1 after assigning each engine equal total observation weight, using observation weights $w_{it} = 1/n_i$ for an engine with n_i selected endpoints. These checks reuse fitted models; they do not retune the LSTM window length or architecture. Appendix B reports each model's mean F1, seed variability, numerical rank, raw sample and positive counts, and prevalence. Weighted positive prevalence is also stated for the equal-engine analysis.

A separate FD002 MLP convergence check increases the maximum iteration budget from 300 to 1,000 for five fresh fits. It preserves the main training-selected hyperparameters, fitting observations, augmentation seeds, optimiser settings, and stopping tolerance. The larger cap is fixed for this diagnostic rather than selected by test performance. The main comparison retains its original 300-iteration specification; Appendix A.1 reports whether the extended fits meet the optimiser's stopping criterion, their population-specific metrics, and a pointwise paired engine-bootstrap interval for the common-population F1 change. This check does not retune all MLP candidates under a new budget or establish a global optimum.

For the final FD001 LR fits, SHAP uses a linear explainer with an independent-feature masker retaining 1,000 balanced, augmented fitting-background observations per seed (Lundberg and Lee 2017). A fixed sample of 1,000 test rows supports comparisons across seeds. Additive reconstruction is checked in log-odds units; signed standardised coefficients are exported alongside attributions. Rank correlation and top-five-set Jaccard overlap summarise attribution stability across the ten seed pairs, assigning zero importance to absent features. This is a descriptive seed/background check, not a full conditional-explanation sensitivity analysis.

4 Results

Results first describe evaluation populations and common-timestamp comparisons, then report paired inference and diagnostic validation contrasts. Population-specific

rankings are quantified using descriptive model-pair reversals, followed by feature-history sensitivity and explanation checks. Statistical inference and rank-reversal summaries serve different purposes and are not conflated.

The training and test population sizes, common endpoints, final-cycle counts, and positive-label prevalence are summarised in Table 2.

Table 2 Evaluation populations. Positives use the main 30-cycle target.

Subset	Population	Engines	N	Positive	Positive (%)
FD001	Training rows	100	20,631	3100	15.03
FD001	Test rows	100	13,096	332	2.54
FD001	Common endpoints	100	962	55	5.72
FD001	Final cycles	100	100	25	25.00
FD002	Training rows	260	53,759	8060	14.99
FD002	Test rows	259	33,991	1087	3.20
FD002	Common endpoints	259	2,501	162	6.48
FD002	Final cycles	259	259	61	23.55

4.1 Common-Timestamp Model Comparison

Table 3 reports the FD001 common-timestamp performance of the seven model families.

Table 3 FD001 common-timestamp performance (percentages). F1 is mean ± sample SD over five seeds; pointwise intervals resample complete engines. AP is average precision.

Model	F1 ± SD	95% CI	Precision	Recall	AP
LR	80.07 ± 0.36	[70.61, 87.58]	72.70	89.09	89.26
SVM	79.81 ± 0.29	[69.94, 87.50]	72.27	89.09	89.13
KNN	72.88 ± 1.10	[61.00, 81.92]	64.39	84.00	70.83
DT	70.15 ± 2.36	[59.76, 78.04]	63.64	78.18	63.16
MLP	72.10 ± 1.07	[60.77, 80.61]	73.42	70.91	82.44
HGB	78.57 ± 1.11	[68.69, 85.88]	73.81	84.00	87.22
LSTM	83.36 ± 3.91	[75.24, 88.71]	80.63	86.55	93.23

On FD001, LSTM has the highest mean common-timestamp F1 (83.36%). This is a numerical ranking conditional on the specified representations and four-candidate development budget.

The corresponding FD002 common-timestamp performance is reported in Table 4.

On FD002, HGB has the highest mean common-timestamp F1 (65.44%). This is a numerical ranking conditional on the specified representations and four-candidate development budget.

Figure 1 compares common-timestamp F1 and the paired engine-bootstrap intervals across the two subsets.

Table 4 FD002 common-timestamp performance (percentages). F1 is mean ± sample SD over five seeds; pointwise intervals resample complete engines. AP is average precision.

Model	F1 ± SD	95% CI	Precision	Recall	AP
LR	62.93 ± 0.11	[55.69, 69.09]	48.34	90.12	80.81
SVM	62.86 ± 0.15	[55.60, 69.04]	48.22	90.25	81.04
KNN	54.64 ± 0.81	[48.37, 59.65]	43.38	73.83	42.93
DT	54.66 ± 1.18	[48.52, 59.68]	41.57	79.88	45.12
MLP	61.81 ± 3.10	[55.45, 67.06]	48.69	85.19	76.99
HGB	65.44 ± 0.62	[58.92, 71.31]	50.90	91.60	81.74
LSTM	63.67 ± 1.76	[57.05, 69.43]	52.96	79.88	74.13

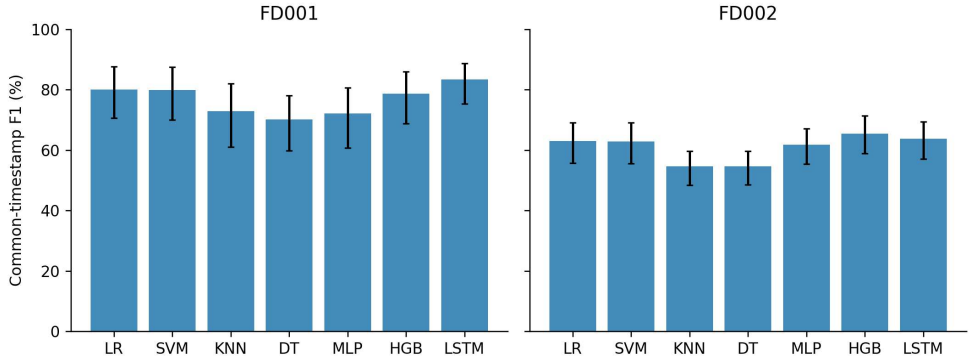

Fig. 1 Common-timestamp F1 across five fitted seeds on each subset. Error bars are pointwise 95% paired engine-bootstrap intervals conditional on the fitted models.

4.2 Paired Engine-Level Comparisons

The paired engine-level LR-reference comparisons and multiplicity-adjusted results are reported in Table 5.

Table 5 LR-reference F1 differences in percentage points. Intervals are pointwise and unadjusted; randomisation p-values are Holm-adjusted jointly across all 12 comparisons.

Subset	Comparison	Difference	95% CI	Adjusted p
FD001	LR–SVM	0.26	[0.00, 0.84]	1.0000
FD001	LR–KNN	7.18	[0.65, 13.82]	0.2547
FD001	LR–DT	9.92	[4.25, 15.93]	0.0110
FD001	LR–MLP	7.97	[0.49, 15.61]	0.2792
FD001	LR–HGB	1.50	[-2.50, 5.40]	1.0000
FD001	LR–LSTM	-3.29	[-10.08, 3.03]	1.0000
FD002	LR–SVM	0.08	[-0.15, 0.33]	1.0000
FD002	LR–KNN	8.29	[3.47, 13.24]	0.0110
FD002	LR–DT	8.28	[4.95, 11.73]	0.0012
FD002	LR–MLP	1.12	[-1.98, 4.51]	1.0000
FD002	LR–HGB	-2.51	[-5.15, 0.13]	0.4480
FD002	LR–LSTM	-0.74	[-4.77, 3.11]	1.0000

For FD001, the adjusted LR–DT comparison supports LR at the 0.05 threshold ($p = 0.0110$).

For FD002, the adjusted LR–KNN comparison supports LR at the 0.05 threshold ($p = 0.0110$).

For FD002, the adjusted LR–DT comparison supports LR at the 0.05 threshold ($p = 0.0012$).

Neither the LSTM–LR nor HGB–LR comparison meets the adjusted threshold on either subset. Other numerical differences should not be described as equivalence or as statistically established superiority. The intervals and adjusted tests use different reference procedures and multiplicity treatment.

4.3 Controlled Validation Diagnostics

Table 6 presents the controlled validation and preprocessing diagnostics for LR and HGB.

Table 6 Diagnostic all-row validation and official all-row test F1 (percentages). Global preprocessing deliberately uses the complete training partition before validation splitting. These validation scores are separate from endpoint-based main tuning.

Subset	Model	Split / preprocessing	Validation	Test ± SD
FD001	LR	Grouped / fitting	86.35	73.93 ± 0.23
FD001	LR	Rows / fitting	86.98	73.77 ± 0.25
FD001	LR	Rows / global	86.96	73.77 ± 0.25
FD001	LR	Grouped / global	86.20	73.93 ± 0.23
FD001	HGB	Grouped / fitting	86.74	73.61 ± 0.38
FD001	HGB	Rows / fitting	94.20	73.61 ± 0.38
FD001	HGB	Rows / global	94.30	73.61 ± 0.38
FD001	HGB	Grouped / global	86.87	73.47 ± 0.69
FD002	LR	Grouped / fitting	79.60	57.95 ± 0.16
FD002	LR	Rows / fitting	79.50	57.95 ± 0.16
FD002	LR	Rows / global	79.57	57.95 ± 0.16
FD002	LR	Grouped / global	79.55	57.95 ± 0.16
FD002	HGB	Grouped / fitting	80.25	59.43 ± 0.56
FD002	HGB	Rows / fitting	80.91	59.43 ± 0.56
FD002	HGB	Rows / global	81.16	59.55 ± 0.38
FD002	HGB	Grouped / global	80.01	59.55 ± 0.38

For FD001 LR, row-wise rather than grouped validation changes the best mean validation F1 by +0.63 percentage points; global rather than fitting-only preprocessing under grouped validation changes it by -0.15 points.

For FD001 HGB, row-wise rather than grouped validation changes the best mean validation F1 by +7.46 percentage points; global rather than fitting-only preprocessing under grouped validation changes it by +0.13 points.

For FD002 LR, row-wise rather than grouped validation changes the best mean validation F1 by -0.09 percentage points; global rather than fitting-only preprocessing under grouped validation changes it by -0.04 points.

For FD002 HGB, row-wise rather than grouped validation changes the best mean validation F1 by +0.66 percentage points; global rather than fitting-only preprocessing under grouped validation changes it by -0.24 points.

Row-wise folds contain validation engines also represented in fitting, whereas grouped folds have no engine overlap. The contrasts are descriptive: changing the validation task or exposing validation information need not uniformly increase scores, and validation gains need not yield better official-test performance.

4.4 Native-Population and Final-Cycle Results

Table 7 reports the native-population and final-cycle F1 results for both subsets.

Table 7 Secondary F1 (percentages), mean ± SD over five seeds. Native tabular results use all test rows; native LSTM results use window endpoints. Final-cycle results use one observation per engine.

Model	FD001 native	FD001 final	FD002 native	FD002 final
LR	74.97 ± 0.22	90.20 ± 0.00	57.95 ± 0.16	80.66 ± 0.37
SVM	73.49 ± 0.22	90.20 ± 0.00	57.87 ± 0.11	80.38 ± 0.25
KNN	66.36 ± 0.96	83.52 ± 2.25	43.78 ± 0.40	77.15 ± 0.55
DT	61.69 ± 2.91	85.46 ± 4.63	44.41 ± 0.93	74.84 ± 2.96
MLP	68.51 ± 1.68	84.11 ± 1.37	54.45 ± 3.04	83.18 ± 1.98
HGB	74.46 ± 0.78	88.89 ± 1.72	59.55 ± 0.38	82.09 ± 0.74
LSTM	83.36 ± 3.91	92.32 ± 2.33	63.67 ± 1.76	82.20 ± 1.20

Population counts and positive prevalence in Table 2 explain why these metrics answer different questions. Native values do not support a paired ranking across model families. Final-cycle metrics describe the truncation endpoints supplied by the benchmark and should not be interpreted as performance over a complete deployed monitoring trajectory.

For RQ1, the numerical leader on FD001 is LSTM in all three populations, but the middle ranks change: SVM exceeds HGB on common timestamps, whereas HGB exceeds SVM on native instances. KNN exceeds MLP on common timestamps, whereas MLP exceeds KNN on native instances. At final cycles, DT moves above both KNN and MLP, and LR and SVM tie. These are rankings of mean F1 rather than additional significance tests.

FD002 gives a stronger population dependence: HGB leads on common timestamps, LSTM leads when each family is scored on its native instances, and MLP leads in the final-cycle population. The native LSTM score uses windows while the native tabular scores use rows, so that apparent native lead cannot establish superiority on matched cases. The final-cycle MLP lead is likewise secondary and has not been subjected to the primary paired comparison family. All these rankings use the original main model configurations, including the 300-iteration MLP budget.

4.5 Numerical Model-Ranking Stability

Equation 2 summarises the model-pair reversals implied by Tables 3–4, Table 7, and Appendix B. Table 8 compares each alternative rule against the original common-timestamp mean-F1 ranking; no additional model fitting or individual-engine inferential analysis was performed for this summary.

Table 8 Descriptive reversals of numerical mean-F1 orderings relative to the main common-timestamp population. Eligible pairs exclude ties in either compared rule. All models share the same evaluation instances within each alternative rule; the rules differ in their sampled observations or weights.

Subset	Alternative rule	Reversals / eligible pairs	Reversals (%)	Alternative numerical leader
FD001	Final cycle	3 / 20	15.0	LSTM
FD001	Stride 20	2 / 20	10.0	LR and SVM (tie)
FD001	Equal-engine weighting	0 / 21	0.0	LSTM
FD002	Final cycle	6 / 21	28.6	MLP
FD002	Stride 20	2 / 21	9.5	HGB
FD002	Equal-engine weighting	0 / 21	0.0	HGB

FD001 retains LSTM as the numerical leader under final-cycle scoring, but three other model-pair orderings reverse; under stride-20 sampling, LR and SVM jointly exceed LSTM on mean F1, yielding two reversals. FD002 shows stronger final-cycle dependence: six of the 21 model-pair orderings reverse and MLP becomes the numerical leader, compared with HGB under the main common-timestamp rule. Equal-engine weighting changes F1 values but does not reverse any pairwise ordering in either subset.

These counts identify sensitivity of *reported numerical rankings* to evaluation choices, not proven changes in relative predictive performance. In particular, the final-cycle and stride-20 populations differ from the main population in sample size and event prevalence, and the descriptive reversals are not multiplicity-adjusted superiority tests. Native-population scores remain useful for representing each model's deployment-like output frequency, but cannot support a paired cross-family rank-stability calculation.

4.6 LR Sensitivity and Explanation

The FD001 LR feature-history, sampling, and target-horizon sensitivity results are summarised in Table 9.

Table 9 FD001 LR sensitivity on common timestamps. Difference intervals are for variant minus main LR; horizons alter the target and have no component-difference interval. All values are percentages or percentage points.

Variant	F1 ± SD	Precision	Recall	Difference 95% CI
raw only	72.82 ± 0.74	64.70	83.27	[-13.64, -1.13]
no oversampling	76.92 ± 0.00	81.63	72.73	[-12.09, 4.55]
horizon20	77.16 ± 0.54	64.44	96.15	—
horizon40	76.94 ± 0.44	69.48	86.19	—
rolling10	73.80 ± 0.50	65.82	84.00	[-12.43, -1.24]
rolling40	87.40 ± 0.91	81.27	94.55	[2.87, 12.83]

These variants were developed independently using training-only grouped selection. Rolling-history effects remain conditional on representation and target; the raw-only contrast also changes feature dimensionality. Oversampling is evaluated as a

precision–recall trade-off, and horizon-specific F1 values are not directly comparable as solutions to the same classification task.

The 40-cycle LR variant reaches 87.40% mean F1, compared with 80.07% for the 20-cycle reference: a change of 7.34 percentage points, with pointwise paired 95% interval [2.87, 12.83]. This is substantial sensitivity to the historical representation within the tested LR pipeline. The interval is descriptive and unadjusted across sensitivity checks.

The 20-cycle reference remains the main specification because replacing it after inspecting these test results would turn a sensitivity check into test-based window selection. Its retention does not establish that 20 cycles is optimal. Although the 40-cycle LR point estimate exceeds the main LSTM point estimate, this is not a matched-representation, jointly tuned architecture comparison or an adjusted test of LR superiority. Both representations would need training-only window selection and a new independent evaluation to support that claim.

Appendix B gives the complete model-level sampling results. On FD001, the stride-20 grid gives LR and SVM equal highest mean F1, with LSTM third; the main-population ordering is therefore not preserved. On FD002, HGB and LSTM retain first and second positions, while MLP moves above LR and SVM. Equal engine weighting preserves the original numerical ranking on both subsets. These are changes in evaluation populations or weights without refitting, not additional architecture-superiority tests.

For seed 42, sensor 20 rolling mean and raw sensor 11 have positive standardised LR coefficients, whereas sensor 17 rolling mean has a negative coefficient. These are conditional model slopes, not physical causal effects. SHAP magnitudes additionally reflect variation relative to the chosen background; coefficient magnitude and attribution magnitude therefore need not give the same ranking.

SHAP additive reconstruction has maximum absolute log-odds error 1.07e-14. Across seed pairs, mean feature-importance rank correlation is 0.932, and mean top-five Jaccard overlap is 0.867. These checks describe stability under the selected independent-feature backgrounds; correlated-feature allocation and causal interpretation remain unresolved.

Figure 2 displays signed SHAP contributions for the FD001 LR model.

5 Discussion

The principal finding is not that one architecture dominates the other model families, but that the numerical conclusions of predictive-maintenance benchmarking are conditional on several choices often treated as secondary. The common-timestamp design aligns the prediction instances scored across families, the validation diagnostics test selected protocol boundaries, and the ranking-reversal analysis quantifies sensitivity to changing the sampled prediction population. These elements collectively sharpen the interpretation of an established comparison task rather than introducing a new prediction algorithm.

The ranking-stability synthesis reveals a distinction between stable and unstable evaluation choices. Equal-engine weighting retains all main model orderings on both

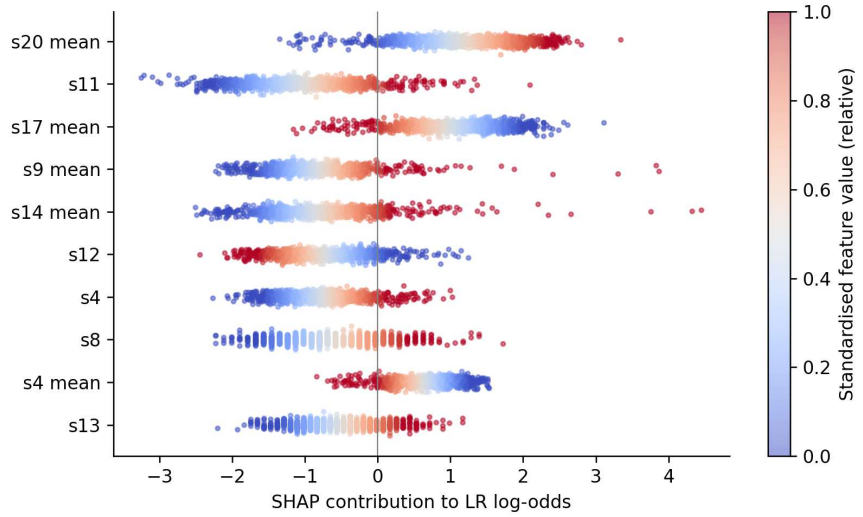

Fig. 2 Signed SHAP contributions for the seed-42 FD001 LR fit on a fixed sample of 1,000 test rows, using 1,000 balanced augmented fitting-background rows and an independent-feature masker. Colours show relative standardised feature values. Feature labels abbreviate sensors as s and rolling statistics as mean or SD. Attributions are in log-odds units.

subsets despite changing each engine's contribution to pooled F1. By contrast, reducing the common endpoint grid to stride 20 reverses two model pairs in each subset and changes the FD001 numerical leader. Final-cycle scoring reverses three eligible pairs on FD001 and six on FD002; the FD002 numerical leader changes from HGB to MLP. Because the scored observations and positive prevalence also change, these reversals do not by themselves isolate a causal effect of sampling, demonstrate statistically significant model differences, or justify using the highest-scoring population to choose a model. The native-population ranking is less comparable still because the LSTM and tabular methods score different observation sets.

Validation effects are model- and subset-specific. On FD001, row-wise validation raises HGB's best development F1 by 7.46 percentage points but leaves its selected configuration and official-test F1 unchanged. LR's corresponding increase is only 0.63 points, and effects on FD002 are smaller. Global preprocessing before splitting also does not uniformly inflate the measured validation F1. Accordingly, the complete development–test gap cannot be interpreted as leakage inflation, especially because complete training trajectories and truncated official-test histories have different label and degradation-stage distributions. These diagnostics quantify specified changes in the development task; they do not identify a universal leakage penalty.

Historical representation provides a separate source of conclusion sensitivity. The FD001 40-cycle LR pipeline achieves 87.40% mean F1 against 80.07% for the fixed 20-cycle reference, with a pointwise paired 95% interval for the difference of [2.87, 12.83] percentage points. However, the 40-cycle variant was reported after inspecting benchmark outcomes, and the temporal comparator was not jointly redeveloped for alternative history lengths. It would therefore be inappropriate to replace the main

LR specification retrospectively or claim LR superiority to LSTM based on those point estimates. A stronger confirmation would choose representation length entirely within grouped training folds and evaluate it in a held-out or independent test.

The development budget is transparent rather than fully equivalent: four configurations per model family do not equalise search-space coverage, optimisation cost, or the information available to temporal and tabular classifiers. The additional FD002 MLP convergence diagnostic shows that increasing its iteration cap to 1,000 removes warnings without raising mean common-timestamp F1 at the fixed selected hyperparameters. Paired engine-level inference also restricts claims: the adjusted LR-reference analysis does not establish LSTM–LR or HGB–LR superiority on either subset. Neither this non-significance nor descriptive rank stability demonstrates equivalence.

Finally, signed logistic-regression coefficients and stable SHAP magnitude rankings describe the fitted prediction function rather than a physical degradation mechanism. Correlated features, explanation-background choice, and differences between simulated and real industrial sensors limit interpretability. Model F1, rank reversals, and attribution stability cannot establish calibrated failure risk, reduced downtime, or useful maintenance intervention decisions.

5.1 Limitations and Future Work

The analysis is confined to NASA C-MAPSS FD001 and FD002. FD003, FD004, independent industrial fleets, maintenance interventions, and condition-specific normalisation for FD002 have not been evaluated. The standard training partitions are fixed, five fitting seeds do not resample fleets, and engine bootstrap intervals condition on the fitted models. The four-candidate search is narrow, the LSTM sequence length is fixed, and feature-history sensitivity is limited to LR on FD001. Shared feature filtering and augmentation may favour different model families; identical candidate counts are not a substitute for matched development quality.

The new ranking-reversal fractions are post hoc descriptive summaries of published mean F1 values, not a statistical test or a novel estimator. They exclude ties, use rounded tabular results, and reflect changes in case mix and prevalence across evaluation populations. No paired significance tests were performed for the full set of model-pair/rule contrasts, and comparing a ranking across populations does not hold the generalisation task constant. The existing LR-reference inference addresses a different family of comparisons. Further work should use complete per-engine prediction outputs to evaluate uncertainty in ranking stability and should define the intended prediction population before inspecting test results.

Additional priorities are nested, training-only selection of historical representation and model configurations, extension to FD003/FD004 and independent datasets, stronger sequence baselines, operating-condition-specific preprocessing, probability calibration, and explicit cost-sensitive maintenance-policy evaluation. The expanded analyses followed prior benchmark inspection and should therefore be considered exploratory rather than prospectively blinded confirmation.

6 Conclusion

This study examines the stability of predictive-maintenance benchmark conclusions under changes in validation design, evaluation population, and historical feature representation. A descriptive model-pair reversal analysis shows that numerical rankings can change when models are scored at final cycles or on a different common sampling grid, while equal-engine weighting preserves the original orderings in the evaluated subsets. Controlled validation experiments further show that a large increase in row-wise development F1 need not lead to an official-test improvement. Sensitivity to the rolling-feature history reinforces the need to distinguish representation design from model-family superiority.

The resulting contribution is a quantified and qualified analysis of protocol-dependent benchmark interpretations, not a new predictive architecture or an established universal model hierarchy. Broader datasets, training-only representation selection, ranking-uncertainty assessment, calibration, and maintenance-decision studies are needed before operational claims can be supported.

A Selected Configurations and Fit Diagnostics

Table 10 lists the configurations selected using the training-only grouped development procedure.

Table 10 Configurations selected from training-only grouped folds. All final seeds use the selected configuration within each subset.

Model	FD001	FD002
LR	$C = 0.1$	$C = 10$
SVM	$C = 0.01$	$C = 1$
KNN	$k = 7$, uniform	$k = 3$, distance
DT	depth 10, leaf 1	depth None, leaf 10
MLP	$\alpha = 0.01$, lr 0.003	$\alpha = 0.01$, lr 0.001
HGB	leaves 7, $L_2 = 1$	leaves 15, $L_2 = 1$
LSTM	dropout 0.3, lr 0.001, epochs 19	dropout 0, lr 0.001, epochs 5

For FD001 MLP, the selected candidate produced convergence warnings in 0 of three tuning fits; warnings occurred in 0 of five final fits. Final iteration counts ranged from 49 to 96. When warnings occur, the optimisation stopping criterion was not met within the stated budget, limiting conclusions about attainable model performance. Full loss and iteration diagnostics are retained in the supplementary archive.

For FD002 MLP, the selected candidate produced convergence warnings in 1 of three tuning fits; warnings occurred in 2 of five final fits. Final iteration counts ranged from 256 to 300. When warnings occur, the optimisation stopping criterion was not met within the stated budget, limiting conclusions about attainable model performance. Full loss and iteration diagnostics are retained in the supplementary archive.

A.1 Extended-Iteration MLP Sensitivity

All five extended fits met the optimiser stopping criterion before the 1,000-iteration cap; extended iteration counts range from 256 to 328. The number of convergence warnings changes from 2 in the main fits to 0 in the extended fits. Common-population mean F1 changes from 61.81% to 61.37%, a difference of -0.44 percentage points with pointwise paired 95% interval [-1.13, 0.27].

The three baseline fits that already stopped without warnings reproduce their class predictions exactly under the larger cap, with scores agreeing to absolute tolerance 10^{-12}. Changes therefore concern the two previously capped fits. The check uses fresh initialisation with the same seed for each paired fit, rather than warm-starting a trained optimiser. The extended results are diagnostic and do not replace the original budget-defined main comparison; satisfying a stopping criterion does not prove a global optimum.

Seed-specific iteration counts and F1 results for the FD002 MLP iteration-cap diagnostic are given in Table 11.

Table 11 FD002 MLP budget sensitivity at fixed training-selected hyperparameters. An asterisk marks an iteration-cap convergence warning. F1 uses the same 2,501 common timestamps and is expressed as a percentage.

Seed	Iterations: cap 300	Iterations: cap 1,000	F1: cap 300	F1: cap 1,000
42	300*	328	66.01	62.24
43	269	269	62.67	62.67
44	268	268	58.12	58.12
45	256	256	62.78	62.78
46	300*	301	59.47	61.02

Table 12 compares FD002 MLP performance across evaluation populations under the two iteration caps.

Table 12 FD002 MLP F1 across populations under the two iteration caps. Percentages are mean ± sample SD over the same five seeds.

Population	N	Cap 300	Cap 1,000
Common endpoints	2,501	61.81 ± 3.10	61.37 ± 1.95
All native rows	33,991	54.45 ± 3.04	53.83 ± 2.20
Final cycles	259	83.18 ± 1.98	82.96 ± 1.30

B Model-Level Sampling Sensitivity

Each row reports one model and evaluation rule. Counts and positive percentages refer to observed, unweighted endpoint rows; all models use identical labelled endpoints within a rule. Main and stride-20 F1 pool those observations. Equal-engine F1 instead

assigns each engine equal total weight, retaining the main endpoint count. Numerical ranks use unrounded mean F1; exact ties share a rank and the following ranks are skipped. These secondary ranks have not been subjected to additional pairwise tests.

The complete FD001 model-level sampling results appear in Table 13.

Table 13 FD001 sampling sensitivity. F1 is mean ± SD over five seeds (percentages); Pos. is the raw positive count. Equal-engine weighting has weighted positive prevalence 4.15%, while Pos. (%) reports raw prevalence.

Rule	Model	N	Pos.	Pos. (%)	F1 ± SD	Rank
Main	LSTM	962	55	5.72	83.36 ± 3.91	1
Main	LR	962	55	5.72	80.07 ± 0.36	2
Main	SVM	962	55	5.72	79.81 ± 0.29	3
Main	HGB	962	55	5.72	78.57 ± 1.11	4
Main	KNN	962	55	5.72	72.88 ± 1.10	5
Main	MLP	962	55	5.72	72.10 ± 1.07	6
Main	DT	962	55	5.72	70.15 ± 2.36	7
Stride 20	LR	557	37	6.64	84.34 ± 0.00	1
Stride 20	SVM	557	37	6.64	84.34 ± 0.00	1
Stride 20	LSTM	557	37	6.64	83.43 ± 4.27	3
Stride 20	HGB	557	37	6.64	81.75 ± 1.43	4
Stride 20	KNN	557	37	6.64	76.32 ± 1.45	5
Stride 20	MLP	557	37	6.64	75.48 ± 2.21	6
Stride 20	DT	557	37	6.64	75.09 ± 4.66	7
Equal engine	LSTM	962	55	5.72	83.42 ± 4.19	1
Equal engine	LR	962	55	5.72	78.69 ± 0.23	2
Equal engine	SVM	962	55	5.72	78.52 ± 0.19	3
Equal engine	HGB	962	55	5.72	76.87 ± 1.46	4
Equal engine	KNN	962	55	5.72	71.54 ± 1.54	5
Equal engine	MLP	962	55	5.72	69.69 ± 1.36	6
Equal engine	DT	962	55	5.72	68.06 ± 2.49	7

The corresponding FD002 model-level sampling results appear in Table 14.

Table 14 FD002 sampling sensitivity. F1 is mean ± SD over five seeds (percentages); Pos. is the raw positive count. Equal-engine weighting has weighted positive prevalence 4.57%, while Pos. (%) reports raw prevalence.

Rule	Model	N	Pos.	Pos. (%)	F1 ± SD	Rank
Main	HGB	2,501	162	6.48	65.44 ± 0.62	1
Main	LSTM	2,501	162	6.48	63.67 ± 1.76	2
Main	LR	2,501	162	6.48	62.93 ± 0.11	3
Main	SVM	2,501	162	6.48	62.86 ± 0.15	4
Main	MLP	2,501	162	6.48	61.81 ± 3.10	5
Main	DT	2,501	162	6.48	54.66 ± 1.18	6
Main	KNN	2,501	162	6.48	54.64 ± 0.81	7
Stride 20	HGB	1,434	113	7.88	70.61 ± 0.88	1
Stride 20	LSTM	1,434	113	7.88	69.20 ± 1.73	2
Stride 20	MLP	1,434	113	7.88	68.18 ± 2.98	3
Stride 20	LR	1,434	113	7.88	67.98 ± 0.22	4
Stride 20	SVM	1,434	113	7.88	67.84 ± 0.40	5
Stride 20	DT	1,434	113	7.88	60.98 ± 0.60	6
Stride 20	KNN	1,434	113	7.88	60.92 ± 0.67	7
Equal engine	HGB	2,501	162	6.48	66.99 ± 0.74	1
Equal engine	LSTM	2,501	162	6.48	64.81 ± 1.85	2
Equal engine	LR	2,501	162	6.48	64.73 ± 0.21	3
Equal engine	SVM	2,501	162	6.48	64.71 ± 0.16	4
Equal engine	MLP	2,501	162	6.48	62.22 ± 3.11	5
Equal engine	DT	2,501	162	6.48	52.59 ± 3.22	6
Equal engine	KNN	2,501	162	6.48	51.07 ± 0.87	7

Statements and Declarations

- **Funding:** This research did not receive external funding.
- **Competing interests:** The authors declare no competing interests.
- **Ethics approval and consent to participate:** Not applicable; the study uses a simulated equipment benchmark and involves no human participants or animals.
- **Consent for publication:** Not applicable.

  Data availability. The NASA C-MAPSS benchmark datasets (FD001 and FD002) used in this study are publicly available through the NASA Prognostics Data Repository at https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/.
- **Code availability:** The supplementary reproducibility archive contains experiment and analysis scripts, all tested configurations, per-engine/cycle predictions, fit diagnostics, statistical outputs, software versions, data checksums, and figure-generation code. These files are supplied as supplementary material with the manuscript.
- **Author contributions:** W.A.Z: Conceptualisation, Methodology, and Investigation . T.A: Methodology, Data Curation, Formal Analysis, Supervision, and Writing—Review and Editing. O.A., and N.A.H: Supervision, Writing—Review. All authors read and approved the final manuscript.

References

Angelov PP, Soares EA, Jiang R, et al (2021) Explainable artificial intelligence: An analytical review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11(5):e1424. https://doi.org/10.1002/widm.1424

Breiman L, Friedman JH, Olshen RA, et al (1984) Classification and Regression Trees. Wadsworth International Group, Belmont, CA, USA

Cawley GC, Talbot NLC (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research 11:2079–2107. URL https://jmlr.org/papers/v11/cawley10a.html

Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20(3):273–297. https://doi.org/10.1007/BF00994018

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap. Chapman & Hall, New York, NY, USA, https://doi.org/10.1007/978-1-4899-4541-9

Fink O, Wang Q, Svensén M, et al (2020) Potential, challenges and future directions for deep learning in prognostics and health management applications. Engineering Applications of Artificial Intelligence 92:103678. https://doi.org/10.1016/j.engappai.2020.103678

Heimes FO (2008) Recurrent neural networks for remaining useful life estimation. In: 2008 International Conference on Prognostics and Health Management. IEEE, pp 1–6, https://doi.org/10.1109/PHM.2008.4711422

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Computation 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6(2):65–70. URL https://www.jstor.org/stable/4615733

Hosmer DWJr., Lemeshow S, Sturdivant RX (2013) Applied Logistic Regression, 3rd edn. John Wiley & Sons, Hoboken, NJ, USA, https://doi.org/10.1002/9781118548387

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, vol 30. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Özcan H (2026) Predictive maintenance in aircraft engine maintenance using the C-MAPSS dataset: performance comparison and evaluation of machine learning classification algorithms. AI EDAM 40:e4. https://doi.org/10.1017/S0890060426100249

Phipson B, Smyth GK (2010) Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn. Statistical Applications in Genetics and Molecular Biology 9(1). https://doi.org/10.2202/1544-6115.1585, article 39

Roberts DR, Bahn V, Ciuti S, et al (2017) Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40(8):913–929. https://doi.org/10.1111/ecog.02881

Saxena A, Goebel K (2008) Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository, Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/

Saxena A, Goebel K, Simon D, et al (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management. IEEE, URL https://ieeexplore.ieee.org/document/4711414

scikit-learn developers (2026) HistGradientBoostingClassifier: implementation documentation. URL https://scikit-learn.org/1.8/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html, implementation used: scikit-learn 1.8.0

Serradilla O, Zugasti E, Rodriguez J, et al (2022) Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. Applied Intelligence 52:10934–10964. https://doi.org/10.1007/s10489-021-03004-y

Zhao R, Yan R, Chen Z, et al (2019) Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing 115:213–237. https://doi.org/10.1016/j.ymssp.2018.05.050