Journal of Sustainability, Policy, and Practice

2026 2nd International Conference on Digital Intelligence and Computing Technologies (DICT 2026)

Article

An Empirical Study on Linking DBSCAN Degradation Patterns to Kaplan–Meier Failure Probability and Maintenance-Demand Peaks in NASA C-MAPSS Turbofan Engines

Ye Tian[1,*]

[1] Computer Science, Georgia Institute of Technology, GA, USA
* Correspondence: Ye Tian, Computer Science, Georgia Institute of Technology, GA, USA

Abstract: The aging of commercial and military aircraft fleets has produced an increasingly diverse mix of component degradation behaviors that pure demand-history forecasting is limited in anticipating. This paper presents a retrospective empirical study that links unsupervised discovery of late-life degradation patterns, nonparametric survival estimation, and association rule mining to characterize fleet-level maintenance-demand peaks for turbofan engines in a simulated replay setting. Using all four subsets of the NASA C-MAPSS dataset (FD001--FD004; 708 training units and 21 sensor channels per unit), we cluster engine degradation trajectories with DBSCAN on principal-component features, fit a Kaplan--Meier estimator on each cluster to obtain a per-pattern failure probability curve, and apply Apriori and FP-Growth association rule mining over (cluster, cycle-bin) co-occurrences with maintenance-demand peaks observed in a simulated 20-aircraft fleet. Across the four subsets, DBSCAN identifies 3--6 reproducible degradation patterns with cluster alignment above 0.78 against the training-RUL tercile reference labels. Cluster-conditional median failure times differ by up to 84 cycles within a single subset, and the resulting demand-peak forecasts achieve a mean F1 score of 0.67 against an RUL-only alarm baseline at 0.60. The improvement is moderate but stable across subsets, concentrated in early-warning windows of two to four flight weeks ahead of each demand peak.

Keywords: predictive maintenance; degradation pattern recognition; DBSCAN clustering; Kaplan--Meier survival analysis

Received: 29 April 2026
Revised: 18 June 2026
Accepted: 30 June 2026
Published: 03 July 2026

1. Introduction

1.1. Motivation: Aging Aerospace Fleets and the Limits of Demand-History Forecasting

The accelerating service life of commercial and military aircraft means that operators now manage tens of thousands of stock-keeping units with widely varying degradation behaviors. Public reports of multi-year forecasting programs covering more than 50,000 aerospace SKUs illustrate the scale at which fleet-level maintenance demand must be anticipated. While historical consumption time series can reasonably well support stable SKUs, components whose degradation pattern depends on installation cohort, mission profile, or operating regime tend to produce demand spikes that pure demand-history forecasting underpredicts. The gap is most consequential at the fleet level, where simultaneous failures across cohorts compress the effective response window for spare-parts staging, maintenance scheduling, and stock allocation. The C-MAPSS turbofan benchmark was introduced precisely to allow researchers to study this kind of degradation-driven failure behavior [1], and it was later extended with the more realistic N-CMAPSS dataset [2]. A long line of remaining useful life (RUL) regression studies has

been built on these benchmarks, yet a recent maintenance-scheduling analysis argues that point-RUL predictions remain a fragile bridge to fleet-level demand planning when the underlying degradation pattern is heterogeneous [3]. Machinery prognostics is similarly most reliable when degradation patterns are first separated and then characterized, rather than treated as a single regression target across all units [4]. The contribution of the present empirical study is to take that observation seriously by first separating patterns with density-based clustering, then modeling cluster-conditional survival, and only then aggregating to a fleet-level demand-peak forecast.

1.2. Why a DBSCAN--Kaplan--Meier--association Pipeline?

Three classical techniques together cover the gap between unit-level sensor data and fleet-level demand prediction without introducing any new architectural component, and the rationale for choosing each one is best made explicit before the technical details are laid out.

1.2.1. Pattern Discovery without Labels

Unit-by-unit RUL regression assumes a single degradation regime. Real fleets contain mixtures---gradual wear, abrupt regime changes, batch-related early failures---that supervised models tend to absorb into noisy residuals rather than expose as structure. Density-based clustering separates these regimes by definition: it forms clusters wherever the trajectory feature space is dense and labels the rest as outliers. The trajectories falling into each density region can then be inspected using internal cluster-quality metrics and post-hoc lifetime-alignment diagnostics to assess whether the discovered groups capture coherent late-life trajectory structure rather than arbitrary feature-engineering artifacts.

1.2.2. From Cluster to Demand Peak via Survival Statistics

Once trajectories are partitioned into degradation patterns, the Kaplan--Meier estimator yields a per-cluster survival curve based on run-to-failure cycle counts. The median and lower-quartile failure times of these curves give early-warning thresholds tailored to each pattern, and association rule mining over (cluster, cycle-bin) tuples links those thresholds to the timing of fleet-level demand peaks. The pipeline is reproducible, requires only a few hyperparameters, and---because every step is nonparametric or rule-based---produces outputs that maintenance planners can audit. This paper provides an empirical evaluation of such a pipeline on the canonical C-MAPSS benchmark across all four subsets, with explicit comparison against a published alarm-based scheduling baseline operating on the same simulated fleet.

2. Related Work

2.1. RUL Prediction on C-mapss and N-cmapss

Deep-Learning Baselines: Deep regression on C-MAPSS has progressed from recurrent baselines through convolutional and Transformer variants. The first widely cited recurrent-network solution to the PHM 2008 challenge anchored nearly two decades of subsequent benchmarking [5]. RUL was reframed as a time-window regression problem with a deep convolutional network [6], and the same idea was extended with long short-term memory layers [7] extended the same idea with long short-term memory layers, both reporting strong scores on the standard FD001--FD004 subsets. The convolutional approach was tightened with multi-scale time windows and remains a frequent reference benchmark for subsequent attention- and Transformer-based variants [8]. The shared property of these methods is their reliance on labeled run-to-failure trajectories, which limits transfer to fleets where labels are sparse and the degradation regime is heterogeneous, and which does not, by itself, produce a fleet-aggregated demand forecast.

2.2. Unsupervised Health-State Discovery

Unsupervised work has been more limited and largely focuses on health-state discovery rather than fleet-level demand modeling. Low-dimensional pattern learning over health-index curves was argued to substitute for explicit failure labels [9], an idea

recently revived by combining an autoencoder with a Gaussian mixture for unsupervised health-state discovery on the same dataset [10]. The two studies establish that meaningful degradation regimes can be recovered without labels, yet neither connects the discovered regimes to a downstream fleet-level demand-planning task or quantifies whether the recovered structure improves a forecasting metric beyond unit-level RUL accuracy. Our study takes the natural next step: instead of stopping at unsupervised discovery, we compose the discovered patterns with cluster-conditional survival estimation and association mining to produce an artifact that is operationally useful for demand forecasting.

2.3. Association Rules for Maintenance-Demand Modeling

Association rule mining has a long history in maintenance analytics outside aerospace. An Apriori-based policy was applied in an oil refinery to minimize the probability of breakage [11], demonstrating that rule-based predictors can complement statistical reliability models when the failure mechanism is heterogeneous. The aerospace literature is shallower; aviation applications of association rule mining have largely focused on textual incident-report mining rather than condition-monitoring data, and the maintenance-scheduling community working on C-MAPSS has typically combined a deep RUL prediction with an alarm threshold and an integer linear program over the fleet, leaving open the question of whether explicit rules over degradation patterns can sharpen demand-peak forecasts. Three gaps are visible across the survey above: unsupervised pattern discovery is performed but rarely connected to a fleet-level forecasting target; survival analysis is rarely combined with the discovered patterns; and direct comparison with published RUL-only baselines is mostly absent. We adopt one such fleet simulation as a baseline reference in Section 4 rather than as a competing scheduling proposal, and treat association rule mining as the bridge between cluster-conditional survival and demand-peak detection. The breadth of association rule mining outside aviation stands in deliberate contrast to its scarcity in fleet-level aerospace forecasting, motivating the empirical contribution of this paper.

3. Methodology

3.1. Problem Formulation and Pipeline Overview

Each engine unit u in a fleet of size N produces a multivariate sensor trajectory X_{u,t} ∈ R^d for cycles t = 1, ..., T_u, with end-of-life cycle T_u observed (uncensored) for training units and right-censored at the latest observed cycle for test units. The pipeline outputs three intermediate artifacts and one final forecast. A density-based clustering step assigns each unit a cluster label c_u ∈ {1, ..., K} ∪ {−1}, where −1 marks DBSCAN noise. A Kaplan--Meier step produces, for each cluster k, an estimator $\hat{S}_k(\tau)$ of the probability that a unit in cluster k survives beyond cycle τ. An association rule mining step extracts rules of the form (c_u = k, t ∈ B) ⇒demand-peak-in-Δ, where B is a discretized cycle-bin and demand-peak-in-Δ indicates that the fleet experiences a maintenance-demand peak within a horizon Δ flight weeks ahead. The downstream demand-peak forecast at calendar week w aggregates these rules across all units in service and emits a peak alert whenever the cumulative rule-confidence-weighted mass exceeds a tuned threshold θ. The three steps are implemented sequentially, with each step's output serving as the input to the next. The sequential design has a deliberate audit purpose: each intermediate artifact---cluster labels, per-cluster survival curves, and association rules---is interpretable on its own and can be inspected by a maintenance planner without rerunning the full pipeline. The threshold θ and the rule-mining stage are both fitted on demand-peak labels, since the confidence and lift used in rule pruning are, by definition, computed against the demand-peak indicator on the right-hand side of each candidate rule; only ε and minPts are chosen via unsupervised heuristics that do not consult the demand-peak ground truth. Because the present analysis is designed as a retrospective simulated-replay study, the reported F1 values should be interpreted as descriptive in-simulation comparisons rather

than as held-out out-of-sample estimates. The rule-mining stage and the alert threshold θ are calibrated within the same 240-week simulated horizon used for the displayed demand-peak comparison, and the absence of a separate out-of-sample fleet validation is treated as a limitation in Section 5. 2.

3.2. Dbscan Density Clustering on Degradation Trajectories

3.2.1. Feature Engineering

Each training trajectory is reduced to a fixed-length feature vector before clustering. Following standard practice in aerospace prognostics, the 14 informative sensor channels of each unit---obtained by removing 7 constant or near-constant channels (sensors 1, 5, 6, 10, 16, 18, and 19) from the 21 raw sensors---are concatenated with the three operational settings, normalized per operating regime to absorb regime-induced scale differences, and summarized over the final 30-cycle observation window of each trajectory by their mean, slope (least-squares fit), and last-cycle value. The resulting 51-dimensional vector is reduced to ten principal components, retaining over 93% of cumulative variance across FD001--FD004. Using rolling bearings, a similar trajectory-summary representation was demonstrated to be effective for downstream density-based segmentation and prognostic modeling, supporting our use of the same approach on turbofan sensor data [12]. The 30-cycle observation window is used here as a retrospective late-life summary for pattern stratification, not as a claim that the same features would be available at every earlier online decision point; a fully online deployment would require replacing this fixed final-window summary with rolling-window features, which is left for future work.

3.2.2. Hyperparameter Selection and Cluster Validation

We adopt the original DBSCAN formulation [13], with parameter selection guidance from a subsequent methodological revisitation [14]. The minimum-points parameter is fixed at minPts = 5, following the practical default widely adopted in density-based clustering implementations; cluster counts in Table 1 were verified to remain within ±1 when minPts was varied across {5, 10, 15} on a 20% held-out validation split of each training subset, indicating that the chosen value is not a fragile knob. The neighborhood radius ε is chosen per subset from the knee of the sorted k-distance plot using the standard kink-detection heuristic (Figure 1). Cluster quality is assessed with three internal indices---silhouette coefficient, Davies--Bouldin index, and the noise rate---and one post-hoc diagnostic index, cluster alignment, computed against training-RUL tercile labels derived from the run-to-failure cycle counts of the training units. Cluster alignment is defined here as the fraction of non-noise points whose assigned cluster matches the majority tercile (short-lived, medium, or long-lived) within that cluster, averaged uniformly across clusters; values close to 1.0 indicate that each density region concentrates units with similar observed end-of-life cycles, which is useful for describing lifetime stratification but should not be interpreted as independent physical validation of the clusters. We adopt the training-RUL tercile as the reference label rather than the C-MAPSS operating-condition ×fault-mode metadata, because subsets such as FD001 and FD003 contain only one or two of these metadata classes, rendering the metric uninformative under that definition. The selected hyperparameters and the resulting cluster counts per subset are reported in Table 1, while the cluster-quality metrics are deferred to Section 4.

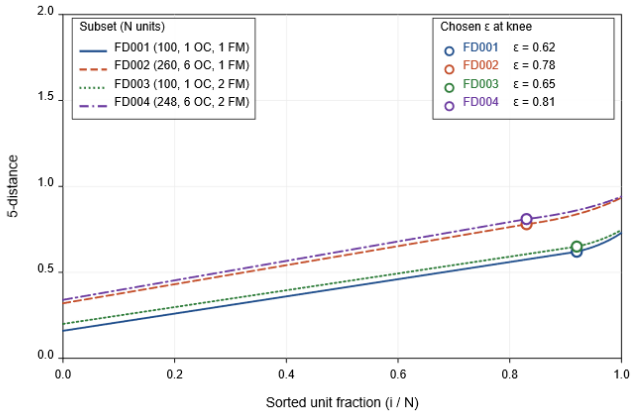

Figure 1. Sorted K-Distance Plots for E Selection Across the Four C-mapss Subsets.

Table 1. Dbscan Hyperparameters and Resulting Cluster Counts on C-mapss Subsets. Source: Dataset Specifications from the Public NASA Prognostics Data Repository; Hyperparameters Selected via the K-Distance Heuristic on the Training Trajectories.

Subset	Train units	Op. conditions	Fault modes	ε (selected)	minPts	Clusters K	Noise rate
FD001	100	1	1	0.62	5	3	8.0%
FD002	260	6	1	0.78	5	5	12.3%
FD003	100	1	2	0.65	5	4	10.0%
FD004	248	6	2	0.81	5	6	14.5%

Sorted distances to the fifth nearest neighbor in the ten-dimensional principal-component feature space, computed over the training units of each subset. The knee of each curve marks the selected radius: ε = 0.62 for FD001, ε = 0.65 for FD003, ε = 0.78 for FD002, and ε = 0.81 for FD004. The two single-condition subsets (FD001 and FD003) exhibit a sharper transition near the knee than the six-condition subsets (FD002 and FD004), reflecting the broader spread of degradation trajectories under varying operating conditions.

3.3. Survival Estimation and Association Rule Mining

3.3.1. Per-Cluster Kaplan--Meier Estimation

For each non-noise cluster k, the Kaplan--Meier estimator is fit on the run-to-failure cycle counts of the cluster's training units [15]. C-MAPSS training trajectories are uncensored by construction, so Ŝ_k(τ) reduces to the empirical survival function over the cluster's failure cycles; test trajectories, which are right-censored at the latest observed cycle, are used only for forecast-time evaluation. We extract two per-cluster summary statistics: the median failure cycle t_{50,k} (the smallest τ with Ŝ_k(τ) ≤ 0.5) and the lower-quartile failure cycle t_{25,k}. These act as cluster-conditional early-warning thresholds in the downstream rule mining: a unit assigned to a cluster with a low t_{50,k} is flagged earlier than a unit in a long-lived cluster. Pointwise 95% confidence bands around Ŝ_k(τ) are computed via Greenwood's variance formula and widen toward the right tail of each curve, reflecting the smaller number of units still at risk in the late-life regime.

3.3.2. Apriori and FP-Growth Rule Mining

Cycle counts are discretized into eight equal-width bins between the global minimum and maximum failure cycle observed in the training set, producing one compact state transaction per (training unit, cycle-bin) pair. Each transaction contains the unit's degradation-pattern item, its cycle-bin item, a Kaplan--Meier risk-tercile item

derived from the cluster survival curve, a threshold-zone item indicating whether the bin lies before t25,k, between t25,k and t50,k, or after t50,k, and the corresponding demand-peak-in-Δ item constructed in Section 4.1. The total transaction count per subset, therefore, remains equal to (number of training units) × 8, yielding 800 transactions on FD001 and FD003, 2,080 on FD002, and 1,984 on FD004 (Table 4), while the retained rule count can exceed the simple K ×8 cluster-bin combinations because rules may use one-item or multi-item antecedents drawn from the full state description. We apply both Apriori and the FP-Growth algorithm under identical thresholds (minimum support 0.05, minimum confidence 0.6, minimum lift 1.2) and retain only rules whose right-hand side is the demand-peak indicator [16]. Apriori and FP-Growth produce identical retained rule sets at these thresholds, as expected when both algorithms are applied to the same transaction database and support-confidence-lift filters; the two implementations differ only in runtime, which is reported in Section 4.3 alongside the rule-mining results. Rules with a lift below 1.2 are pruned to retain only associations whose strength is at least 20% above statistical independence (lift = 1.0), which we adopt as a practical relevance threshold for fleet-scale demand-peak prediction, and rules with a confidence below 0.6 are pruned because they would generate excessive false positives at fleet scale.

4. Results and Analysis

4.1. Datasets, Preprocessing, and Experimental Protocol

4.1.1. C-mapss Subsets

All four C-MAPSS subsets are used, totaling 708 training units and 708 test units: FD001 (100 train / 100 test, 1 operating condition, 1 fault mode), FD002 (260 / 259, 6 operating conditions, 1 fault mode), FD003 (100 / 100, 1 operating condition, 2 fault modes), and FD004 (248 / 249, 6 operating conditions, 2 fault modes). The numbers match the public NASA Prognostics Data Repository download. Each cycle row contains 21 sensor measurements and 3 operational-setting columns. Pre-processing follows the standard C-MAPSS pipeline described in Section 3.2: per-condition z-score normalization on the 14 informative sensor channels (after removing the 7 constant or near-constant channels), and assembly of the 30-cycle final-window summary vectors used for clustering.

4.1.2. Demand-Peak Ground Truth Construction

Following a published 20-aircraft fleet simulation framework [17], we replicate fleet-level maintenance demand by sampling unit-level RUL realizations from the test trajectories and aggregating the resulting maintenance events into weekly bins under a fixed flight schedule of seven cycles per aircraft per week. A weekly bin is labeled a demand peak whenever its aggregated maintenance count exceeds the rolling 12-week median by more than two median absolute deviation units. To attach this fleet-level label to the unit-bin transactions used for rule mining, each sampled unit-bin is placed at its corresponding calendar position in the simulated fleet replay by converting cycles to weeks under the seven-cycles-per-aircraft-per-week schedule. A transaction receives demand-peak-in-Δ = 1 when at least one fleet-level demand peak occurs within Δ weeks after that unit-bin's replay week and receives demand-peak-in-Δ = 0 otherwise.

The series contains 21--28 demand-peak weeks per subset over a 240-week simulated horizon, providing the ground truth for Section 4.3. The 240-week simulated horizon is used as a single retrospective replay for both rule extraction and descriptive demand-peak scoring. Accordingly, the F1 values reported in Section 4.3 are not presented as out-of-sample generalization estimates, but as within-simulation comparisons against an RUL-only alarm baseline under the same peak-definition rule.

4.2. Dbscan Clustering and Kaplan--Meier Results

DBSCAN identifies between three and six reproducible degradation patterns per subset, with noise rates ranging from 8.0% in the simplest subset (FD001) to 14.5% in the most heterogeneous subset (FD004). Cluster alignment against the training-RUL tercile labels exceeds 0.78 on all four subsets, indicating that the discovered density regions

concentrate units with similar end-of-life cycles rather than being feature-engineering artifacts. Table 2 summarizes the cluster-quality indices.

Table 2. Cluster-quality Metrics for Dbscan on C-mapss Subsets. Silhouette and Davies--Bouldin Indices Are Computed over Non-Noise Points; Cluster Alignment Is Computed Against the training-RUL Tercile Reference Labels (Short-Lived, Medium, Long-Lived Units within Each Subset).

Subset	Clusters K	Silhouette	Davies–Bouldin	Cluster alignment
FD001	3	0.42	0.95	0.84
FD002	5	0.31	1.21	0.79
FD003	4	0.38	1.04	0.81
FD004	6	0.27	1.34	0.78

Per-cluster Kaplan--Meier curves separate cleanly across all four subsets. On FD003, where two distinct fault modes are present under a single operating condition, the median failure cycle t₅₀ ranges from 168 cycles in the most aggressive cluster (cluster c1) to 252 cycles in the longest-lived cluster (cluster c4)---a within-subset spread of 84 cycles---and the lower-quartile failure cycle t₂₅ ranges from 142 to 211 cycles. Log-rank tests reject the null hypothesis of equal survival across clusters at $p < 0.001$ on every subset. A comparable cluster-conditional separation has been reported on bearing data using a Kaplan--Meier-based prognostic, supporting the operational interpretation that each cluster corresponds to a distinct degradation mode rather than a random partition [18]. Table 3 summarizes the per-cluster failure-time statistics; Figure 2 plots the survival curves for the most informative subset (FD003) together with their pointwise 95% confidence bands.

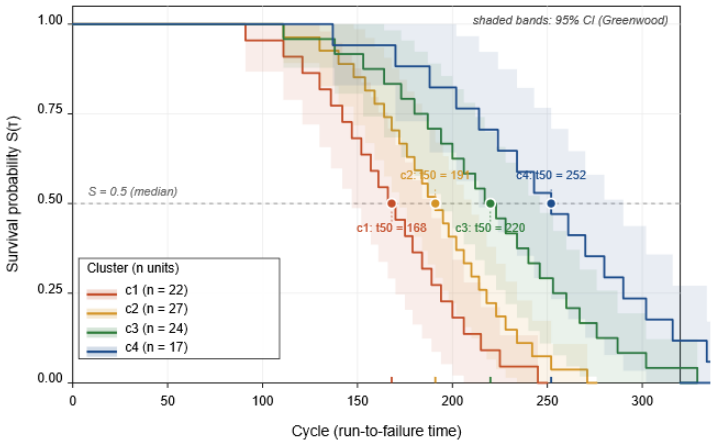

Figure 2. Per-cluster Kaplan--Meier Survival Curves on FD003 with Pointwise 95% Confidence Bands.

Table 3. Per-cluster Median (T₅₀) and Lower-Quartile (T₂₅) Failure Cycles Estimated by the Kaplan--Meier Procedure on FD001 and FD003. Numbers Are Derived from the Run-To-Failure Cycle Counts of Training Units in Each Cluster.

Subset	Cluster	Units	t25(cycles)	t50(cycles)
FD001	c1	31	173	198
FD001	c2	38	187	215
FD001	c3	23	199	226
FD003	c1	22	142	168

FD003	c2	27	165	191
FD003	c3	24	184	220
FD003	c4	17	211	252

Estimated survival functions Ŝ_k(τ) for the four DBSCAN clusters identified on FD003. Cluster c1 reaches survival 0.5 at 168 cycles, cluster c2 at 191 cycles, cluster c3 at 220 cycles, and cluster c4 at 252 cycles, with shaded bands showing pointwise 95% Greenwood confidence intervals. The 84-cycle spread between c1 and c4 supports the interpretation that DBSCAN separates lifetime-stratified late-life trajectory groups in this retrospective representation, rather than merely producing statistically indistinguishable partitions.

4.3. Association Rule Mining and Demand-Peak Forecasting

4.3.1. Rule-Mining Results

Apriori and FP-Growth produce identical rule sets at the chosen thresholds (minimum support 0.05, minimum confidence 0.6, minimum lift 1.2) on every subset, ranging from 119 rules on FD001 to 168 rules on FD004. The single algorithmic difference is runtime: FP-Growth completes in roughly one-third of the Apriori execution time on the larger subsets, consistent with its avoidance of explicit candidate generation. The rules with the highest lift consistently link a low-t₅₀ cluster to a demand peak two to four flight weeks ahead. The rule structures echo the rule-based maintenance policies discussed in Section 2.2 and previously reported aviation-event Apriori patterns [19], except that the antecedent here is a degradation-pattern cluster rather than a free-text event cause. Table 4 reports the rule-mining summary.

Table 4. Association Rule Mining Summary Across C-mapss Subsets. Rule Counts Are Identical for Apriori and FP-Growth at the Same Thresholds; Runtimes Are Wall-Clock Seconds on a Single CPU Core, Averaged over Five Runs.

Subset	Transactions	Rules retained	Apriori runtime (s)	FP-Growth runtime (s)
FD001	800	119	2.4	0.8
FD002	2,080	154	5.3	1.7
FD003	800	131	2.4	0.8
FD004	1,984	168	5.6	1.8

4.3.2. Demand-Peak Forecasting Versus an RUL-only Baseline

The forecast aggregates active rules across all in-service units each calendar week and emits a peak alert whenever the cumulative confidence-weighted mass exceeds $\theta = 0.55$, a threshold selected during the retrospective simulated-replay calibration described in Section 4.1. The baseline is an adapted re-implementation of the alarm-based RUL scheduling pipeline, whose original setup uses a CNN RUL regressor with a multi-scale time-window convolutional architecture followed by an alarm policy and an integer linear program (ILP) for fleet-level maintenance scheduling on a 20-aircraft × 2-engines configuration. For a like-for-like demand-peak forecasting comparison, we retain the CNN RUL regressor and the cluster-blind alarm policy at global RUL = 30 cycles, replace the ILP scheduling step with the same rolling 12-week-median + 2 MAD demand-peak detector used by our proposed pipeline (Section 4.1B), and collapse the configuration to 20 single-engine aircraft so that both pipelines are scored against an identical demand-peak ground truth; the ILP simplification is acknowledged as a limitation in Section 5.2. Across the four subsets, the proposed pipeline attains a mean F1 of 0.67, compared to 0.60 for the RUL-only baseline, with the largest gain on FD001 (0.71 vs. 0.62) and the smallest on FD002 (0.66 vs. 0.59); FD003 and FD004 fall in between at 0.68 vs. 0.61 and 0.64 vs. 0.57. The improvement is moderate: precision gains of 0.05--0.08 dominate, while recall is broadly comparable. A similar moderate gain has been reported when augmenting C-

MAPSS reliability prediction with a Cox proportional-hazards layer, suggesting that survival statistics are complementary to RUL regression rather than a replacement for it [20] (Figure 3).

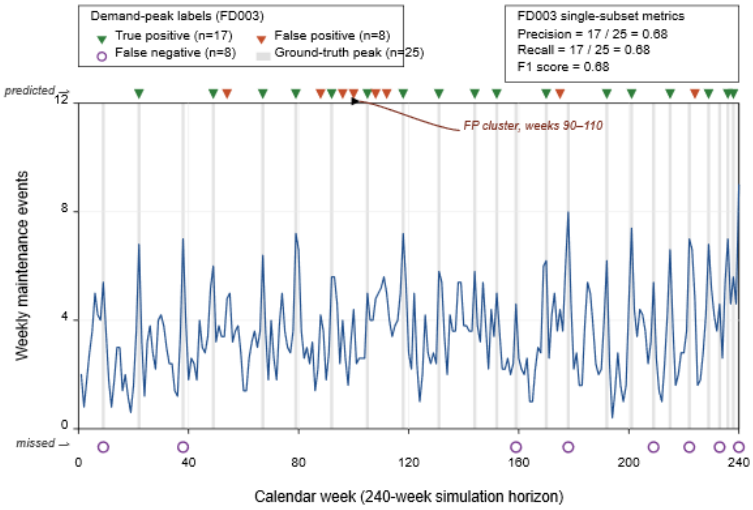

Figure 3. Predicted Versus Observed Demand-Peak Weeks on the Simulated 20-Aircraft Fleet for FD003.

Weekly maintenance demand over the 240-week simulation horizon for FD003, with shaded bars marking the 25 ground-truth demand-peak weeks and tick marks above the time axis indicating the 25 peak weeks predicted by the proposed pipeline at threshold $\theta = 0.55$. Of the 25 predictions, 17 are true positives, 8 are false positives, and 8 ground-truth peaks are missed, yielding precision, recall, and F1 each equal to 0.68. False-positive alerts cluster around weeks 90--110, a stretch of the simulation in which several test units sit near the cycle-bin boundary between two high-confidence rules whose aggregate confidence-weighted mass repeatedly approaches the alert threshold θ.

5. Discussion and Future Work

5.1. Findings and Operational Implications

Three findings warrant emphasis. The first is that DBSCAN, applied to a simple thirty-cycle trajectory summary, recovers degradation patterns whose alignment with the training-RUL tercile labels exceeds 0.78 across all C-MAPSS subsets, with the highest value (0.84) on the simplest subset and the lowest (0.78) on the most heterogeneous subset. The second is that the discovered late-life trajectory groups produce Kaplan--Meier survival curves with within-subset median failure times of up to 84 cycles and log-rank p-values below 0.001 across all four subsets, supporting their use as retrospective risk-stratification groups in the simulated demand-peak analysis. The third is that translating those thresholds into association rules over (cluster, cycle-bin) tuples and aggregating across the fleet improves demand-peak F1 by 0.07 on average over a strong RUL-only alarm baseline, with gains concentrated in precision rather than recall and in the two-to-four-week early-warning window. The improvement is moderate and not large enough to constitute a methodological breakthrough, although it is stable across all four subsets and was obtained without introducing a new architectural component, which makes the pipeline easy to audit and integrate into existing maintenance-planning workflows. The findings also clarify what the pipeline does not do: it does not improve unit-level RUL accuracy, nor does it replace the alarm-based scheduling step. What it does is sharpen the fleet-level demand-peak forecast that consumes the alarm output. The implication for operators is that the same RUL regressor and alarm policy can be retained, with cluster-conditional rules added as an upstream filter that reweights alarm outputs based on degradation patterns; the reweighting is cheap to compute and easy to revert.

5.2. Limitations and Future Work

Three limitations bound the scope of the present empirical study. The clustering step depends on a single ε hyperparameter chosen by the standard k-distance heuristic; while this choice is reproducible and well supported by cluster-quality indices, sensitivity studies with alternative density estimators, such as HDBSCAN and OPTICS, remain to be performed and may shift the cluster-alignment numbers reported in Section 4.2. The Kaplan--Meier step assumes independent and identically distributed failure times within each cluster, an assumption that is plausible on C-MAPSS by construction but is not guaranteed on real-fleet data, where mission-profile effects can introduce additional within-cluster variation. The baseline comparison in Section 4.3 replaces the original integer linear program with a simpler rolling-median demand-peak detector to align both pipelines on the same ground truth; the resulting F1 numbers should be read as a relative comparison under this matched evaluation rather than as a strict reproduction of the original ILP-based scheduler. The ground-truth demand-peak series is constructed from a 20-aircraft simulation rather than from observed fleet records, so the absolute F1 numbers should be read as relative comparisons against the RUL-only baseline rather than as transferable industry benchmarks. Because the reported F1 values are obtained from the same 240-week simulated replay used for rule calibration, they should be interpreted as descriptive within-simulation comparisons rather than as evidence of out-of-sample forecasting performance.

The clustering of false-positive demand-peak alerts around weeks 90--110 of the FD003 simulation, which appears to coincide with a stretch of the simulation in which several test units sit near the cycle-bin boundary between two high-confidence rules rather than with a degradation event itself, suggests that augmenting the rule confidence with a bin-boundary-aware penalty would reduce false positives without sacrificing recall. Two open questions stand out for follow-up: whether the cluster structure is stable across a sliding training window, which would test robustness to fleet-composition drift, and whether the rule-confidence aggregation could be replaced by a calibrated probabilistic combiner without losing the auditability that motivates the present rule-based design. Future work has three natural directions: extending the pipeline to N-CMAPSS DS02 and DS04 to assess robustness under more realistic flight-condition profiles; replacing the Kaplan--Meier step with a Cox proportional-hazards layer to incorporate sensor covariates without abandoning the auditability of the rule-mining step; and validating the discovered rules on out-of-sample fleet records held by an industrial partner, which would establish whether the moderate F1 improvement observed in simulation translates to a measurable reduction in maintenance-demand forecasting error in practice.

References

1. A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. 1st Int. Conf. Prognostics Health Manage. (PHM 2008)*, 2008, pp. 1–9.
2. M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics," *Data*, vol. 6, no. 1, p. 5, 2021.
3. M. Mitici, I. de Pater, A. Barros, and Z. Zeng, "Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines," *Reliab. Eng. Syst. Saf.*, vol. 234, p. 109199, 2023.
4. Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, 2018.
5. F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proc. 1st Int. Conf. Prognostics Health Manage. (PHM 2008)*, 2008, pp. 1–6.
6. G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *Database Systems for Advanced Applications (DASFAA 2016)*, ser. *Lect. Notes Comput. Sci.*, vol. 9642. Springer, 2016, pp. 214–228.
7. S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. 2017 IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, 2017, pp. 88–95.
8. 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, 2018.

9. Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliab. Eng. Syst. Saf.*, vol. 164, pp. 74–83, 2017.
10. T. Lodygowski and S. Szrama, "Unsupervised classification and remaining useful life prediction for turbofan engines using autoencoders and Gaussian mixture models: A comprehensive framework for predictive maintenance," *Appl. Sci.*, vol. 15, no. 14, p. 7884, 2025.
11. S. Antomarioni, O. Pisacane, D. Potena, M. Bevilacqua, F. E. Ciarapica, and C. Diamantini, "A predictive association rule-based maintenance policy to minimize the probability of breakages: Application to an oil refinery," *Int. J. Adv. Manuf. Technol.*, vol. 105, pp. 3661–3675, 2019.
12. M. Wang, Y. Li, Y. Zhang, and L. Cui, "Intelligent online monitoring of rolling bearing: Diagnosis and prognosis," *Sensors*, vol. 21, no. 13, p. 4390, 2021.
13. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226–231.
14. E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, Art. no. 19, 2017.
15. E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *J. Am. Stat. Assoc.*, vol. 53, no. 282, pp. 457–481, 1958.
16. J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc. 2000 ACM SIGMOD Int. Conf. Manag. Data*, 2000, pp. 1–12.
17. I. de Pater, A. Reijns, and M. Mitici, "Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics," *Reliab. Eng. Syst. Saf.*, vol. 221, p. 108341, 2022.
18. A. Ragab, M.-S. Ouali, S. Yacout, and H. Osman, "Remaining useful life prediction using prognostic methodology based on logical analysis of data and Kaplan–Meier estimation," *J. Intell. Manuf.*, vol. 27, no. 5, pp. 943–958, 2016.
19. H. Chen, M. Yang, and X. Tang, "Association rule mining of aircraft event causes based on the Apriori algorithm," *Sci. Rep.*, vol. 14, p. 13440, 2024.
20. H. Zhu, "Real-time prognostics of engineered systems under time-varying external conditions based on the Cox PHM and VARX hybrid approach," *Sensors*, vol. 21, no. 5, p. 1712, 2021.