ABNUS JOURNAL
Your Research Partner

Jurnal Teknika dan Informatika (JTI)
E-ISSN: 2797-0000 | Vol. 6, No. 2 (2026) | pp. 131-153

A Summability Assisted Machine Learning Framework for Predictive Maintenance of Industrial Equipment

Avishek Shah[1], Prem Kumar Nayak[2], Suresh Kumar Sahani[3*]

Faculty of Science, Technology, and Engineering, Rajarshi Janak University, Janakpurdham, Nepal[1,2,3]

***Corresponding author:** avishekshah@rju.edu.np | premnayak6511@gmail.com | sureshsahani@rju.edu.np

Received: 13 September 2026 | Revised: 10 October 2026 | Revised: 25 October 2026 | Published: 05 November 2026

Abstract

Purpose: Industrial Predictive Maintenance (PdM) relies on sensor streams that are noisy, irregular, and affected by changing operating regimes, which can cause machine learning models to overreact to transient spikes or miss slow drift. This study develops a summability assisted machine learning framework that converts noisy time series into stable, interpretable features before prediction.

Research Methodology: The study uses a methodological design that combines a literature synthesis, mathematical definitions of finite Cesàro means, weighted means, cumulative and robust summaries, a leakage resistant evaluation protocol, a cost sensitive decision rule, and simulation based illustrations on a synthetic pump record.

Results: On the synthetic record, an adjacent window difference and a clipped cumulative sum detected a slow drift that a single sample limit missed while ignoring an isolated spike. Window length governed a clear trade off between false alerts and detection delay, and a worked example showed that a calibrated failure probability of 0.18 justifies inspection when the break even probability is 0.048.

Conclusions: Summability features are a low cost and interpretable complement to raw signals, but their value must be demonstrated for each asset class and decision horizon.

Limitations: The framework is not empirically validated, and all numerical results come from synthetic data.

Contributions: TThe study provides an implementable integration pattern, an evaluation protocol linking smoothing to maintenance decisions, and a real-data anchor based on the Intelligent Maintenance Systems (IMS) bearing tests for future empirical work.

Keywords: *Cesàro Mean, Condition Monitoring, Machine Learning, Predictive Maintenance, Summability Methods*

How to Cite: Shah, A., Nayak, P. K., & Sahani, S. K. (2026). A Summability Assisted Machine Learning Framework for Predictive Maintenance of Industrial Equipment. *Jurnal Teknika dan Informatika (JTI)*, 6(2), 131-153.

1. Introduction

Maintenance decisions are made under uncertainty. A plant can inspect equipment at fixed intervals, repair it after failure, or use condition evidence to intervene when degradation becomes sufficiently likely. Predictive Maintenance (PdM) aims to move the decision closer to the onset of a fault without creating excessive inspections, premature replacement, or false alarms. Its value depends on the model, data quality, the cost of acting, and the organization's ability to execute a work order (Jardine, Lin, & Banjevic, 2006; Zonta, da Costa, de Lima, da Trindade, & Li, 2020).

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

Industrial telemetry is rarely a clean stationary sequence. Vibration, temperature, current, pressure, acoustic emissions, rotational speed, and process load may be sampled at different rates. Sensors can drift or lose packets, and normal operation changes with product mix, ambient conditions, and operator practice. A model trained on one operating regime can therefore interpret a legitimate regime change as degradation or miss a fault whose signature is masked by noise (Lu, Liu, Dong, Gu, Gama, & Zhang, 2019).

This study proposes adding summability methods to the feature construction stage of a machine learning PdM pipeline. In mathematical analysis, summability methods assign meaningful limits or stable summaries to sequences whose ordinary partial behavior may be irregular (Sahani & Mishra, 2020). In a finite industrial window, the same family of ideas can be used pragmatically to aggregate observations with defined weights, track cumulative means, and compare summaries over adjacent windows. This use does not claim that every industrial sequence is divergent or that a classical convergence theorem predicts failure. Rather, summability provides a principled vocabulary for stabilizing local evidence (Lei, Li, Guo, Li, Yan, & Lin, 2018; Sahani, Paudel, Ghimire, & Thakur, 2022).

The study addresses four research questions. The first is how summability derived features can complement raw and conventional statistical features. The second is what safeguards are required to prevent smoothing from hiding short duration faults. The third is how models should be evaluated under rare failures and temporal dependence. The fourth is how a prediction can be converted into a maintenance action whose cost and safety implications are explicit. The study contributes a layered architecture, finite window summary definitions, a leakage resistant evaluation protocol, a cost sensitive decision rule, simulation based illustrations, and a protocol anchored to real bearing data, while noting that impulsive faults still require raw features and that model outputs should inform rather than replace engineering judgment. The overall pipeline is shown in Figure 1.

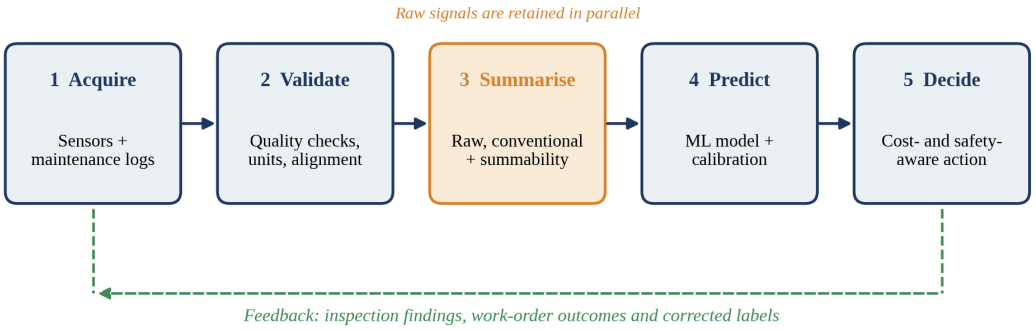


Figure 1. Proposed summability assisted predictive maintenance pipeline

Figure 1 presents the five stages of the proposed pipeline, namely acquiring sensor and maintenance data, validating their quality, summarizing them into raw, conventional, and summability features, predicting risk with a calibrated machine learning model, and deciding on a cost and safety aware action. Summability is deliberately placed between data quality control and model prediction, because aggregation should operate only on validated data and should never erase the raw signals, which are retained in parallel as the note above the third stage indicates. The dashed feedback path shows that inspection findings, work order outcomes, and corrected labels flow back to the start of the pipeline, so the system learns from the consequences of its recommendations.

2. Literature Review and Hypothesis/es Development

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

2.1 Predictive Maintenance and Condition Based Maintenance

Jardine, Lin, and Banjevic (2006) describe condition based maintenance as a sequence of data acquisition, data processing, and maintenance decision making. Systematic reviews of PdM in Industry 4.0 likewise emphasize the integration of monitoring, interpretation, and work planning rather than isolated algorithmic accuracy (Carvalho, Soares, Vita, Francisco, Basto, & Alcalá, 2019; Zonta, da Costa, de Lima, da Trindade, & Li, 2020). Data driven surveys find machine learning valuable for fault classification and Remaining Useful Life (RUL) estimation but warn that results depend strongly on evaluation design (Lei, Yang, Jiang, Jia, Li, & Nandi, 2020).

The PdM task must be specified before a model is chosen. Binary classification asks whether a failure will occur within a horizon, multiclass classification distinguishes failure modes, anomaly detection identifies deviations without complete fault labels, RUL regression estimates the time or cycles remaining, and survival analysis models time to event while handling censored observations (Serradilla, Zugasti, Rodriguez, & Zurutuza, 2022). These outputs are related but not interchangeable. A model with high classification accuracy can still be unhelpful if it detects a fault only minutes before a long procurement lead time. The common tasks and their operational outputs are summarized in Table 1.

Table 1. Common PdM tasks and their operational outputs

Task	Typical Output	Decision Supported	Main Risk
Failure classification	Probability of failure within a horizon	Inspect, repair, or prepare parts	Rare failures and false alarms
Anomaly detection	Deviation score	Trigger investigation	Novel but harmless operating modes
RUL estimation	Estimated remaining cycles or time	Plan the intervention window	Uncertain end of life trajectory
Survival modeling	Failure time distribution	Compare risk over time	Censoring and changing hazards
Fault diagnosis	Fault mode probabilities	Choose an inspection or repair procedure	Overlapping fault signatures

Table 1 shows that each PdM task produces a different kind of output and supports a different maintenance decision, from preparing parts for a predicted failure to choosing a specific repair procedure. The last column shows that each task carries its own risk, such as false alarms under rare failures or censored data (Pang, Shen, Cao, & van den Hengel, 2022). The table therefore underlines that the choice of task should follow from the decision the plant needs to make and from the data available, rather than from the popularity of a particular algorithm.

2.2 Machine Learning and Temporal Modeling

Tree ensembles suit industrial settings because they handle nonlinear interactions without a large labeled corpus. Random forests reduce variance through aggregation (Breiman, 2001), gradient boosting builds an additive predictive model (Friedman, 2001), and support vector machines work well with engineered features in moderate dimensional spaces (Cortes & Vapnik, 1995). Recurrent, convolutional, and transformer architectures can represent temporal dependencies and estimate RUL directly from sensor sequences (Li, Ding, & Sun, 2018; Zhao, Yan, Chen, Mao, Wang, & Gao, 2019; Lei, Li, Guo, Li, Yan, & Lin, 2018), but they require careful temporal validation and can be less transparent to maintenance teams (Serradilla, Zugasti, Rodriguez, & Zurutuza, 2022). No model family is universally best.

Recent reviews reinforce the need to compare models with consistent splits and metrics. A systematic review of time series algorithms in PdM analyzes preprocessing, benchmark datasets, and forecasting tasks (Syed, Hasan, Chowdhury, Rahman, & Ahmed, 2025), and a review of supervised methods published from 2019 through 2024 notes the limited availability of real industrial datasets and the

importance of explainability and generalization (Guidotti, Pandolfo, & Pulina, 2025). Application focused reviews emphasize that an algorithm's suitability depends on sector specific operating conditions and failure consequences (Tsallis, Papageorgas, Piromalis, & Munteanu, 2025), and comparative studies of machine learning and deep learning for machine failure prediction show that results vary with data preparation and class balance (Yadav, Kaushik, & Yadav, 2024). Reviews of the shift toward Industry 5.0 also stress human centric maintenance (Murtaza, Saher, Zafar, Moosavi, Aftab, & Sanfilippo, 2024).

2.3 Summability Methods and Their Relevance

Summability theory has a long tradition in the analysis of series and signals. Sahani and Mishra (2020) investigated Nörlund summability and its role in the convergence of infinite series, Sahani and Mishra (2021) explored Borel's method for Fourier series, and Sahani and Mishra (2020) examined Cesàro sequence spaces in functional analysis. Further studies addressed the approximation of functions and signals through Fourier Laguerre, Legendre, and derived Fourier series (Sahani & Mishra, 2021) and investigated decreasing sequences and Nörlund summability in ultraspherical and double Fourier series (Sahani, Mishra, & Rathour, 2022). Related works explored matrix summability, Jacobi series, and Borel type summation (Sahani et al., 2022; Sahani, Kumar, & Dubey, 2023), convergence rates and almost decreasing sequences (Pokharel, Pahari, & Sahani, 2023; Sahani, & Mishra, 2023; Sahani, & Das, 2023), generalized almost matrix summability (Kushwaha, Thakur, Sahani, & Suryawanshi, 2023), matrix convergence and absolute summability (Kushwaha et al., 2023), and series transformations and analytic continuation (Sahani et al., 2022; Sahani, & Thakur, 2020; Sahani, Prasad, & Thakur, 2022). Collectively, these studies show the importance of summability in convergence analysis, function approximation, and applied mathematics.

Conventional moving averages, exponentially weighted means, and cumulative statistics already appear in signal processing and process monitoring, so the contribution here is not to claim that smoothing is new (Sahani, Paudel, & Thakur, 2022; Sahani, Thakur, & Sahu, 2022). It is to formulate a transparent and reproducible feature layer using summability inspired operators and then to evaluate whether this layer improves stability and decision utility relative to raw only and conventional feature baselines. Because a smoother can suppress a real transient as well as noise, a summability assisted system should keep parallel raw, frequency domain, robust, and weighted representations and let validation establish which features are useful (Sahani, Mishra, & Pahari, 2020; Sahani, Mishra, & Pahari, 2021; Sahani, Thakur, & Dubey, 2022; Sahani, Thakur, & Kushwaha, 2021).

2.4 Research Gaps

The literature suggests a persistent gap between prediction performance and the reliability of the entire maintenance decision chain. Many studies report a single train and test score, although adjacent time windows from the same asset can leak nearly identical information across splits (Kapoor & Narayanan, 2023). Public benchmarks are useful for reproducibility, but simulated or curated data may not represent the complexity of a deployed plant. The AI4I 2020 dataset is explicitly synthetic even though it reflects industrial PdM variables (Matzka, 2020), and the National Aeronautics and Space Administration (NASA) Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset contains simulated turbofan engine degradation trajectories that support RUL estimation methods development but not direct evidence of field performance on pumps or motors (Saxena et al., 2008). A second gap is that smoothing is treated as generic preprocessing although its effect depends on window width, weights, and the fault time scale, and a third is that classification probabilities are often detached from inspection capacity, lead times, safety, and the relative costs of errors. These gaps and the corresponding design responses are summarized in Table 2.

Table 2. Research gaps and corresponding design responses

Gap in Common Practice	Potential Consequence	Response in this Framework
Single accuracy score	Rare failures may be missed	Report precision, recall, PR AUC, calibration, and cost

Gap in Common Practice	Potential Consequence	Response in this Framework
Random row split on time series	Temporal leakage	Use chronological or asset wise splits
Smoothing applied without ablation	Transient fault may be hidden	Compare raw, conventional, and summability features
Probability treated as action	Threshold ignores operational context	Use expected cost and safety constrained decisions
Synthetic benchmark treated as field evidence	Overstated generalizability	Label data provenance and validate locally
Model not monitored after release	Drift and sensor changes go unnoticed	Monitor inputs, calibration, alerts, and outcomes

Table 2 links six common weaknesses in PdM practice to their consequences and to the specific responses built into the proposed framework. Reliance on a single accuracy score and random row splits can make a model look far better than it is, so the framework requires precision, recall, the Precision Recall Area Under the Curve (PR AUC), calibration, and cost to be reported under chronological or asset wise splits (Kapoor & Narayanan, 2023). The remaining gaps are addressed through ablation against baselines, expected cost decisions, explicit data provenance, and continuous monitoring against drift (Lu et al., 2019). The table thus serves as a design checklist that connects each element of the framework to a documented gap.

3. Research Methodology

3.1 Study Design

This study is a methodological framework paper without a new controlled experiment, combining established literature, recent reviews, and dataset documentation, with clearly synthetic illustrations. Its four objectives are to define finite window summability features, to specify a leakage resistant validation protocol, to demonstrate an auditable decision pathway for a critical pump, and to identify limitations and questions for future evaluation. The data preparation chain and leakage controls are shown in Figure 2.

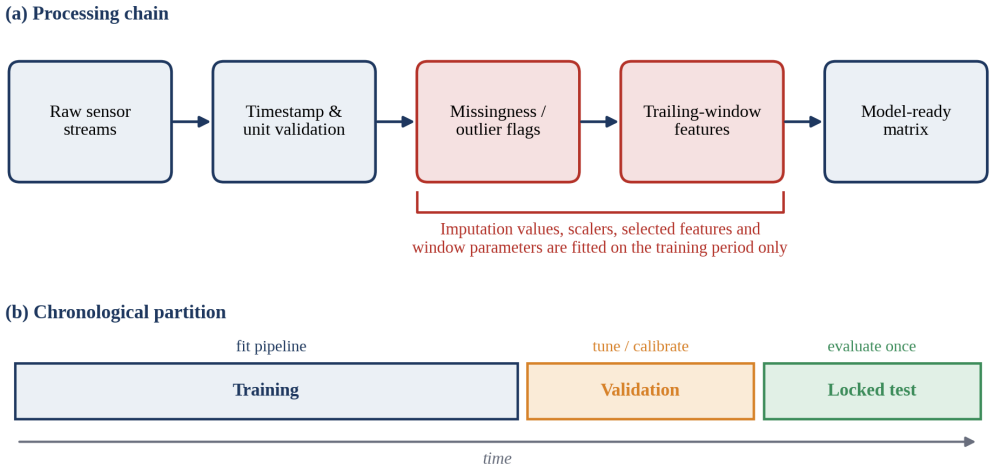


Figure 2. Data preparation and leakage controls

Figure 2 shows in panel (*a*) the processing chain that turns raw sensor streams into a model ready matrix through timestamp and unit validation, missingness and outlier flags, and trailing window features. The bracket under the two highlighted steps states the key rule that imputation values, scalers, selected features, and window parameters are fitted on the training period only. Panel (*b*) shows the chronological partition into a training period used to fit the pipeline, a validation period used to tune and calibrate the model, and a locked test period evaluated once. The figure therefore

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

emphasizes that quality checks and split discipline precede modeling, and that every parameter learned from data must be fixed before it is applied to later periods, which is the main defense against the leakage problems documented by Kapoor and Narayanan (2023).

3.2 Mathematical Foundations

Let x_t denote a sensor measurement at time t, such as vibration Root Mean Square (RMS), temperature, current, or pressure. For a trailing window of length m, the window is $X_t^{(m)} = \{x_{t-m+1}, \ldots, x_t\}$. All online features must use observations available at or before time t, and future values are prohibited. For multivariate telemetry, the operators are applied channel by channel or to a physically meaningful derived signal. The finite window arithmetic mean, or finite Cesàro mean, is the simplest summability inspired operator and is given in Formula 1.

$$C_t(m) = \frac{1}{m} \sum_{i=0}^{m-1} x_{t-i} \tag{1}$$

This mean reduces sensitivity to zero meFormula 2an noise when the signal is locally stable. The running Cesàro mean over the full history, $C_n = (1/n)\ \Sigma_k\ x_k$, becomes insensitive to recent deterioration, so a finite window is more responsive, and its length should follow the degradation time scale and intervention lead time. A weighted mean emphasizes some observations more than others, as shown in Formula 2.

$$W_t = \frac{\sum_{i=0}^{m-1} w_i\, x_{t-i}}{\sum_{i=0}^{m-1} w_i}, \quad w_i \geq 0 \tag{2}$$

For a recency weighted feature, the largFormula 3est weight is assigned to the newest observation. An exponential choice $w_i = \lambda^i$ with $0 < \lambda < 1$ gives an exponentially weighted moving average after normalization. Because different conventions reverse the index, an implementation must state whether $i = 0$ denotes the newest or the oldest sample. A cumulative sum can expose persistent departures from a baseline. Given a reference mean μ_0 and a tolerance parameter k, a one sided cumulative deviation statistic is defined in Formula 3.

$$S_t = max\,(0, S_{t-1} + x_t - \mu_0 - k) \tag{3}$$

This is a CUSUM style monitoring statistic rather than a universal failure predictor. Where spikes or heavy tailed noise make the standard deviation unstable, a robust dispersion summary such as the Median Absolute Deviation (MAD) is useful, as given in Formula 4.

$$MAD_t = \text{median}_i \left| x_{t-i} - \text{median}\left(X_t^{(m)}\right) \right| \tag{4}$$

These features describe different behaviors, namely mean level, recent level, persistent deviation, and within window variability, and their complementarity is the reason to retain several of them rather than compress every signal into one smoothed series. In addition, the adjacent window difference $\Delta C_t = C_t(m_s) - C_t(m_l)$ compares a short and a long window, and a sustained positive value may indicate rising condition severity. The transformation from a trailing window to model inputs is shown in Figure 3.

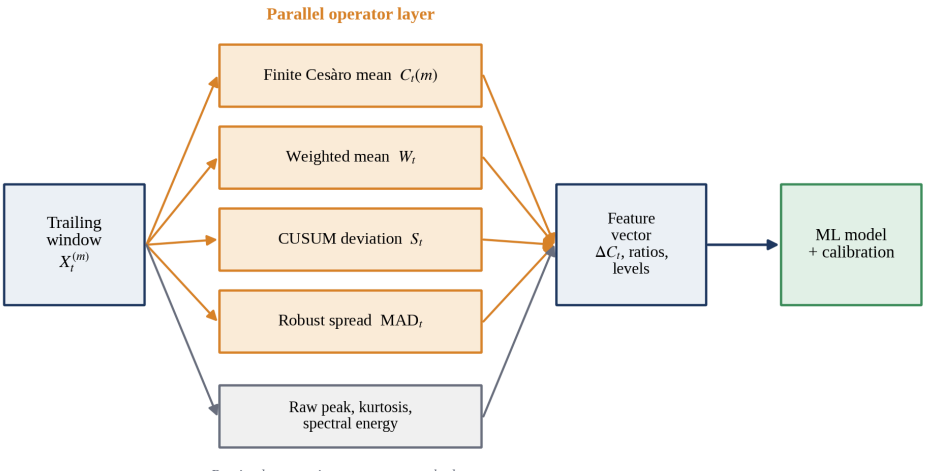


Figure 3. How summability operators become interpretable features

Figure 3 shows how a single trailing window feeds a parallel operator layer that computes the finite Cesàro mean, the weighted mean, the CUSUM deviation, and the robust spread, while a separate branch keeps raw peak, kurtosis, and spectral energy. All outputs are combined into one feature vector that also contains the adjacent window difference, ratios, and levels, and this vector is passed to a calibrated machine learning model. The parallel arrangement is the central design choice of the framework, because it allows the model to use stable summaries for slow drift while still seeing the raw evidence needed to detect impulsive faults.

3.3 Framework, Algorithm, and Deployment

The framework has six stages, namely acquiring and aligning sensor channels and maintenance events, validating units, timestamps, sensor health, and missingness, constructing trailing windows, computing parallel raw, conventional, and summability features, training and calibrating a predictive model, and mapping risk to an inspection or maintenance decision. Raw window features include the latest value, range, peak to peak amplitude, and slope, conventional features include the mean, standard deviation, quantiles, Root Mean Square (RMS), kurtosis, and band energy, and summability features include the operators defined above and ratios of short and long window summaries. For classification, the target $y_t(H)$ equals 1 if a documented failure occurs within the next H hours or cycles and 0 otherwise, and H must reflect the maintenance response time. Fault labels should be reconciled with work orders and engineering evidence, because a log timestamp is not always the true onset of a fault.

During online inference, the system flags invalid data, forms trailing windows from data available at time t, computes all feature families, applies the training fitted pipeline without refitting, estimates the failure probability with uncertainty flags, evaluates expected cost under safety and capacity constraints, issues a ranked and explained recommendation, and logs the action and outcome. The deployment architecture is shown in Figure 4.

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

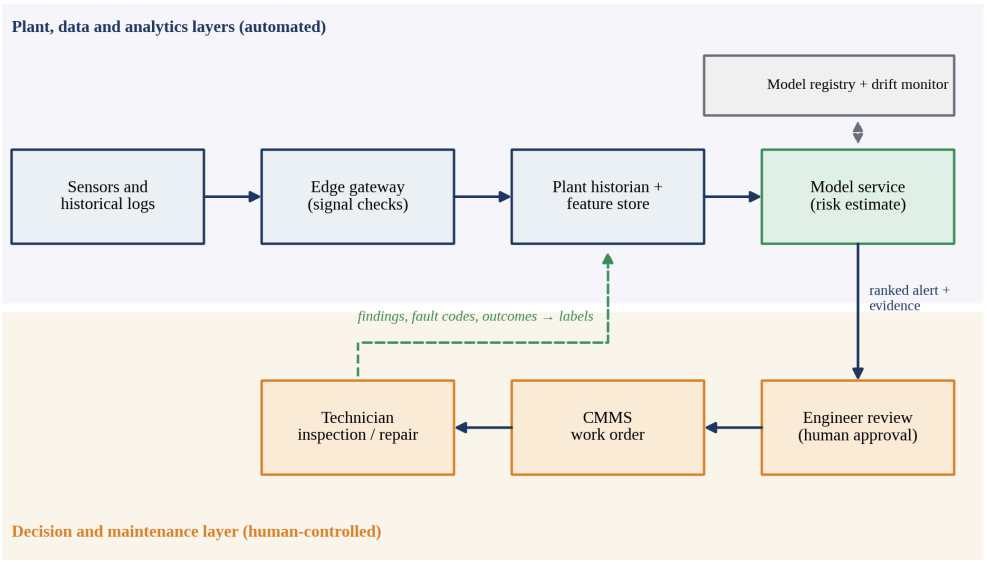


Figure 4. Deployment architecture with human approval and monitoring

Figure 4 separates the deployment into an automated plant, data, and analytics layer and a human controlled decision and maintenance layer. In the upper layer, sensors and historical logs pass through an edge gateway for signal checks to a plant historian and feature store, and a model service produces risk estimates under the supervision of a model registry and drift monitor. In the lower layer, ranked alerts are reviewed by an engineer, converted into a work order in the Computerized Maintenance Management System (CMMS), and executed by a technician. The dashed return path shows that findings, fault codes, and outcomes re enter the feature store as labels. This separation allows a model to be updated without silently changing maintenance procedures (Njor et al., 2024).

3.4 Model Choices and Training Protocol

Logistic regression is a transparent first baseline, random forests and gradient boosted trees capture nonlinear interactions (Breiman, 2001; Friedman, 2001), and sequence models are justified only when temporal order carries information that summaries cannot retain (Zhao, Yan, Chen, Mao, Wang, & Gao, 2019). The proposed design compares at least four configurations, namely raw and statistical features with a simple baseline, conventional features with a tree ensemble, summability features with the same ensemble, and a sequence model where data volume supports it. Holding the model family constant isolates the contribution of summability.

For each asset, records are ordered by time, with training on an earlier period, validation on a later period, and testing on the latest untouched period. When assets share manufacturing batches or environmental conditions, grouping by asset or production lot prevents near duplicate windows from crossing partitions. Scaling, imputation, feature selection, resampling, and threshold tuning are fitted only on the training and validation data, and the final test set is used once. Calibration is assessed with reliability diagrams, the Brier score, and the calibration slope on data not used to fit the model, and confidence intervals are estimated by resampling assets or time blocks rather than individual time points (Kapoor & Narayanan, 2023). Three split designs are compared in Figure 5.

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

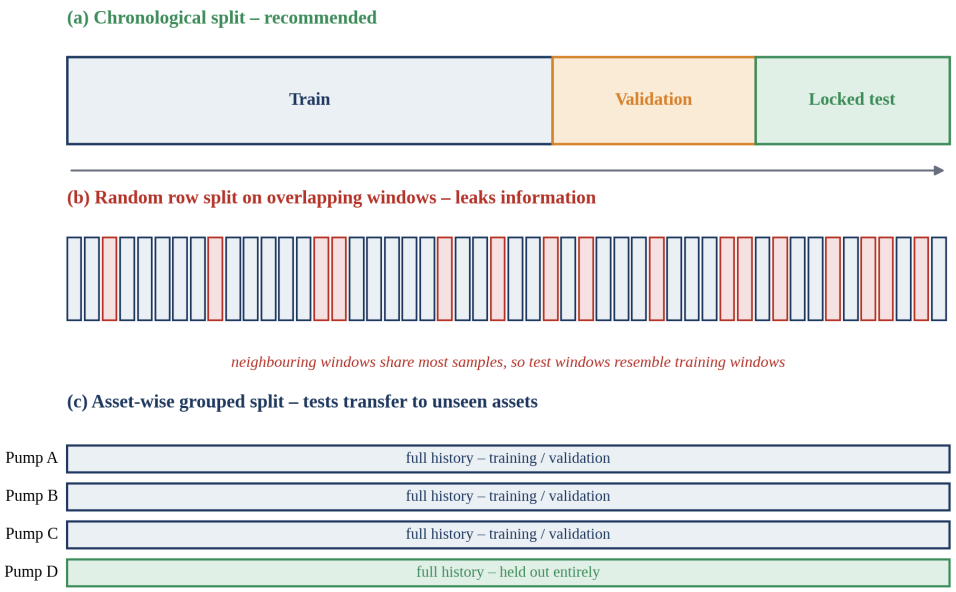


Figure 5. Leakage resistant temporal evaluation with (a) chronological split, (b) random row split on overlapping windows, and (c) asset wise grouped split

Figure 5 contrasts three ways of partitioning the same records. The chronological split in panel (*a*) mirrors deployment, where the model only ever sees the past, and it is the recommended default. The random row split in panel (*b*) scatters training and test windows throughout the record, so neighboring overlapping windows that share most samples fall on both sides of the partition and inflate scores. The asset wise split in panel (*c*) withholds whole machines, here pump D, and therefore tests transfer to equipment the model has never seen. Where assets share manufacturing batches or operating conditions, the asset wise design is the stricter and often more honest test, and the figure shows visually why random splits should not be used for overlapping time windows.

3.5 Evaluation Metrics and Decision Rule

Class imbalance is typical when failures are rare, and accuracy can remain high even if a model misses nearly all failures (Haixiang et al., 2017). Precision measures the fraction of alerts that correspond to a target event, and recall measures the fraction of target events detected. The precision recall curve is often more informative than receiver operating characteristic analysis when positive cases are uncommon, but PR AUC depends on prevalence, so the event rate must accompany reported values. Lead time and false alarms per asset month should be reported alongside predictive metrics, and RUL errors should be examined near the intervention boundary, because underestimating remaining life causes early maintenance while overestimating it causes missed failures (Li et al., 2018). The evaluation measures are summarized in Table 3.

Table 3. Evaluation measures and interpretation

Measure	Definition or purpose	Caution
Precision	TP / (TP + FP)	Depends on event prevalence and alert policy
Recall	TP / (TP + FN)	High recall can create excessive alerts
F1 score	Harmonic mean of precision and recall	Hides probability calibration and costs
PR AUC	Area under the precision recall curve	Compare at similar prevalence and split design
Brier score	Mean squared probability error	Combines discrimination and calibration effects

Measure	Definition or purpose	Caution
Lead time	Time from alert to event or action	Must be compared with response and parts lead time
False alerts per asset time	Operational alert burden	Requires an explicit asset time denominator
Expected cost	Probability weighted maintenance and failure cost	Cost inputs must be documented and sensitivity tested

Table 3 lists eight measures that together describe both predictive quality and operational usefulness. Precision and recall, defined from True Positives (TP), False Positives (FP), and False Negatives (FN), capture the balance between missed failures and unnecessary alerts, while the F1 score and PR AUC summarize this balance but hide calibration and cost. The Brier score adds calibration information, and lead time, alert burden, and expected cost translate output into maintenance consequences, while the caution column shows that each measure can mislead when reported alone. The table therefore supports the recommendation that these measures be reported together rather than replaced by a single accuracy figure (Guidotti et al., 2025).

Prediction is not the decision. Let p be the calibrated probability of failure within horizon H, C_{F} the expected consequence cost of an unmitigated failure, C_{I} the cost of inspection, and C_{P} the cost of planned intervention. A simplified two action model compares the expected loss of waiting with the expected loss of acting, as shown in Formula 5.

$$L(\text{wait}) = p \cdot C_F + C_{\text{monitor}}, \qquad L(\text{act}) = C_I + C_P \tag{5}$$

The rule of acting when L(act) is lower than L(wait) is intentionally simple, omitting residual risk and imperfect inspection, and for safety critical assets a regulatory threshold may require intervention regardless of cost. A practical system should return an evidence summary rather than a bare red or green label. The decision flow is shown in Figure 6.

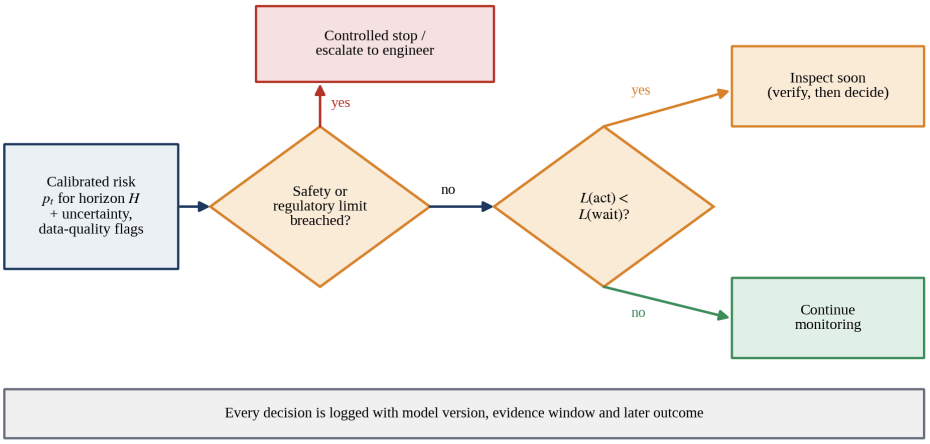


Figure 6. From prediction to maintenance decision

Figure 6 shows a calibrated risk estimate, together with its uncertainty and data quality flags, first passing a safety or regulatory check. If a limit is breached, the flow goes directly to a controlled stop or escalation to an engineer, regardless of cost. Otherwise, the expected loss of acting is compared with the expected loss of waiting, and the recommendation is either to inspect soon and verify before deciding or to continue monitoring. The bar at the bottom states that every decision is logged with the model version, evidence window, and later outcome, so that it can be audited. The figure thus places the economic comparison of Formula 5 inside a safety first structure and keeps the final action reviewable.

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

4. Results and Discussions

4.1 Behavior of the Operators on a Simulated Pump Record

To make the operators concrete, a synthetic pump record was generated with a fixed random seed. Vibration RMS is sampled every 10 minutes for ten days around a baseline of 2.0 mm/s, with Gaussian noise of standard deviation 0.12 mm/s and a small daily load cycle. Two effects are superimposed by construction, namely one isolated spike of 1.7 mm/s on day 3.2 and a slow accelerating drift that starts on day 4.5 and reaches 0.9 mm/s above baseline by day 10. Because the record is simulated, it shows how the operators behave and does not measure how well they would perform on field data. The results are shown in Figure 7.

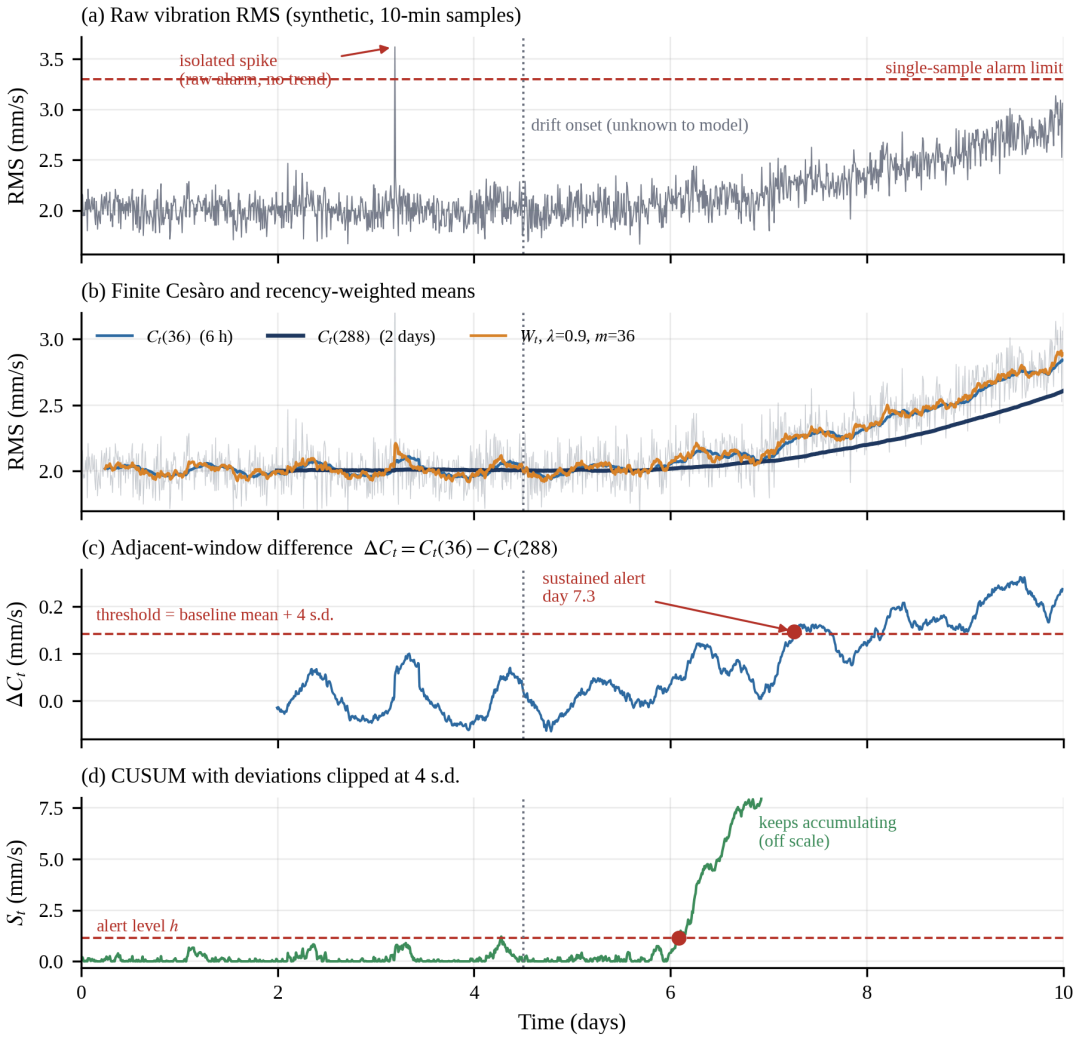


Figure 7. Behavior of the summability operators on a synthetic pump vibration record, showing (a) raw RMS, (b) finite Cesàro and recency weighted means, (c) adjacent window difference, and (d) clipped CUSUM statistic

Figure 7 shows in panel (*a*) that the spike reaches 3.62 mm/s and crosses a 3.3 mm/s single sample alarm limit, whereas the drift never does, so a raw threshold rule produces a false alarm and misses the trend. Panel (*b*) shows that the short Cesàro mean and the recency weighted mean follow the drift while the long mean lags behind it. In panel (*c*), the spike moves ΔC_t only to 0.08 mm/s, below the alert threshold of 0.14 mm/s, and the difference $\Delta C_t = C_t(36) - C_t(288)$ raises a sustained alert, defined as twelve consecutive samples above threshold, on day 7.3, when the drift is 0.23 mm/s above baseline. In panel (*d*), the clipped CUSUM peaks at 0.90 mm/s during the spike, below its alert level

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

of 1.13 mm/s, and crosses that level on day 6.1, when the drift is only 0.07 mm/s. The CUSUM alert is earlier but rests on a smaller departure, and neither rule is claimed to be superior. Clipping at four standard deviations prevents the spike from triggering the CUSUM, and a first trial with windows (6.144) raised no alert before day 10, illustrating sensitivity to window choice.

The window length trade off was quantified with 200 Monte Carlo replications of the same generator, without the spike or the daily cycle, using an alert rule of a trailing mean above 2.3 mm/s, a level the drift reaches on day 7.7. The results are presented in Figure 8.

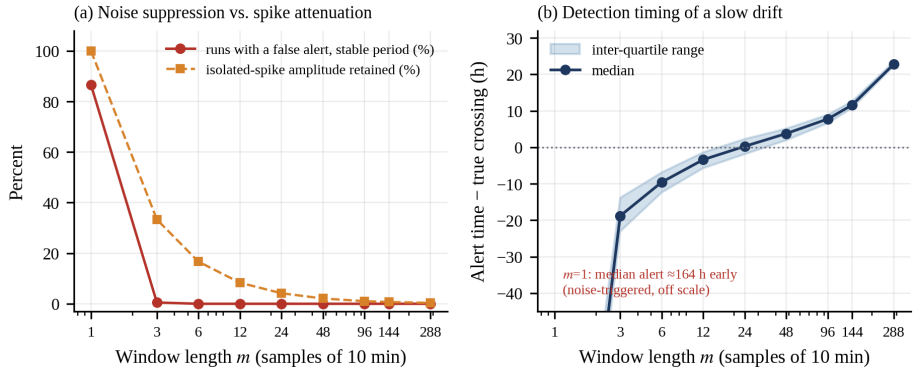


Figure 8. Window length trade off on the synthetic record (200 replications), showing (a) false alert rate and isolated spike amplitude retained and (b) alert timing relative to the true level crossing

Figure 8 shows in panel (*a*) that with *m*=1, that is raw samples, 86.5% of replications raised at least one false alert during the stable days 2.0 to 4.4, whereas with *m*=3 the rate fell to 0.5% and with m of 6 or more no false alert occurred. The price is attenuation, because an isolated spike keeps only 100/m percent of its amplitude in a window mean. Panel (*b*) shows the price in delay, with the median alert time moving from about 9.5 hours before the true crossing at *m*=6, a noise triggered early alert on a rising trend, to about 0.3 hours after it at *m*=24 and to 11.6 and 22.8 hours after it at *m*=144 and *m*=288. The figure demonstrates that no window length is best in general, and that the window should be chosen against the fault time scale and the available response time rather than by minimizing a validation score alone.

4.2 Why Accuracy Misleads Under Rare Failures

The importance of the evaluation measures in Table 3 was illustrated with synthetic scores for 200.000 windows with 1% failure prevalence (2.043 positives), with negative scores drawn from a unit variance normal distribution and positive scores shifted by two standard deviations. The results are presented in Figure 9.

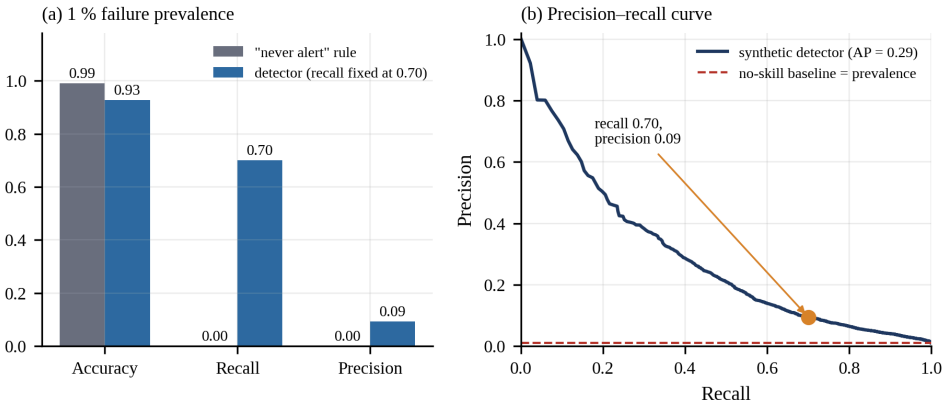


Figure 9. Why accuracy alone misleads under rare failures, showing (a) a never alert rule versus a detector at fixed recall and (b) the precision recall curve of a synthetic scoring model at 1% prevalence

Figure 9 shows in panel (*a*) that a rule that never raises an alert attains 0.99 accuracy with zero recall and zero precision. A detector tuned to 0.70 recall has lower accuracy (0.93) because it raises 15.256 alerts, of which only 1.431 are true, giving a precision of 0.09, or roughly ten false alerts for every true one. Panel (*b*) shows the full precision recall curve, with an average precision of 0.29 against a no skill baseline equal to the prevalence of 0.01. The figure demonstrates why accuracy alone is misleading for rare failures and why recall, precision, PR AUC, calibration, lead time, and alert burden must be reported together, with prevalence stated alongside any PR AUC value (Haixiang et al., 2017).

4.3 Worked Example for an Industrial Centrifugal Pump

Consider a centrifugal pump supplying a continuous process in a water treatment or process plant. Degradation may stem from bearing wear, lubrication problems, misalignment, or cavitation, and the example is industrially plausible but illustrative, with synthetic values and no named plant. A single threshold alarm might be triggered by an isolated spike that could come from a sensor disturbance, while a slow increase over several days may remain below a fixed threshold even though it represents progressive degradation. A summability feature set tracks both phenomena, because the raw peak remains available for transient detection while short and long window means quantify persistent drift. When the alert rules fire on the synthetic record is shown in Figure 10.

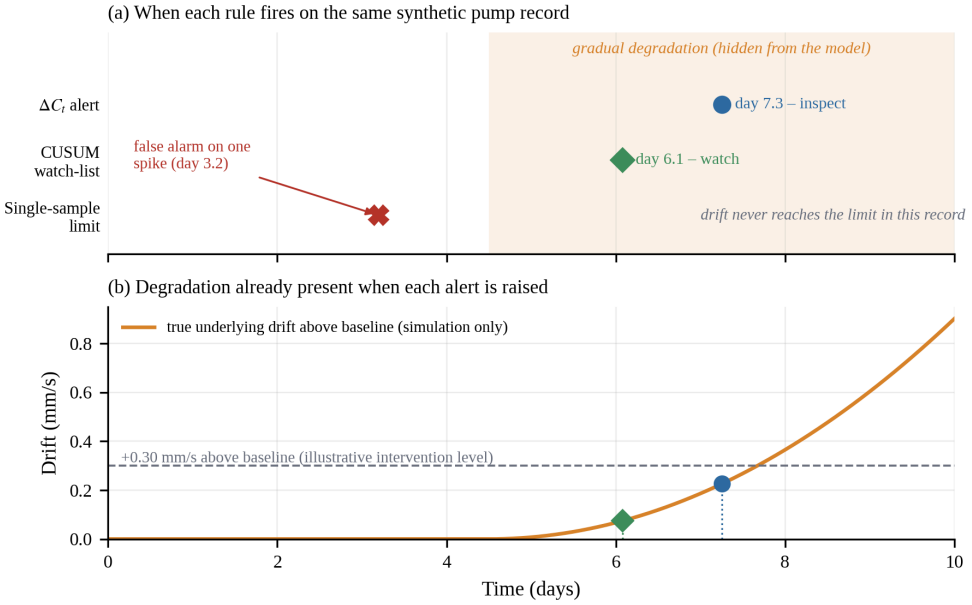


Figure 10. Synthetic pump record showing (a) when a single sample limit, a CUSUM watch list rule, and the ΔC_t rule fire and (b) the degradation already present when each alert is raised

Figure 10 reuses the synthetic record of Figure 7, so its numbers can be checked against that figure. Panel (*a*) shows that the single sample limit fires once, on the isolated spike of day 3.2, and never on the drift, whereas the CUSUM watch list rule fires on day 6.1 and the ΔC_t rule on day 7.3. Panel (*b*) shows that the underlying drift at those moments, visible only because the data are simulated, is 0.07 and 0.23 mm/s above baseline, and that the illustrative intervention level of 0.30 mm/s above baseline is reached on day 7.7. Both summability rules therefore alert before the intervention level is reached. The figure is not a performance result, because the drift shape, thresholds, and window lengths were chosen for illustration and a real asset would not supply the lower panel.

For a short illustrative window with vibration RMS values of 2.0, 2.2, 2.1, 2.5, and 2.8 mm/s, the five point mean is 2.32 mm/s. If weights of 1, 2, 3, 4, and 5 are assigned from oldest to newest, the weighted mean is (1 × 2.0 + 2 × 2.2 + 3 × 2.1 + 4 × 2.5 + 5 × 2.8) / 15 = 2.45 mm/s. The adjacent window difference $\Delta C = C(\text{short}) - C(\text{long})$ can then indicate whether the local level is rising relative to a longer baseline. This feature is not a diagnosis, since vibration can rise because of load or speed

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

changes, so the system should compare like for like operating regimes and use spectral features and technician inspection to distinguish fault modes.

4.4 Alert and Action Logic

Assume that the model estimates a calibrated probability of 0.18 that the pump fails within the next seven days. For illustration only, suppose that an unmitigated failure costs NPR 250.000 and a planned inspection costs NPR 12.000. The expected failure loss while waiting is then 0.18 × NPR 250.000 = NPR 45.000, which exceeds the inspection cost, so the framework recommends inspection, subject to safety rules and actual inspection effectiveness. The calculation is summarized in Table 4.

Table 4. Illustrative decision calculation for the pump example

Quantity	Illustrative Value	Interpretation
Prediction horizon	7 days	Chosen to allow a planned response
Calibrated failure probability	0.18	Hypothetical model output
Expected loss if failure occurs	NPR 250,000	Hypothetical consequence estimate
Expected failure loss while waiting	NPR 45,000	0.18 × 250.000
Inspection and preparation cost	NPR 12,000	Hypothetical direct cost
Simplified comparison	45.000 > 12.000	Inspection is favored in this simplified case
Required qualification	Not a field result	Residual risk and inspection efficacy omitted

Table 4 shows the full chain from a hypothetical model output to a recommendation. With a seven day horizon chosen to allow a planned response, the calibrated probability of 0.18 and a consequence of NPR 250.000 imply an expected failure loss of NPR 45.000 if the plant waits, which is almost four times the NPR 12.000 cost of inspection. The comparison therefore favors inspection. The last row is as important as the arithmetic, because it states that the result is not a field estimate and that residual risk after maintenance and imperfect inspection effectiveness are omitted. In a real plant, the failure cost should be decomposed into its components and tested for sensitivity. Such a sensitivity analysis is shown in Figure 11.

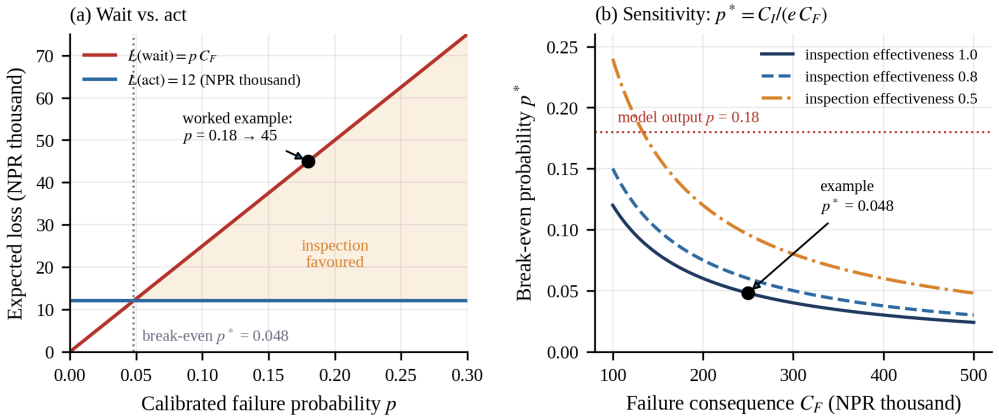


Figure 11. Expected loss comparison for the pump example, showing (a) waiting versus acting as a function of failure probability and (b) break even probability under different inspection effectiveness

Figure 11 turns the arithmetic of Table 4 into a decision boundary. Panel (*a*) shows that, with the illustrative costs, waiting costs more than acting once p exceeds the break even value p = 12.000 / 250.000 = 0.048, so the model output of 0.18 lies well inside the region where inspection is favored. Panel (*b*) adds the effect of imperfect inspection, because if an inspection removes only a fraction e of the avoidable loss, the break even probability becomes $p = C_I / (e \times C_F)$, which equals 0.060 at e = 0.8

© 2026 Abnus | Open Access — CC BY 4.0 | abnus.co | DOI:10.52909/jti.v6i2.461

and 0.096 at $e = 0.5$ when C_F is NPR 250.000. For $p = 0.18$, the recommendation would change only if C_F fell below about NPR 67.000 at $e = 1.0$ or NPR 133.000 at $e = 0.5$. The figure therefore shows that the inspection decision is robust across a wide range of assumptions, and the recommended output remains to inspect, verify bearing condition, and then decide whether to continue monitoring or schedule a repair.

4.5 Application Pathways and Real Data Anchor

Rotating machinery is a strong initial use case because its condition variables are routinely measured and gradual deterioration creates trends over multiple windows, although impulsive defects still require raw waveform or envelope analysis. In manufacturing, the synthetic AI4I 2020 dataset with 10.000 instances supports teaching and initial benchmarking (Matzka, 2020). In aerospace, CMAPSS provides simulated turbofan trajectories for comparing RUL models (Saxena, Goebel, Simon, & Eklund, 2008). Sensor and event log fusion (Liu & Hui, 2024) and continual learning under nonstationary conditions (Benatia, Hafsi, & Ben, 2025; Parisi, Kemker, Part, Kanan, & Wermter, 2019) further extend these pathways. The application families and their evidence status are summarized in Table 5.

Table 5. Application map with evidence status

Application Family	Signals and Fault Time Scale	Role of Summability Features	Evidence Status
Rolling element bearings (test rig)	Accelerometer snapshots, defects develop over days of continuous running	Trailing means and ΔC_t of snapshot RMS and kurtosis, CUSUM on RMS	Real run to failure data with naturally developed defects (IMS tests), no results reported here
Process plant centrifugal pumps	Vibration, bearing temperature, current, pressure, and flow, drift over days to weeks	Drift screening with short and long means, raw peak kept for transients	Synthetic worked example, field validation required
Motors, gearboxes, and fans	Current, vibration, and temperature, changes over weeks	Persistent level changes, spectral features for impulsive faults	Not evaluated in this study
Machine tools and discrete manufacturing	Torque, speed, temperature, and tool wear, per production cycle	Windows counted in cycles, CUSUM on wear indicators	AI4I 2020, synthetic, 10.000 instances
Aircraft turbofan engines	Multisensor degradation trajectories under varying conditions	Windowed trend features for RUL models conditioned on regime	CMAPSS, simulated run to failure data
Utility pumping stations and small plants	Pump and motor condition data with maintenance logs	Pilot on one critical asset in shadow mode	No data yet, candidate pilot only

Table 5 records, for each application family, the signals normally available, the fault time scale that sets the window length, the role of summability features, and the status of public evidence. Only the bearing test rig is backed by real run to failure data, while the pump example is synthetic, AI4I 2020 is synthetic, CMAPSS is simulated, and motors, gearboxes, fans, and small utility plants have not yet been evaluated. The distinction between real, simulated, and synthetic data is deliberate, because only real data can say anything about field behavior. The table therefore makes explicit how much of the proposed framework rests on demonstrated evidence and how much remains a candidate for future validation.

The IMS bearing tests, distributed through NASA's Prognostics Data Repository, give the framework a real physical anchor with the same 10 minute cadence used in the pump example (Lee, Qiu, Yu, Lin, & Rexnord, 2007). Four Rexnord ZA 2115 double row bearings were mounted on one shaft driven at

a constant 2000 rpm under a 6000 lb radial load with forced lubrication, and accelerometers on the bearing housings recorded one second snapshots at 20 kHz (20.480 points) every ten minutes until failure (Qiu, Lee, Lin, & Yu, 2006). The three tests ended with an inner race defect in bearing 3 and a roller element defect in bearing 4 in test 1, an outer race defect in bearing 1 in test 2, and an outer race defect in bearing 3 in test 3. The rig and its mapping to the framework are shown in Figure 12.

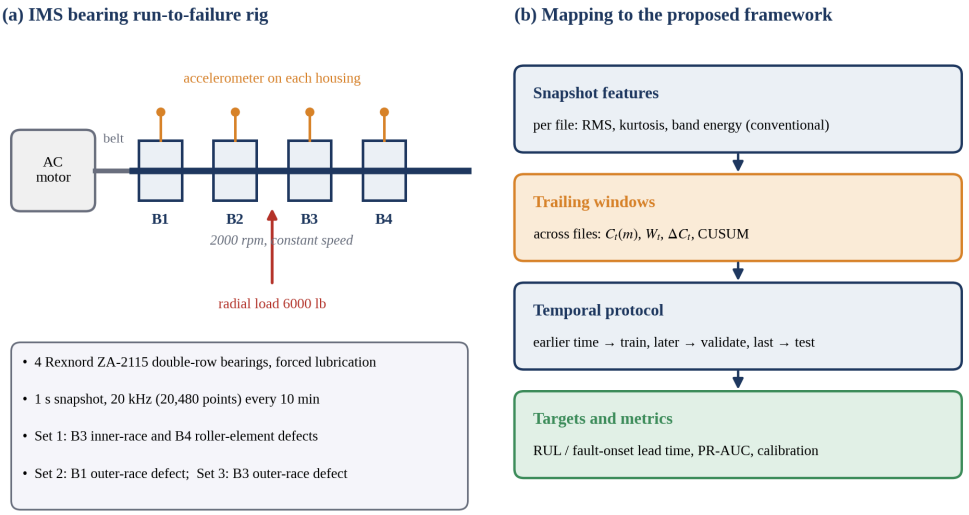


Figure 12. IMS bearing run to failure rig and the way its snapshots map onto the proposed framework (dataset facts follow Lee et al., 2007, and Qiu et al., 2006, and the schematic is the authors' own drawing)

Figure 12 shows in panel (*a*) the IMS test rig, with an AC motor driving a shaft that carries four bearings, accelerometers on each housing, and a radial load of 6000 lb at a constant 2000 rpm, together with a summary of the three test sets. Panel (b) maps the data onto the framework, with snapshot features such as RMS, kurtosis, and band energy forming the conventional channel, trailing windows computing C_t(m), W_t, ΔC_t, and CUSUM across files forming the summability channel, a temporal protocol that trains on earlier data and tests on the latest, and targets defined as RUL or fault onset lead time evaluated with PR AUC and calibration. An asset wise arrangement is natural, tuning on one test, validating on another, and reporting the third once. Evidence from this anchor would address hypotheses H_1 to H_3 in Table 8 directly, but no such experiment is reported here.

For plants in Nepal and similar settings, the most defensible pilot sites are critical, instrumented assets covered by a work order process, such as water supply pumps, fans, compressors, and bearing heavy drives. The closed learning loop that such pilots should follow is shown in Figure 13.

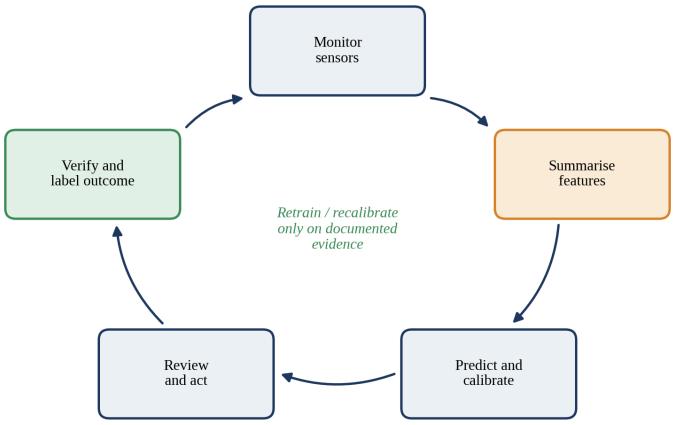


Figure 13. Closed loop maintenance learning cycle

Figure 13 presents maintenance as a continuous cycle that moves from monitoring sensors to summarizing features, predicting and calibrating risk, reviewing and acting, and finally verifying and labeling the outcome before monitoring resumes. The note at the center states that retraining or recalibration should occur only on documented evidence. Without post maintenance evidence, false alarms, missed failures, and successful interventions cannot be distinguished.

4.6 Comparative Analysis of Feature Families

Summability is a feature construction strategy rather than an alternative to machine learning, so the relevant question is whether the additional features improve robustness, lead time, or decision cost beyond raw and conventional features. The strengths and weaknesses of the main feature families are summarized in Table 6.

Table 6. Feature family comparison

Feature Family	Strength	Weakness	Best Role
Raw latest values	Immediate response	Sensitive to spikes and noise	Fast alarms and transient detection
Window statistics	Compact and interpretable	Can hide temporal order	Baseline tabular machine learning
Summability assisted	Stable level and drift summaries	Window and weight sensitivity	Persistent degradation signals
Frequency domain	Captures periodic and impulsive signatures	Needs suitable sampling and expertise	Rotating machinery diagnosis
Sequence models	Learn complex temporal patterns	Data hungry and harder to audit	Long, dense histories
Hybrid representation	Balances trend and transients	More features and validation effort	Critical assets with mixed fault modes

Table 6 shows that each feature family has a distinct strength and a matching weakness. Raw values respond immediately but are sensitive to spikes, summability features give stable drift summaries but depend on window and weight choices, frequency domain features capture impulsive signatures but require suitable sampling, and sequence models learn complex patterns but are data hungry and harder to audit (Serradilla, Zugasti, Rodriguez, & Zurutuza, 2022). The hybrid representation, which combines several families, balances trend and transient detection at the cost of more validation effort and is therefore best suited to critical assets with mixed fault modes. A fair experiment uses identical splits, labels, horizons, model families, and tuning budgets when comparing these feature sets, and if performance improves only on random splits but not out of time, the likely explanation is leakage or overfitting (Kapoor & Narayanan, 2023). Interpretable features do not make a tree ensemble explainable, so explanation should combine model analysis with engineering logic (Barredo, Díaz-Rodríguez, Bennetot, Tabik, Barbado, Garcia, Gil-Lopez, Molina, Benjamins, Chatila, & Herrera, 2020; Rudin, 2019).

4.7 Reliability, Economics, and Implementation Roadmap

For a repairable system under simple steady state assumptions, inherent availability is defined in Formula 6, where MTBF is the mean time between failures and MTTR is the mean time to repair.

$$A = \frac{MTBF}{MTBF + MTTR} \tag{6}$$

PdM may improve availability by reducing unplanned downtime, but unnecessary interventions can increase planned downtime, so net benefit should be estimated at the policy level as avoided failure loss and savings minus implementation, false alarm, and added intervention costs. The U.S.

Department of Energy estimates that a well run predictive program can save roughly 8% to 12% relative to a program based on preventive maintenance alone, while noting the up front investment required (Sullivan et al., 2010), but such a range is only a reference for checking a business case. Thresholds should also reflect asset criticality. A staged implementation roadmap is summarized in Table 7.

Table 7. Deployment stages and exit criteria

Stage	Key Activity	Evidence Required to Proceed
1. Scope	Asset, failure mode, horizon, and risk	Approved use case and consequence analysis
2. Instrument	Sensor and historian audit	Reliable units, timestamps, and sampling
3. Baseline	Data profiling and label review	Documented data quality report
4. Offline test	Feature and model comparison	Out of time metrics and ablation
5. Shadow mode	Human reviewed alerts	Acceptable lead time and alert burden
6. Controlled use	CMMS integration and escalation	Approved procedures and rollback plan
7. Monitor	Drift and outcome review	Scheduled governance and ownership

Table 7 divides deployment into seven stages, each with a key activity and an explicit exit criterion that must be met before the next stage begins. The first three stages define the asset, audit sensors, and profile data quality before any model is trained. The fourth stage compares raw, statistical, and summability feature groups offline with out of time metrics and ablation, and the fifth issues recommendations to engineers in shadow mode without triggering work orders, so that false positives, missed events, lead time, and workload can be reviewed safely. Only then are approved alerts integrated with the CMMS, and the final stage monitors drift and outcomes, with retraining triggered by documented evidence (Lu et al., 2019). The table therefore turns the framework into a gated roadmap in which no step depends on unproven assumptions.

4.8 Discussion, Governance, and Testable Hypotheses

The framework's central premise is modest. Stable summaries may help machine learning models distinguish persistent degradation from random variation, but only when they complement rather than replace raw signals. The strongest case for summability features is a process in which meaningful deterioration evolves over multiple samples and the noise contains short term fluctuations, and the weakest case is a fault whose critical evidence is a single transient or a high frequency signature erased by averaging. The operators are inexpensive and suit plants with limited infrastructure or edge devices (Njor et al., 2024), but centered windows, globally fitted scalers, and random splits of overlapping windows can all leak information (Kapoor & Narayanan, 2023), and the horizon must match maintenance logistics.

PdM recommendations affect worker safety and asset integrity, so they should be governed as decision support, and an alert must never bypass lockout and tagout procedures, statutory inspections, or a qualified engineer. Data and model governance should cover access control, cybersecurity, validation scope, and retraining criteria, and a reproducible evaluation should preserve the dataset version, preprocessing configuration, window conventions, seeds, split dates, and the locked test report. To guide future empirical work, five testable hypotheses are proposed in Table 8.

Table 8. Proposed hypotheses and falsification criteria

Hypothesis	Test	Evidence Against the Hypothesis
H_1. Summability features improve out of time recall at a fixed alert burden	Compare feature ablation under the same temporal splits	No improvement, or improvement disappears out of time

Hypothesis	Test	Evidence Against the Hypothesis
H_2. Weighted summaries detect gradual drift earlier than raw thresholding	Compare lead time distributions on labeled events	No earlier actionable lead time
H_3. Parallel raw and smooth features outperform smooth only features on transient faults	Stratify evaluation by fault type	Raw plus smooth does not reduce missed transients
H_4. Calibrated risk thresholds reduce expected maintenance cost	Back test the decision policy using documented costs	Cost rises or safety constraints are violated
H_5. Local calibration degrades under regime shift without monitoring	Evaluate by operating regime and later periods	Calibration remains stable across all shifts

Table 8 states five hypotheses that are deliberately phrased without assuming a positive result, each paired with a test and an explicit criterion for evidence against it. H_1 to H_3 concern the predictive contribution of summability features, namely recall at a fixed alert burden, earlier detection of gradual drift, and the value of keeping raw features for transient faults, and they can be tested directly on the IMS bearing data. H_4 concerns the economic value of calibrated thresholds, and H_5 concerns the expected degradation of calibration under regime shift, which motivates continuous monitoring. Stating falsification criteria guards against confirmation bias, because a negative result is also informative. The checklist that summarizes the reporting requirements of the framework is presented in Table 9.

Table 9. Predeployment and reporting checklist

Item	Required Practice	Status for this Framework Study
Data provenance	State the source, license, and whether data are synthetic	Specified, no new dataset claimed
Target definition	Failure mode and horizon are explicit	Defined conceptually
Feature causality	Use only past and present observations	Required by design
Baseline comparison	Raw and conventional features included	Recommended protocol
Temporal evaluation	Chronological or asset wise holdout	Specified
Metrics	Recall, precision, PR AUC, calibration, lead time, and cost	Specified
Figures and tables	Captions and in text discussion	Included
Empirical claims	Only measured results are reported	No field performance claimed
Governance	Human review, audit, rollback, and drift monitoring	Specified
Limitations	Benchmark and deployment limitations disclosed	Included

Table 9 lists ten reporting and deployment requirements together with the status of each in the present study. Data provenance, temporal evaluation, metrics, and governance are fully specified, feature causality is required by design, and baseline comparison is recommended as a protocol, while the target definition remains conceptual because no specific plant is studied. The row on empirical claims records that no field performance is claimed, which is consistent with the methodological nature of the study. The checklist is reusable in pilot plans, although it documents assumptions rather than certifying safety.

5. Conclusions

This study presented a summability assisted machine learning framework for the predictive maintenance of industrial equipment. The framework uses finite window means, weighted summaries, adjacent window differences, cumulative deviations, and robust dispersion alongside raw and conventional features, so that persistent condition changes can be captured while transient fault signatures are preserved. It places feature construction within a complete lifecycle that includes sensor quality assurance, temporally valid training, calibrated prediction, cost and safety aware decisions, human review, CMMS integration, and feedback from inspections and repairs, and it recommends evaluation with precision recall measures, calibration, lead time, alert burden, and expected cost under out of time testing and ablation. The synthetic illustrations show that summability rules can detect slow drift that a single sample limit misses, that window length governs a trade off between false alerts and delay, and that a calibrated risk estimate can be turned into a transparent inspection decision. The principal conclusion is conditional, because summability features are a low cost and interpretable candidate for stabilizing noisy telemetry, but their value must be demonstrated for each asset class and decision horizon.

Acknowledgement

The authors thank the Faculty of Science, Technology, and Engineering, Rajarshi Janak University, Janakpurdham, Nepal, for its academic support, and acknowledge the providers of the public datasets discussed in this study, including the NASA Prognostics Center of Excellence and the Center for Intelligent Maintenance Systems, University of Cincinnati.

Author Contributions

AS contributed to the conceptualization of the study, methodology, mathematical formulation, simulation and visualization, and the writing, review, and editing of the manuscript. PKN contributed to the conceptualization of the study, methodology, mathematical formulation, simulation and visualization, and the writing, review, and editing of the manuscript. SKS contributed to the conceptualization of the study, methodology, mathematical formulation, simulation and visualization, and the writing, review, and editing of the manuscript. All authors read and approved the final manuscript.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this article. The study received no specific funding from any agency, and the authors have no financial, commercial, or personal relationships that could have influenced the work reported in this paper.

References

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58, 82–115*. https://doi.org/10.1016/j.inffus.2019.12.012

Benatia, M. A., Hafsi, M., & Ben Ayed, S. (2025). A continual learning approach for failure prediction under non stationary conditions: Application to condition monitoring data streams. *Computers & Industrial Engineering, 204, Article 111049*. https://doi.org/10.1016/j.cie.2025.111049

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering, 137, Article 106024*. https://doi.org/10.1016/j.cie.2019.106024

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

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Guidotti, D., Pandolfo, L., & Pulina, L. (2025). A systematic literature review of supervised machine learning techniques for predictive maintenance in Industry 4.0. *IEEE Access, 13, 102479–102504*. https://doi.org/10.1109/ACCESS.2025.3578686

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73, 220–239*. https://doi.org/10.1016/j.eswa.2016.12.035

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483-1510. https://doi.org/10.1016/j.ymssp.2005.09.012

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine learning based science. *Patterns*, *4*(9). https://doi.org/10.1016/j.patter.2023.100804

Kushwaha, J. K., Thakur, A. K., Sahani, S. K., & Suryawanshi, H. (2023). Existence and convergence of gamma matrices and their application to infinite series. *The Mathematical Education*, *57*(2), 17-30.

Lee, J., Qiu, H., Yu, G., Lin, J., & Rexnord Technical Services. (2007). *Bearing data set [Data set]*.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing, 104, 799–834*. https://doi.org/10.1016/j.ymssp.2017.11.016

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing, 138, Article 106587*. https://doi.org/10.1016/j.ymssp.2019.106587

Li, X., Ding, Q., & Sun, J. Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety, 172, 1–11*. https://doi.org/10.1016/j.ress.2017.11.021

Liu, Z., & Hui, J. (2024). Advancing predictive maintenance: A deep learning approach to sensor and event log data fusion. *Sensor Review*, *44*(5), 563-574. https://doi.org/10.1108/SR-03-2024-0183

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346-2363. https://doi.org/10.1109/TKDE.2018.2876857

Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. Paper presented at In Proceedings of the Third International Conference on Artificial Intelligence for Industries (AI4I) (pp. 69–74). IEEE.

Murtaza, A. A., Saher, A., Zafar, M. H., Moosavi, S. K. R., Aftab, M. F., & Sanfilippo, F. (2024). Paradigm shift for predictive maintenance and condition monitoring from Industry 4.0 to Industry 5.0: A systematic review, challenges and case study. *Results in Engineering, 24, Article 102935*. https://doi.org/10.1016/j.rineng.2024.102935

Njor, E., Hasanpour, M. A., Madsen, J., & Fafoutis, X. (2024). A holistic review of the TinyML stack for predictive maintenance. *IEEE Access, 12, 184861–184882*. https://doi.org/10.1109/ACCESS.2024.3512860

Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2022). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, *54*(2). https://doi.org/10.1145/3439950

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks, 113, 54–71*. https://doi.org/10.1016/j.neunet.2019.01.012

Pokharel, J. K., Pahari, N. P., & Sahani, S. K. (2023). Critical analyzing on some new application of almost decreasing sequence to Legendre series associated with [B] sum. *Advances in Nonlinear Variational Inequalities*, *26*(2), 36-40. https://doi.org/10.52783/anvi.v26.253

Qiu, H., Lee, J., Lin, J., & Yu, G. (2006). Wavelet filter based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration, 289(4–5), 1066–1090*.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206-215. https://doi.org/10.1038/s42256-019-0048-x

Sahani, S. K., & Das, G. (2023). Generalized almost matrix summability of Fourier series and its conjugate series. *Tuijin Jishu/Journal of Propulsion Technology*, *43*(3), 2635-2655.

Sahani, S. K., & Mishra, L. N. (2021). Degree of approximation of signals by Nörlund summability of derived Fourier series. *The Nepali Mathematical Sciences Report*, *38*(2), 13-19. https://doi.org/10.3126/nmsr.v38i2.42845

Sahani, S. K., & Mishra, V. N. (2020). A certain study on Nörlund summability of series. *Journal of Linear and Topological Algebra*, *9*(4), 311-321.

Sahani, S. K., & Mishra, V. N. (2023). Degree of approximation of functions by Nörlund summability of double Fourier series. *Mathematical Sciences and Applications E Notes*, *11*(2), 80-88. https://doi.org/10.36753/mathenot.1008750

Sahani, S. K., & Thakur, A. K. (2020). Some ideas of Cesàro sequence spaces through summability in functional analysis. *NeuroQuantology*, *18*(12), 54-60. https://doi.org/10.48047/nq.2020.18.12.NQ20238

Sahani, S. K., Kumar, A., & Dubey, R. P. (2023). An estimate of the rate of convergence for the absolute summability of factors of infinite series. *Power System Technology*, *47*(4), 359-370.

Sahani, S. K., Mishra, V. N., & Pahari, N. P. (2020). On the degree of approximation of functions by N means of their Fourier Laguerre series. *Nepal Journal of Mathematical Sciences, 1, 65–70*. https://doi.org/10.3126/njmathsci.v1i0.34164

Sahani, S. K., Mishra, V. N., & Pahari, N. P. (2021). Some problems on approximations of functions (signals) in matrix summability of Legendre series. *Nepal Journal of Mathematical Sciences*, *2*(1), 43-50. https://doi.org/10.3126/njmathsci.v2i1.36566

Sahani, S. K., Mishra, V. N., & Rathour, L. (2022). On Nörlund summability of double Fourier series. *Open Journal of Mathematical Sciences*, *6*(1), 99-107. https://doi.org/10.30538/oms2022.0180

Sahani, S. K., Paudel, G. P., & Thakur, A. K. (2022). On a new application of positive and decreasing sequence to double Fourier series associated with (N, $p^{(1)}$, $p^{(2)}$). *Journal of Nepal Mathematical Society*, *5*(1), 58-64. https://doi.org/10.3126/jnms.v5i2.50011

Sahani, S. K., Paudel, G. P., Ghimire, J. L., & Thakur, A. K. (2022). On certain series to series transformation and analytic continuations by matrix methods. *Nepal Journal of Mathematical Sciences*, *3*(1), 75-80.

Sahani, S. K., Prasad, K. S., & Thakur, N. P. (2022). On a new application of almost non increasing sequence to ultraspherical series associated with (N, p, q)k means. *Applied Science Periodical*, *24*(1), 1-11. https://doi.org/10.5281/zenodo.7813613

Sahani, S. K., Thakur, A. K., & Dubey, A. (2022). Degree of approximation of conjugate of function belonging to (N, p, q) class by matrix summability means of conjugate Fourier series. *Mathematical Statistician and Engineering Applications*, *71*(4), 321-331.

Sahani, S. K., Thakur, A. K., & Kushwaha, K. J. (2021). On a new application of Fourier series by Borel's method. *NeuroQuantology*, *19*(12), 81-85. https://doi.org/10.14704/nq.2021.19.12.NQ21199

Sahani, S. K., Thakur, A. K., & Sahu, S. K. (2022). On a new application of summation of Jacobi series by B method. *Journal of Algebraic Statistics*, *13*(3), 3261-3268.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run to failure simulation. Paper presented at In Proceedings of the International Conference on Prognostics and Health Management (PHM08). IEEE.

Serradilla, O., Zugasti, E., Rodriguez, J., & Zurutuza, U. (2022). Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. *Applied Intelligence*, *52*(10), 10934-10964. https://doi.org/10.1007/s10489-021-03004-y

Sullivan, G. P., Pugh, R., Melendez, A. P., & Hunt, W. D. (2010). *Operations and maintenance best practices: A guide to achieving operational efficiency (Release 3.0)*.

Syed, M. A. B., Hasan, M. R., Chowdhury, N. I., Rahman, M. H., & Ahmed, I. (2025). A systematic review of time series algorithms and analytics in predictive maintenance. *Decision Analytics Journal, 15, Article 100573*. https://doi.org/10.1016/j.dajour.2025.100573

Tsallis, C., Papageorgas, P., Piromalis, D., & Munteanu, R. A. (2025). Application wise review of machine learning based predictive maintenance: Trends, challenges, and future directions. *Applied Sciences*, *15*(9). https://doi.org/10.3390/app15094898

Yadav, D. K., Kaushik, A., & Yadav, N. (2024). Predicting machine failures using machine learning and deep learning algorithms. *Sustainable Manufacturing and Service Economics, 3, Article 100029*. https://doi.org/10.1016/j.smse.2024.100029

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (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

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering, 150, Article 106889*. https://doi.org/10.1016/j.cie.2020.106889