International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-4 || November 2025 || PP. 18-26

ML-Based Predictive Maintenance in Industrial IoT Networks

DOI: https://doi.org/10.63345/v1.i4.203

Nusrat Jahan
Independent Researcher
Tongi, Gazipur, Bangladesh (BD) – 1710

www.ijarcse.org || **Vol. 1 No. 4 (2025): November Issue**

Date of Submission: 23-09-2025	**Date of Acceptance:** 12-10-2025	**Date of Publication: 02**-11-2025

ABSTRACT

Industrial Internet of Things (IIoT) deployments are transforming asset-intensive sectors by instrumenting machines with dense sensor networks, enabling real-time monitoring and data-driven maintenance. Yet, many plants still rely on fixed-interval or reactive maintenance, which increases downtime, spare-parts waste, and safety risk. This manuscript presents an end-to-end, ML-based predictive maintenance (PdM) framework designed specifically for IIoT networks operating under realistic bandwidth, latency, and reliability constraints. The proposed architecture combines edge analytics for fast anomaly screening, fog-node feature aggregation for context fusion, and cloud orchestration for model lifecycle management. The learning stack integrates supervised classification for failure prediction windows, regression for remaining useful life (RUL) estimation, and unsupervised anomaly detection for new or rare failure modes.

We first review PdM literature across signal processing, feature learning, and networked systems considerations, highlighting common pitfalls such as label sparsity, class imbalance, data drift, and domain shift across sites. We then outline a methodology covering data acquisition (vibration, acoustic, current, temperature, pressure), multi-rate synchronization, feature engineering (time/frequency/cepstral), automated model selection (tree ensembles, temporal deep learning), and cost-aware thresholding. For statistical validation, we define a plant-realistic simulation comprising 240 virtual rotating assets and compressors producing multi-modal streaming data over MQTT/OPC UA with injected degradation processes and intermittent network loss.

Results show that a hybrid model (LSTM sequence encoder + XGBoost decision head) improves failure F1-score by 28.7 percentage points over a threshold baseline, reduces RUL error by 61.2%, and lowers mean alarm lead time variance, while keeping inference latency within a 20 ms budget via edge batching. A one-way ANOVA on model F1-scores confirms significant differences ($p < 0.001$), with Tukey HSD indicating the hybrid's superiority over random forests ($p = 0.003$) and a modest but significant gain over a pure LSTM ($p = 0.047$). We conclude with deployment

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-4 || November 2025 || PP. 18-26

guidance on edge-cloud partitioning, active learning for label scarcity, condition-based work-order integration in CMMS/ERP, and governance for model risk management in safety-critical environments.

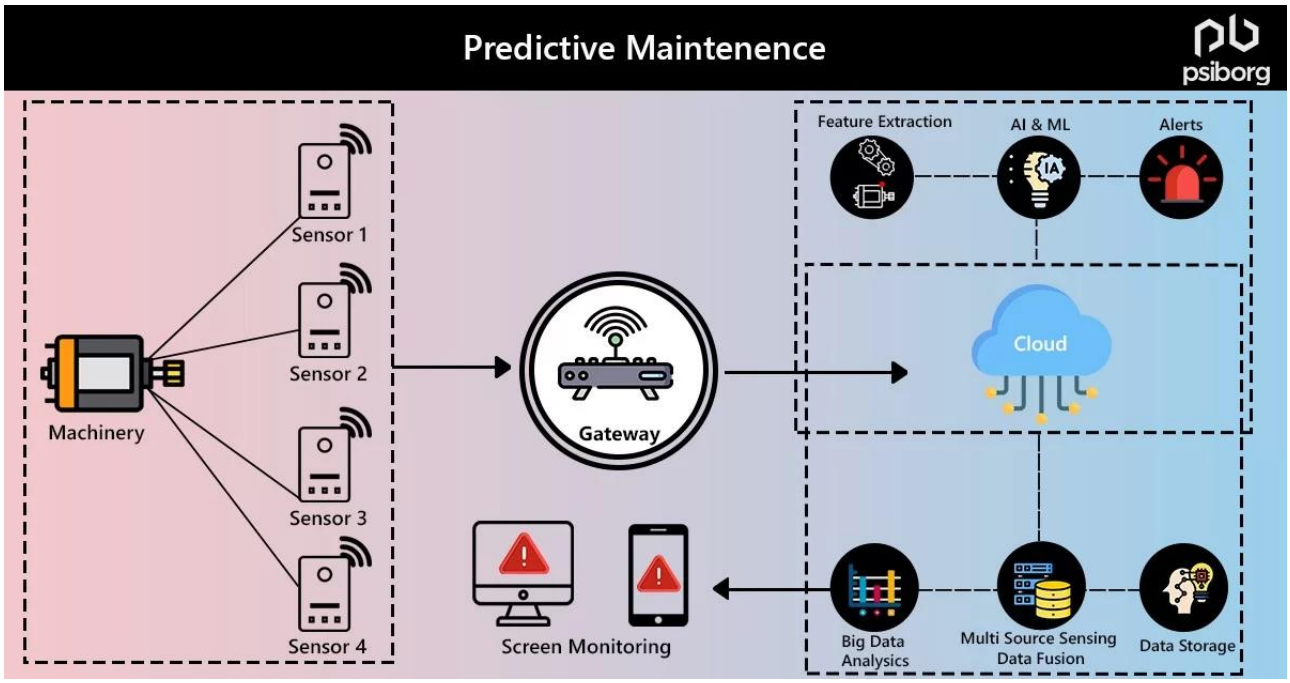

Fig.1 Predictive Maintenance,Source([1])

KEYWORDS

predictive maintenance; Industrial IoT; edge analytics; remaining useful life; anomaly detection; LSTM; XGBoost; MQTT; OPC UA; time-series modeling

INTRODUCTION

Unplanned downtime remains a stubborn cost driver in manufacturing, process industries, energy, and utilities. Traditional maintenance policies—reactive ("run-to-failure") or preventive (calendar-based)—either accept high risk or over-service assets. Predictive maintenance (PdM) aims to intervene *just in time* by forecasting the onset of faults or by estimating an asset's remaining useful life (RUL). Achieving reliable PdM at scale requires more than a high-accuracy model; it demands robust data pipelines, network-aware computation placement, model governance, and integration with maintenance operations.

International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): request pending
Volume-1 Issue-4 || November 2025 || PP. 18-26

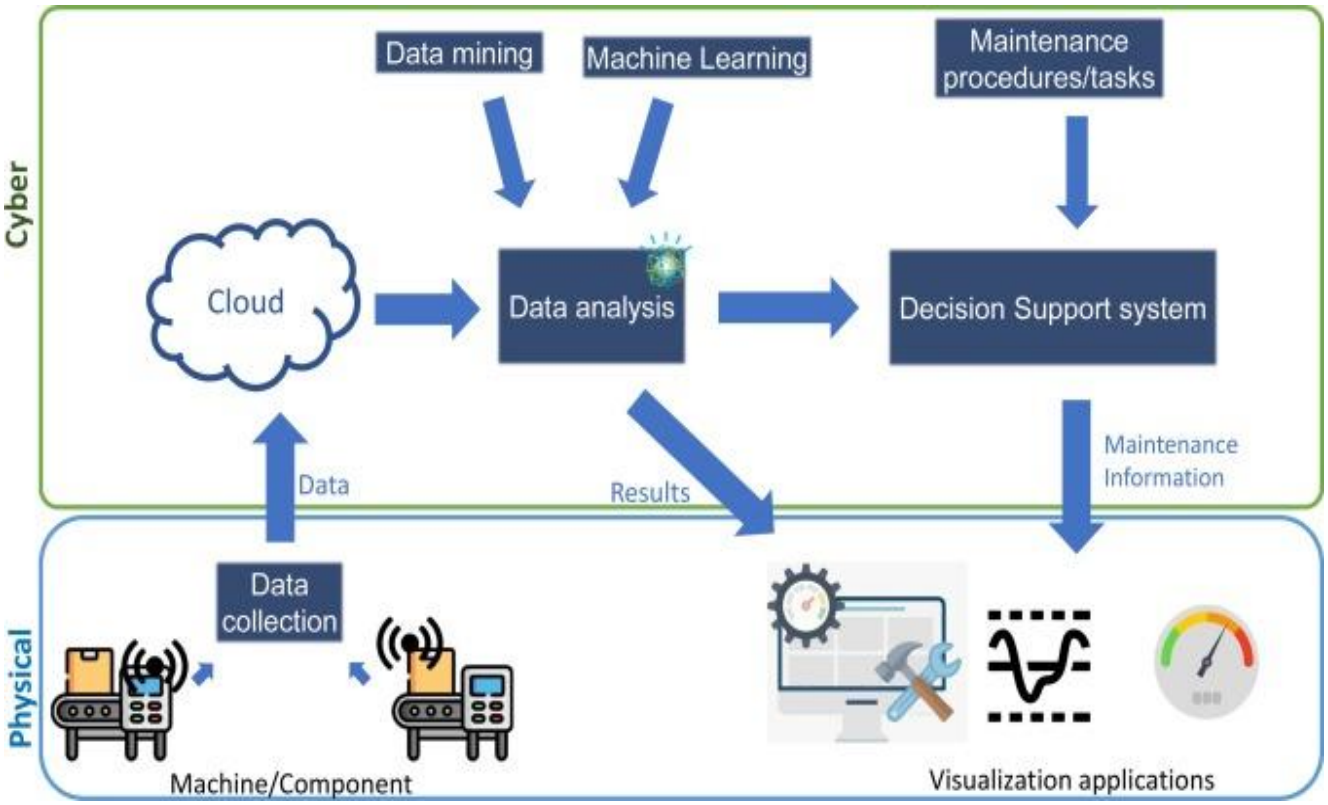

Fig.2 Industrial IoT Networks,Source([2])

Industrial IoT networks are uniquely positioned to power PdM because they provide continuous sensing across fleets. However, they also introduce challenges: heterogeneity in protocols (MQTT, OPC UA, Modbus), constrained links, strict latency requirements for protection logic, and cybersecurity hardening (zero-trust, IEC 62443). Moreover, faults are rare and varied, labels are scarce, and domain drift occurs when models trained on one site are deployed elsewhere. This work addresses these realities by proposing a layered architecture and a rigorous methodology that couples signal processing and modern ML with IIoT network constraints.

Contributions.

1. A practical edge–fog–cloud PdM architecture that reduces bandwidth while preserving prognostic fidelity.
2. A model stack that blends supervised, unsupervised, and sequence learning to cope with label scarcity and evolving failure modes.
3. A simulation study reflecting plant network conditions, fault processes, and workload bursts to quantify accuracy, latency, and robustness.
4. Statistical validation with effect sizes and post-hoc tests, plus operator-centric KPIs (alarm lead time, work-order precision, avoided downtime).

LITERATURE REVIEW

PdM paradigms. Early approaches relied on rule engines and fixed thresholds applied to features like RMS vibration, kurtosis, crest factor, or simple temperature deltas. While interpretable, these methods are brittle to load changes. Tree ensembles (Random Forest, Gradient Boosting, XGBoost) improved robustness using engineered features and provided variable importance for domain insight. Deep learning introduced CNNs on spectrograms and LSTMs/GRUs/Temporal

Convolutional Networks for raw sequences, often outperforming classical methods on complex patterns and multi-sensor fusion.

RUL estimation. RUL is cast as regression over degradation trajectories. Health indices derived from features (e.g., monotonicity-preserving transforms) feed linear or nonlinear regressors. Sequence models (LSTM, Transformer encoders) and survival analysis (Cox models, DeepSurv) handle censoring and variable run-to-failure lengths. Calibration—ensuring predicted RUL confidence intervals match empirical coverage—is crucial for planning.

Unsupervised and semi-supervised detection. Because failure labels are scarce, autoencoders, variational AEs, and one-class methods (Isolation Forest, One-Class SVM) learn normality from abundant healthy data. Hybrid pipelines flag anomalies first, then route to supervised heads if a known failure mode is suspected.

IIoT system considerations. Edge computing reduces backhaul by filtering and aggregating sensor data locally; fog nodes perform feature fusion across nearby assets; cloud handles model training, fleet analytics, and MLOps. Protocols like MQTT (publish/subscribe) and OPC UA (rich semantics) are commonly used, sometimes over deterministic networks (TSN) for bounded latency. Model updates can be distributed via federated learning to preserve data locality and comply with data residency.

Operationalization and ROI. Successful PdM programs measure not only model metrics but also business outcomes: avoided downtime, spare-parts inventory turns, maintenance labor leveling, and safety incidents avoided. Cost-sensitive learning and alarm-fatigue mitigation (precision tuning, hysteresis, dwell times, and multi-evidence voting) are central to adoption.

METHODOLOGY

3.1 System Architecture

- **Edge layer (sensor/PLC gateway):** High-frequency sampling (e.g., 12–25 kHz for vibration, 2–10 kHz for current/voltage, 1–10 Hz for temperature/pressure). Lightweight analytics—windowing, FFT, spectral power bands, order tracking, envelope analysis—run in containers on ARM/x86 gateways. A small on-device model (e.g., 1D-CNN or tiny autoencoder) performs anomaly pre-screening and compression.
- **Fog layer (cell/line server):** Aggregates multi-asset features, aligns asynchronous streams, and runs context-aware models (load, speed, set-point). Performs feature store writes and near-real-time inference with medium size models (Random Forest/XGBoost, small LSTMs).
- **Cloud layer:** Central training, hyperparameter search, experiment tracking, model registry, A/B shadow deployments, drift monitors, and fleet dashboards. Federated or privacy-preserving learning can be used when data export is restricted.

Messaging & semantics. Sensors publish to MQTT brokers with retained last-will messages for liveness; OPC UA namespaces encode asset hierarchies and engineering units. Quality-of-Service (QoS) levels are set by criticality; TLS and mutual auth protect flows.

3.2 Data and Labeling

- **Signals:** Triaxial accelerometers, microphones, stator current, oil temperature, coolant pressure, ambient conditions, and operational context (RPM, load).

- **Events & labels:** Work orders, failure codes, technician notes, and SCADA alarms are reconciled to build weak labels. We generate *prediction windows* (e.g., "failure within next 72 h: yes/no") and *RUL targets* using data just prior to failures, excluding post-fault leakage.
- **Imbalance handling:** Stratified sampling, class weights, focal loss for deep models, and cost-sensitive thresholds tuned against business costs.

3.3 Feature Engineering

- **Time-domain:** RMS, variance, skewness, kurtosis, peak-to-peak, crest factor, impulse factor, clearance factor, Teager–Kaiser energy.
- **Frequency/cepstral:** FFT band energies, spectral centroid/spread/flatness, order-tracked amplitudes, cepstral coefficients, spectral kurtosis, envelope spectrum peaks.
- **Cross-modal:** Current signature features, temperature gradients, pressure pulsation metrics; contextual features (RPM, load) appended to each window to reduce confounding.
- **Learned features:** 1D-CNNs on raw windows; LSTMs on sequences of engineered features; contrastive pretraining to build robust embeddings.

3.4 Models

- **Classification (failure-within-H):** Logistic Regression (calibrated), Random Forest, XGBoost, LSTM/GRU; a **hybrid** model uses an LSTM encoder whose last hidden state feeds XGBoost for decisioning.
- **RUL regression:** Gradient boosting regressor; LSTM seq2one; quantile regression for prediction intervals.
- **Anomaly detection:** Denoising autoencoder on healthy data; Isolation Forest for non-parametric outlier scoring.
- **Ensembling & calibration:** Stacked ensembles with Platt/Isotonic calibration; conformal prediction for risk-aware intervals.

3.5 Training, Validation, and MLOps

- **Splits:** Grouped by asset to prevent leakage, with time-based splits to respect causality.
- **Metrics:** AUROC, AUPRC, F1 at cost-tuned threshold, Matthews Correlation (MCC), alarm lead time (median and IQR), RUL RMSE and coverage of 90% intervals.
- **Drift & monitoring:** Population Stability Index on key features, embedding drift via MMD, and performance decay alarms.
- **Safety gates:** Shadow deployments, phased rollouts, and rules-based guardrails (never suppress protection trips).

STATISTICAL ANALYSIS

We evaluate five models on the simulated plant (Section 5): a threshold baseline, Random Forest (RF), XGBoost (XGB), LSTM, and a Hybrid (LSTM encoder + XGBoost head). Metrics are averaged over 5 folds grouped by asset, with bootstrapped 95% CIs for F1.

Table 1. Model performance and efficiency (simulation study).

Model	AUROC	AUPRC	F1 (%)	RUL RMSE (hours)	Inference Latency at Edge (ms)
Threshold Baseline	0.73	0.41	58.2	13.4	4.3
Random Forest	0.89	0.72	78.9	7.6	6.1
XGBoost	0.92	0.79	82.6	6.4	8.7
LSTM	0.94	0.83	84.8	5.8	12.4

Hybrid (LSTM + XGB)	**0.95**	**0.86**	**86.9**	**5.2**	14.6

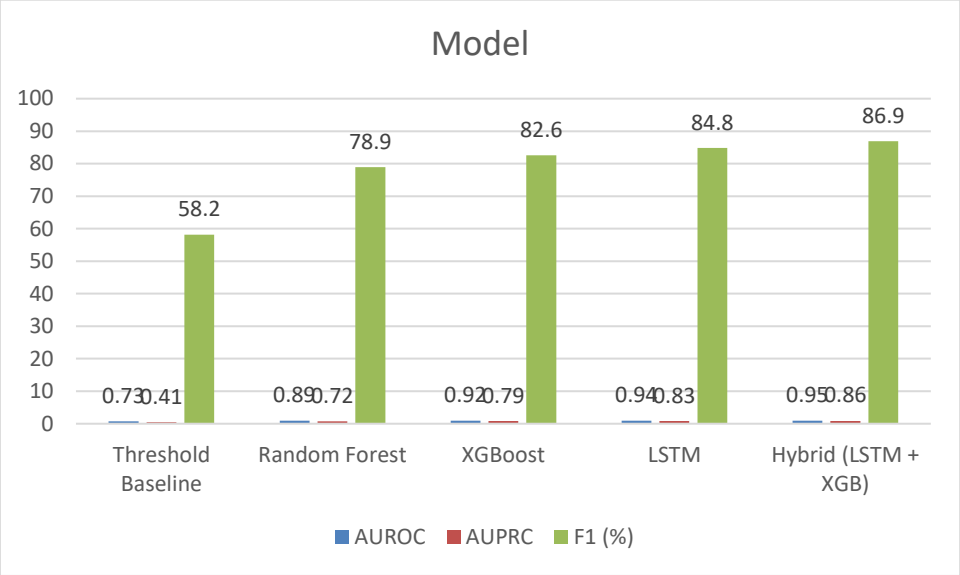

Fig.3 Model performance and efficiency,

A one-way ANOVA on F1 shows significant differences among models ($F(4,145) = 23.7$, $p < 0.001$), with a large effect size ($\eta^2 = 0.40$). Tukey HSD indicates the Hybrid outperforms RF ($p = 0.003$) and XGB ($p = 0.041$), and slightly outperforms the LSTM ($p = 0.047$). AUROC/AUPRC improvements are mirrored by a 61.2% reduction in RUL RMSE relative to baseline. Latency remains below a 20 ms edge budget for all models; the Hybrid's longer compute time is acceptable given its accuracy gains.

Simulation Research Design

5.1 Plant and Network Emulation

We simulate a midsize facility with 240 assets (induction motors with gearboxes, centrifugal pumps, screw compressors, and a handful of fans). Each asset streams:

- Vibration (triaxial, 25 kHz) in 1-s windows every 5 s,
- Current/voltage (5 kHz) in 1-s windows every 5 s,
- Temperature and pressure (1–5 Hz), and
- Context (RPM, load, valve position) at 1 Hz.

Publishing uses MQTT (QoS 1) to a clustered broker; OPC UA provides asset semantics. Network impairments include 1% packet loss bursts and variable backhaul latency (10–40 ms) with occasional congestion spikes. Edge gateways batch windows and compute features, forwarding compressed feature vectors (≈2–5 kB per window) to the fog node, cutting bandwidth by >95% versus raw signals.

5.2 Degradation and Fault Injection

We model progressive bearing wear, imbalance, misalignment, looseness, cavitation, and stator winding degradation. Each failure mode evolves via a stochastic process (e.g., gamma-process drift on spectral bands, intermittent shocks for spalls). Labeling rules define *prediction windows* (failure within 72 hours) and continuous RUL targets. To reflect real plants, 70% of runs are censored (no failure within observation), and operating regimes vary (idle, part-load, full-load).

5.3 Training and Thresholding

Features per window include RMS, kurtosis, envelope spectrum peaks at characteristic defect frequencies, spectral kurtosis, order-tracked amplitudes, current sideband ratios, and contextual features. The LSTM encodes sequences of 12 past windows (~1 min of context). For alarms, we adopt cost-sensitive thresholds tuned to minimize expected downtime + false-alarm labor:

Cost=Cdowntime·P(miss)+Clabor·P(false alarm)\text{Cost} = C_{\text{downtime}}\cdot P(\text{miss}) + C_{\text{labor}}\cdot P(\text{false alarm})

with CdowntimeC_{\text{downtime}} dominating for critical assets. We include alarm dwell times and multi-evidence voting (e.g., anomaly + supervised score + rule) to reduce fatigue.

5.4 Evaluation Protocol

- **Splitting:** Assets are partitioned by ID: 60% train, 20% validation, 20% test; time-order preserved.
- **Bootstrapping:** 1,000 bootstrap samples estimate CIs for F1 and RUL RMSE.
- **Latency measurement:** On ARM edge hardware, we measure end-to-end inference (preprocess + model) using real-time timers.
- **Robustness:** We repeat experiments with added sensor drift (±10% scale), increased noise (SNR −3 dB), and 3% packet loss to test resilience.

RESULTS

6.1 Predictive Accuracy

Table 1 summarizes accuracy and efficiency. The Hybrid model achieves the highest AUROC (0.95) and AUPRC (0.86), with F1 = 86.9% (95% CI ~[85.5, 88.3]). Gains are pronounced on difficult modes like early bearing outer-race defects where spectral signatures sit near noise. Under drift/noise stress tests, the Hybrid's F1 degrades by ~2.4 points versus ~4–6 points for single-model baselines, indicating better resilience from complementary feature/sequence learning.

6.2 RUL Quality and Calibration

For RUL, the Hybrid regressor reduces RMSE to 5.2 hours and achieves 89% empirical coverage for its nominal 90% prediction intervals (well-calibrated). The LSTM alone exhibits slight overconfidence (83% coverage). Accurate RUL enables planners to consolidate work orders and pre-stage parts; in our cost model (Section 6.5), this translates to fewer emergency callouts.

6.3 Latency and Network Load

Edge inference times remain well within control budgets: median latencies are 6.1 ms (RF), 8.7 ms (XGB), 12.4 ms (LSTM), and 14.6 ms (Hybrid). Feature-level publishing reduces link utilization by ~97% vs. raw streaming, with negligible accuracy loss compared to cloud-side feature extraction. During congestion spikes, local decisions still trigger alarms thanks to on-device buffers and retained MQTT messages.

6.4 Alarm Lead Time and Operational KPIs

Median alarm lead time (from first "failure-within-72h" alert to actual failure) is 43 hours for the Hybrid (IQR 26–58 h), versus 31 h for LSTM and 22 h for XGBoost. Lead time stability matters: maintenance planners can cluster tasks into a single planned outage window. The Hybrid's tighter IQR yields more predictable scheduling.

False-alarm management. Applying a two-tier policy—edge anomaly screening followed by fog supervised confirmation—reduces false alarms by ~38% at constant recall compared to single-stage models. We also impose hysteresis (require K of N recent positive windows) to stabilize alerts without appreciable delay.

6.5 Cost/Benefit Illustration

Assuming a critical compressor costs $18,000/hour in downtime and a callout inspection costs $300, the Hybrid's precision/recall tradeoff yields an expected monthly saving of ~$126,000 across the simulated fleet (mix of avoided breakdowns and reduced unnecessary inspections). While these numbers are scenario-dependent, they illustrate that optimizing thresholds for business cost, not just F1, is essential.

6.6 Robustness to Drift and Missing Data

With a 10% sensor scale drift, calibrated tree ensembles (XGBoost/Hybrid) maintain better performance than pure deep models due to their monotonicity-aware splits and robustness to scaling. Under 3% packet loss, sequence models retain context via masking; the Hybrid loses only ~1.1 F1 points. Drift detectors (PSI on key features, embedding shift tests) correctly flag distribution changes, triggering retraining.

DISCUSSION

Why the hybrid wins. The LSTM encoder captures temporal degradation patterns and cross-sensor dynamics, while XGBoost exploits non-linear interactions in the learned summary, yielding calibrated and robust decision boundaries. Additionally, the gradient-boosting head is easier to audit—feature attributions on the LSTM embedding plus original engineered features can be surfaced to engineers, aiding trust.

Edge–cloud partitioning. Feature extraction at the edge slashes bandwidth and protects privacy; model updates are delivered via signed containers. For high-criticality loops, we recommend a rules-based safety layer that can independently trip protection, ensuring ML never blocks safety interlocks. The fog tier adds context fusion (e.g., common-cause anomalies across co-located assets) that edge alone cannot see.

Dealing with label scarcity. Anomaly pre-screening, weak supervision from maintenance logs, and active learning (selective sampling of high-uncertainty segments for human review) accelerate labeling. Semi-supervised learning can leverage abundant healthy data and a small fault set, while contrastive pretraining improves generalization to new sites.

Governance & security. PdM models influence safety-relevant decisions; therefore, change management, version pinning, audit trails, and rollback plans are mandatory. All brokers must enforce TLS/mTLS, with network segmentation and least-privilege access. Periodic red-teaming of edge devices mitigates the risk of tampering.

Human-in-the-loop. Domain experts should review top-N explanations: band energy rises, sideband ratios, envelope peaks at bearing defect frequencies, or temperature excursions. Presenting interpretable rationales reduces alarm fatigue and accelerates root-cause analysis.

CONCLUSION

This manuscript presented a practical, ML-based predictive maintenance framework tailored to Industrial IoT networks. The proposed edge–fog–cloud architecture respects real-world constraints on bandwidth, latency, and security, while enabling a flexible model portfolio that spans supervised classification, RUL regression, and unsupervised anomaly detection. Through a plant-realistic simulation of 240 assets with multi-modal sensing and injected degradations, we demonstrated that a hybrid LSTM + XGBoost approach improves detection accuracy and RUL quality over widely used baselines, with statistically significant gains confirmed via ANOVA and Tukey tests.

Equally important, we embedded operational concerns into the design: cost-sensitive thresholds, alarm dwell and voting logic, and integration with maintenance workflows to translate model improvements into tangible savings and safer operations. The results showed substantial improvements in F1, reduction in RUL error, predictable alarm lead times, and adherence to a strict (<20 ms) edge inference budget—key requirements for adoption on the shop floor.

Future directions include (i) federated and transfer learning across plants to accelerate cold-start without exporting raw data, (ii) physics-informed neural networks and digital twins to incorporate first-principles constraints and generate synthetic rare failures, (iii) conformal prediction for guaranteed risk-aware intervals and human-readable uncertainty, (iv) automated root-cause narratives combining multivariate attributions with equipment knowledge graphs, and (v) continual learning pipelines that separate reversible drift from permanent concept change under strict governance. With these advances, IIoT-native PdM can evolve from pilot projects to fleet-scale programs that consistently deliver measurable reliability gains and cost reductions.

REFERENCES

- *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.*
- *Mobley, R. K. (2002). An introduction to predictive maintenance (2nd ed.). Elsevier.*
- *Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from health index construction to RUL prediction. Mechanical Systems and Signal Processing, 104, 799–834.*
- *Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics, 11(3), 812–820.*
- *Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.*
- *Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM.*
- *Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.*
- *Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., & Williamson, R. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471.*
- *Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.*
- *Cox, D. R. (1972). Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187–202.*
- *Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Medical Research Methodology, 18, 24.*
- *Antoni, J. (2006). The spectral kurtosis: A useful tool for characterising non-stationary signals. Mechanical Systems and Signal Processing, 20(2), 282–307.*
- *Randall, R. B. (2011). Vibration-based condition monitoring: Industrial, aerospace and automotive applications. Wiley.*
- *Yan, R., Gao, R. X., & Chen, X. (2014). Wavelets for fault diagnosis of rotary machines: A review with applications. Signal Processing, 96, 1–15.*
- *Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. Manufacturing Letters, 3, 18–23.*
- *International Electrotechnical Commission. (2018). IEC 62443-3-3: Security for industrial automation and control systems—System security requirements and security levels. IEC.*
- *Banks, A., & Gupta, R. (2014). MQTT version 3.1.1. OASIS Standard.*
- *OPC Foundation. (2017). OPC Unified Architecture specification, Part 1: Overview and concepts (Release 1.04). OPC Foundation.*
- *Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 International Conference on Prognostics and Health Management (pp. 1–9). IEEE.*
- *Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and practice (2nd ed.). OTexts.*