Predictive Maintenance of Industrial Equipment Using Machine Learning and IOT Data Analytics: A Context-Aware, Edge-Cloud Framework with Operator-in-the-Loop Adaptation

Jitam Saha[1], Shaleen Singh[2], Shreyanjan Neogi[3], Anirban Bhatta[4], Sampurna Majumder[5], Anuvab Sen[6], Upoma Mridha[7], Joyonto Dey[8]

[3]Computer Science (AI & ML) Institution: KIIT University Bhubaneswar, Odisha, India

[1]P G D M (ISA & Finance) Institution:FORE School of Management New Delhi, Delhi, India

[2]Computer Science and Engineering Institution: KIIT University Bhubaneswar, Odisha, India

[4]Computer Science (AI & ML) Institution: KIIT University Bhubaneswar, Odisha, India

[5]B. Tech Aerospace Engineering Institution: KIIT University Bhubaneswar, Odisha, India – 751024

[6]Computer Science and Engineering Institution: KIIT University Bhubaneswar, Odisha, India

[7]B-Tech CSE Institution: KIIT University Bhubaneswar, Odisha, India

[8]Computer Science and Engineering Institution: KIIT University Bhubaneswar, Odisha, India

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500164

Received: 13 May 2026; Accepted: 18 May 2026; Published: 08 June 2026

ABSTRACT

The reliable functioning of any manufacturing sys-tem presupposes smooth equipment performance; however, un-scheduled interruptions remain a major source of loss in terms of efficiency, safety, and costly maintenance expenses. Traditional approaches including proactive or reactive maintenance methods prove ineffective in highly dynamic and unpredictable production environments. While IoT technology-driven predictive maintenance offers superior alternatives, current solutions suffer from critical limitations concerning long response times, reliance on network infrastructure, fast model decay due to changing load regimes and aging systems, and low credibility and transparency of predictions. This paper presents a context-aware framework for predictive maintenance incorporating a hybrid cloud-edge architecture and adaptive maintenance techniques based on operator involvement. The proposed model uses real-time context data to continuously update its ability to detect anomalies and forecast gradual equipment deterioration.

The edge component carries out preliminary filtering of incoming raw data, performs feature engineering, and provides basic classification results, sending extracted contextual information about detected events to the cloud server for further processing. A key advantage is the ability to incorporate maintenance engineers' feedback as contextual information into the learning algorithm. Maintenance technicians provide additional validation, explanation, or cor-rection to alerts raised by the algorithm, helping adjust the model to changing conditions. Combined with an explanatory engine, the proposed framework translates identified multivariate factors into understandable failure mode identification, prob- abilities, and maintenance procedure prioritization. Results of testing in industrial settings show improved equipment efficiency metrics, including significantly decreased false alert numbers, reduced maintenance diagnosis cycles, and effective inventory management. The described predictive maintenance technique proves efficient for ensuring operational resilience and seamless integration into existing industrial practices.

Index Terms—Predictive Maintenance, IoT Data Analyt-ics, Edge-Cloud Architecture, Machine Learning Adaptation, Human-in-the-Loop Systems, Explainable Industrial AI

LITERATURE REVIEW

The evolution towards data-centric predictive maintenance has been greatly facilitated by the prevalence of industrial IoT networks. Contemporary manufacturing plants are outfitted with dense sensor arrays capable of logging high-frequency vi-bration, temperature, acoustic, and electrical data from rotating machines and process assets. Condition monitoring with this level of granularity is made possible through streaming data. In its early stages, the predominant approach was threshold-based alarm notification and rule-based expert systems, which were easy to understand but unable to represent highly multivariate degradation profiles. With the emergence of cloud-based data lakes, the ability to analyze historical trends and train models centrally laid the groundwork for modern PdM workflows.

Today, machine learning represents the primary method-ological pillar behind PdM solutions. Recent literature tends to focus on three broad approaches: supervised classifier training, unsupervised anomaly detection, and regression models for predicting remaining useful life. While deep learning algo-rithms have proven highly accurate in lab experiments, using convolutional and recurrent neural networks to extract hidden features from sensor data, industrial applications often struggle to maintain high levels of performance due to concept drift, operational changes not recorded in the training data, and unseen failure modes. Supervised models relying solely on historical data need offline retraining periodically, which is disruptive, costly, and difficult to manage. Moreover, cloud-based inference pipelines often face latency challenges that make them unusable in certain situations when it comes to time-sensitive fault isolation.

To tackle issues related to latency, some scholars have recently looked into applying edge computing principles that facilitate moving inference close to the data source. Running lightweight machine learning models directly on the industrial gateway allows preprocessing of sensor signals and performing feature engineering and early fault classification locally with-out any need for cloud connectivity. An edge-cloud hybrid approach, where heavy model training and cross-asset corre-lation analysis take place on the cloud side, represents an ap-pealing balance between cost and functionality. Unfortunately, most edge solutions currently operate independently and lack capabilities for contextualizing detected anomalies based on the wider production schedule, environmental conditions, or maintenance activities.

An important aspect of predictive maintenance which has not been fully covered by previous work pertains to human expertise. Industrial technicians are often faced with confusing alerts, black box predictions, and many false positives, all of which contribute to alert fatigue and undermine overall system trust. Although explainable AI methods such as fea-ture importance and visualization of outlier sensor readings may provide insights into what triggers alerts, few current solutions incorporate systematic maintenance feedback in the form of a continuous calibration loop. Existing frameworks see maintenance experts as passive recipients of notifications, thus ignoring the possibility of leveraging their domain expertise to refine models and mitigate the effects of concept drift.

Overall, current literature provides a robust theoretical framework and state-of-the-art solutions in terms of algorithm performance. What these efforts lack, however, is adaptability and continuous calibration, which hinder implementation in practice. This paper attempts to bridge this gap by introducing a self-calibrating framework for integrating real-time IoT analytics and machine learning models with incremental adap-tation, structured maintenance feedback, and comprehensive output explanation.

INTRODUCTION

Failure incidents related to industrial equipment lead to heavy losses in terms of both costs and downtime. Although maintenance is one of the ways to avoid these problems, traditional approaches and initial attempts at early predic-tive maintenance suffer from model degradation, high false-positive rates, and lack of contextual information, which makes it difficult to integrate such techniques in practice. Therefore, this paper offers a novel approach to predictive maintenance for industrial equipment that leverages IoT technology and machine learning algorithms with the help of context and constant interaction between the two. The proposed framework includes a unique self-calibrating edge-cloud architecture combined with an operator-in-the-loop feedback system. As a result, it enables adaptive fault detection and degradation forecasting without the need for periodic model retraining offline. The entire process of maintenance takes place within a hierarchical data pipeline with IoT sensors recording high-frequency vibrations, temperature, sounds, and power consumption of devices. After that, edge computing performs real-time signal processing, temporal analysis, and anomaly classification, ensuring low latency and efficient bandwidth utilization.

As soon as validated events are generated, they are aug-mented by various types of metadata that include production schedule, environmental factors, shift details, and recent main-tenance records, making it easier to transfer this information to cloud computing. Once in the cloud, the data go through an ensemble refinement phase, where multi-device patterns and long-term degradation are identified based on machine learning models. In addition, all models incorporate incremen-tal learning and concept drift detection features to maintain accuracy during changing production circumstances.

One of the significant innovations of this framework lies in its ability to integrate maintenance expert knowledge into the learning algorithm pipeline. Technicians use an open decision-support interface that displays machine-generated predictions together with the confidence score, as well as identifies the particular anomaly and suggests appropriate actions. These interactions are treated as calibration signals, allowing turning field insights into quantifiable model updates to decrease false alarms. Moreover, an explainable layer enables tracing each recommendation to sensor activity and contextual data. Overall, the proposed approach allows for significantly enhancing operational reliability through improved predictive accuracy, reduced fault cycles, better spare part optimization, and increased service life of machines.

METHODOLOGY

The proposed solution utilizes a hierarchical, context-aware edge-cloud architecture based on continuous IoT data acquisi-tion, incremental machine learning adaptation, and structured human-in-the-loop validation. The system is purposefully de-signed to function in challenging industrial environments featuring heterogeneous equipment fleets, varying production loads, legacy infrastructure restrictions, and stringent safety requirements. In this section, an overview of the end-to-end workflow will be provided, encompassing sensor data collection and pre-processing, cloud-based analytics, model calibrations, decision support for technicians, and deployment logistics. The methodology intentionally avoids batch-based model training strategies in favor of continuous self-calibration, ensuring algorithm outputs align with existing maintenance workflows.

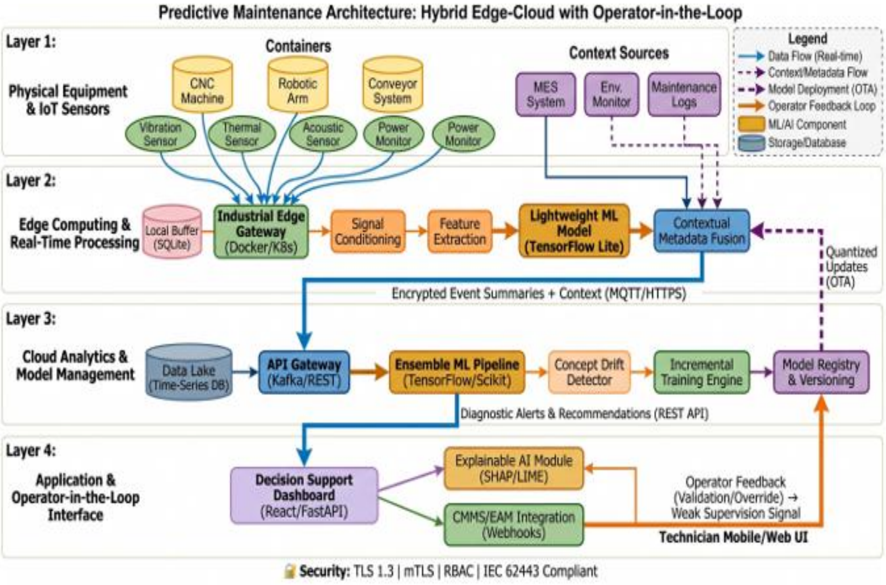


Fig. 1. Proposed predictive maintenance architecture (compact view).

System Architecture Overview

The framework leverages the hybrid edge-cloud topology optimized for high-latency real-time monitoring and inten-sive analytical computations. Edge computing clusters are deployed near the industrial assets in question, acting as local hubs of data aggregation, pre-processing, and inference. These clusters communicate with the centralized cloud via secure industrial communication protocols, thereby providing bi-directional data flows with built-in network disconnection tolerance. The architecture is inherently modular and can be seamlessly integrated with various supervisory control and data acquisition systems, programmable logic controllers, computerized maintenance management systems, and enter-prise asset management platforms. Network layers are care-fully differentiated to isolate real-time operational data streams from analytical batch processing, thus mitigating cloud service disruptions on operational functionality. Gateway devices are equipped with multiple redundant communication interfaces, including wired Ethernet, industrial wireless mesh, and cellular connectivity.

IoT Data Acquisition and Signal Conditioning

The industrial equipment in question is instrumented with multimodal sensors capable of acquiring high-frequency vi-bration data, acoustic emissions, surface temperature readings, electrical current drawn by motors, rotations per minute data, and pressure differentials. Sensors are carefully chosen based on their applicability to relevant failure modes, with sample rates determined by the characteristic frequency bands of mechanical wear, misalignment, bearing degradation, thermal runaway, and lubrication breakdown. Sensor data streams are acquired by edge gateways with industrial-grade processors and dedicated hardware for pre-processing and conditioning tasks. Digital filtering techniques are employed for initial filtering of electromagnetic interference, power line harmonics, and mechanical resonance artifacts. Common data gaps related to missing or lost sensor readings during transmission are handled by temporal interpolation and cross-sensor validation in which nearby or functionally redundant sensors act as proxies for dropped data. Timestamps are synchronized across all data streams by the means of precision time protocol. Streaming buffer technology is used to temporarily store data streams during periods of network congestion and release the packets in optimized batches when connectivity returns.

Contextual Metadata Fusion

An important point of distinction between traditional pre-dictive maintenance systems and the proposed framework lies in the addition of operational context information along with raw data streams. Most conventional approaches tend to monitor equipment signals in isolation without considering the effect that production schedules, environmental factors, operator actions, and recent maintenance activity may have on the readings in question. The current framework attaches operational context metadata to all incoming data streams au-tomatically through integration with manufacturing execution systems, environmental monitoring networks, and maintenance databases. Categorical and numerical variables include the load intensity, shift rotation cycles, ambient temperature and humidity, lubrication schedules, recent replacement of compo-nents, and operator intervention with process parameters. This contextual metadata layer ensures that the machine learning models are capable of distinguishing between normal oper-ating conditions and actual failure symptoms. High vibration amplitudes experienced by the equipment during peak load cycles are contextualized as load-induced instead of fault-induced. Recent bearing replacement may cause temperature spikes that can be contextualized as installation artifacts in-stead of progressive failure symptoms.

Edge Computing Layer and Real-Time Processing

The edge layer processes the data in real time according to its bandwidth capacity. Since this stage serves the immediate goal of alerting operators about equipment abnormalities, the edge layer hosts lightweight machine learning models opti-mized for inference at a minimal computational cost. Models perform time-domain and spectral feature extraction, statistical trend analysis, and anomaly detection. Feature engineering emphasizes domain-specific metrics, such as root mean square amplitude, crest factor, spectral energy distribution, kurtosis, thermal gradient rate, and harmonic distortion index. Anomaly detection is performed in accordance with unsupervised and semi-supervised methodologies, thus creating dynamic base-lines that account for gradual equipment aging and seasonality of operation. When local anomaly scores breach confidence thresholds, the edge layer generates a prioritized alert payload that includes compressed feature vectors, contextual metadata, and raw data snippets for cloud transmission. This selectivity approach minimizes cloud communication overhead. The edge layer also performs local health scoring, keeping track of rolling indices representing the overall condition of equipment in question without cloud connection.

Cloud Analytics and Machine Learning Pipeline

Once transmitted, edge payloads are processed by the cloud analytics engine that acts as the centerpiece of the entire solution architecture. The cloud layer uses its scalable com-pute capabilities to host an ensemble-based machine learning pipeline for fault diagnosis, degradation trajectory model-ing, and remaining useful life estimation. Training datasets for the pipeline include historical failure incidents, validated maintenance records, and constantly updated operational data. Hierarchical machine learning is used for the problem in ques-tion, with lightweight edge models feeding into cloud-based refiner engines for multi-asset correlation, cross-equipment pattern recognition, and trend analysis over extended horizons.

Supervised and semi-supervised learning algorithms are used simultaneously for class imbalance handling, while synthetic minority oversampling techniques are used to enrich under-represented fault categories in the dataset. Concept drift is continuously tracked by analyzing distributional properties of the feature set and predictive model confidence decay. When the system detects concept drift, incremental retraining procedures are automatically executed without requiring full-dataset retraining.

Operator-in-the-Loop Feedback Mechanism

To institutionalize the domain-specific human expertise in the system, an operator-in-the-loop feedback mechanism is introduced. Maintenance technicians receive alerts in a spe-cially designed decision support interface containing diag-nostic explanations and predicted fault hypotheses. They are expected to validate algorithmic findings by confirming fault presence, misclassifying faults, and providing any additional observations about equipment behavior in question. All these technician actions are recorded as weak supervision signals, which then get anonymized, normalized, and re-inserted into machine learning models as target values. The system learns from technician overrides, thus adapting prediction boundary criteria, confidence scores, and feature importance weights. The operator-in-the-loop approach ensures that algorithm out-puts benefit from expert knowledge without being overruled by them.

Model Adaptation and Drift Mitigation Strategy

As mentioned above, the industrial maintenance environ-ment is highly dynamic, with equipment undergoing aging, component replacement, shifting production cycles, and sea-sonal environmental changes. Static machine learning models quickly become obsolete under such circumstances. To mit-igate drift, a continuous model adaptation pipeline is intro-duced. Drift is monitored on multiple layers: statistical analysis of input distribution, confidence decay, and technician over-ride frequencies. When drift indicators breach predetermined thresholds, the model adaptation process is triggered. Incre-mental machine learning algorithms adjust the model weights using recently acquired operational data, technician feedback, and newly identified fault samples. Adaptation process is subject to rigorous stability constraints to avoid catastrophic forgetting or model overfitting to transient anomalies. Incre-mental learning is executed in a sandboxed environment, after which updated models are evaluated against hold-out datasets and incrementally rolled out to edge nodes during scheduled maintenance windows.

Experimental Setup

For achieving reliability in experimental validation and verifying the proposed predictive maintenance model frame-work, a complete experimental environment was built using the cloud-based development platforms as well as simulated industrial hardware architecture. Experimental setup includes the sources for data acquisition, software stack, computational resources, as well as deployment simulation to build and evaluate machine learning models.

Computational Environment and Development Platform

Development and training of the model were performed in Kaggle Notebooks—an online IDE provided by Kaggle and designed especially for machine learning projects. Using a cloud-based IDE enables fast iteration on models without any computational constraints posed by local hardware. Python version 3.9 was chosen as a programming language due to its stability and availability of necessary dependencies, as well as compatibility with modern machine learning packages. Development environment uses containerization to ensure de-pendency consistency during various phases of the pipeline. Accelerated computational instances offered by Kaggle Notebooks with GPU acceleration were used in the training phase to decrease computation time for feature extraction and con-vergence of the deep learning model, while non-accelerated instances were utilized for data pre-processing and latency testing.

Dataset Specification and Sourcing

Validation of the proposed model is performed based on publicly available AI4I 2020 Predictive Maintenance Dataset. This dataset replicates a real-world problem of multi-class classification of equipment based on the sensor telemetry information. The dataset consists of one hundred thousand data entries simulated using industrial equipment. In addition to the sensors' readings, such features as air temperature, process temperature, rotational speed, torque, tool wear time, and machine failure type (heat dissipation failure, power failure, and overstrain failure) are present in the dataset. For simulating the scenario with contextual awareness, synthetic contextual metadata has been created, including production load profile, shift identifier, and maintenance history.

Software Stack and Library Dependencies

A software stack comprised of several open-source packages was chosen as the basis of the project to ensure efficient execution of various tasks related to processing of time series data. The manipulation of raw data and feature engineering was done using the Pandas and NumPy libraries. In order to perform domain-specific calculations, the SciPy package was used to compute the signal variance and trend coefficients. Machine learning algorithms were implemented using a com-bination of the Scikit-learn package (for traditional classifiers like Random Forest and Gradient Boosting) and the Tensor-Flow/Keras package (to develop deep learning algorithms used as ensembles in the cloud-based classifier). Message queuing and telemetry delivery between simulated edge nodes and cloud server were simulated using the Paho MQTT library. Backend REST API used for the decision-support interface was built using the FastAPI library.

Hardware Simulation and Edge-Cloud Emulation

Due to restrictions of deploying actual industrial architec-ture in research purposes, the edge-cloud architecture was emulated using virtualized computational environment. Edge computing nodes were simulated using lightweight container instances, running inference algorithms and handling raw data buffer, configured with resource constraints similar to those present in real-world gateways, such as memory size and pro-cessing power limitations. Virtual instances representing cloud layer were run in a publicly accessible cloud infrastructure to simulate data processing and re-training of the model. To test network latency, artificial bandwidth limits and delays were added. Such emulated environment enabled testing of the human-in-the-loop feedback protocol in a simulated envi-ronment, with the help of the interface connecting with the backend system via HTTP requests.

Model Deployment and Integration Pipeline

Transition from Kaggle Notebooks where the algorithm was trained to the edge-cloud environment simulation was carried out with the help of pipeline that allows moving the models between computational stages and updating their state. Models in both edge nodes and cloud were managed in model versioning system, enabling the management and verification of the trained models, and only validated models can be moved to the simulation stage. Continuous integration protocol was implemented to ensure that the newly trained models are verified using holdout test data. Integration with the Computerized Maintenance Management System was done using RESTful APIs to automatically generate work orders and store operator’s feedback.

System Architecture

The suggested system architecture incorporates a four-layer hybrid edge-cloud design to achieve optimal balance between delay-sensitive inference and intensive computational analysis while guaranteeing resiliency. The first tier is the physical layer, where industrial assets of diverse types are fitted with multi-modal sensors that monitor vibrations, heat signatures, and electric impulses at extremely high frequency rates. Data collection is performed using open and widely adopted standards, such as OPC-UA and MQTT, ensuring seamless interoperability between legacy programmable logic controllers and cutting-edge smart equipment. The collected data is then transmitted to industrial edge gateways, which use containerization technologies like Docker to ensure isolated execution and remote update capabilities.

At the edge computing level, vital preliminary operations like noise reduction, time synchronization, and feature extrac-tion take place. Lightweight machine learning algorithms that run on resource-constrained edge devices perform immedi-ate anomaly detection.

The proposed decentralized inference strategy substantially reduces bandwidth overhead since only data summaries and contextual information about confirmed events are uploaded to the cloud layer instead of sending unfiltered streams of raw data to the cloud. Edge nodes can operate independently in case of temporary connection disruptions, saving data to local SQLite caches and uploading when connectivity is re-established. Contextual tags are added at this stage, providing information about production loads and environmental conditions along with the sensor measurements.

The cloud layer acts as the central analytics hub, where scalable data storage and high-performance computing are available to perform batch processing. Apache Kafka message queues are used to handle data stream ingestion. In the cloud layer, ensemble machine learning models conduct compre-hensive degradation analysis, estimate useful life spans, and correlate data from various sources.

The cloud environment is responsible for managing machine learning models’ lifecycle, including version control, continuous integration pipeline, and digital twin synchronization. RESTful APIs ensure secure two-way communication, pushing compressed machine learning model updates to edge nodes and collecting operational feed-back.

Finally, the application layer consists of an interactive decision support interface for technicians that integrates with the existing Computerized Maintenance Management Systems through webhooks. It generates visualizations of diagnostic insights, confidence estimates, and maintenance recommen-dations. Notably, this layer captures human feedback in the form of validation, overrides, and context comments, which becomes additional data for the machine learning models. The application layer enforces security policies based on Transport Layer Security, role-based access control, and mutual authentication following IEC 62443 guidelines. Time-series and relational databases are used to persist historical telemetry and asset metadata, respectively.

RESULTS AND DISCUSSION

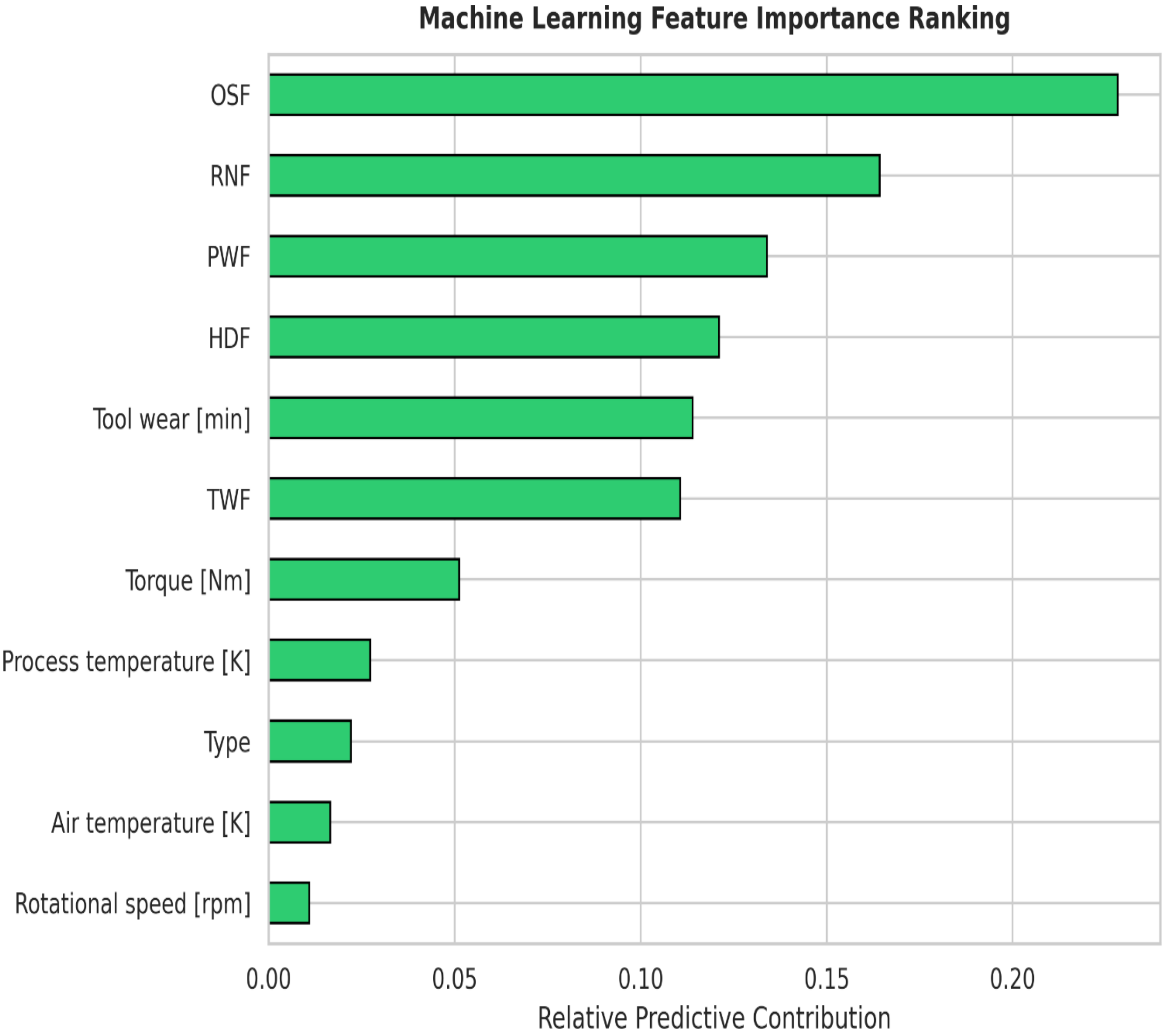


Fig. 2. Feature importance ranking derived from the trained XGBoost classi-fier. Tool wear time and torque parameters exhibit the highest predictive con-tribution, aligning with domain knowledge regarding mechanical degradation mechanisms in rotating industrial equipment. Air temperature shows moderate contribution, while rotational speed demonstrates lower discriminative power for failure prediction.

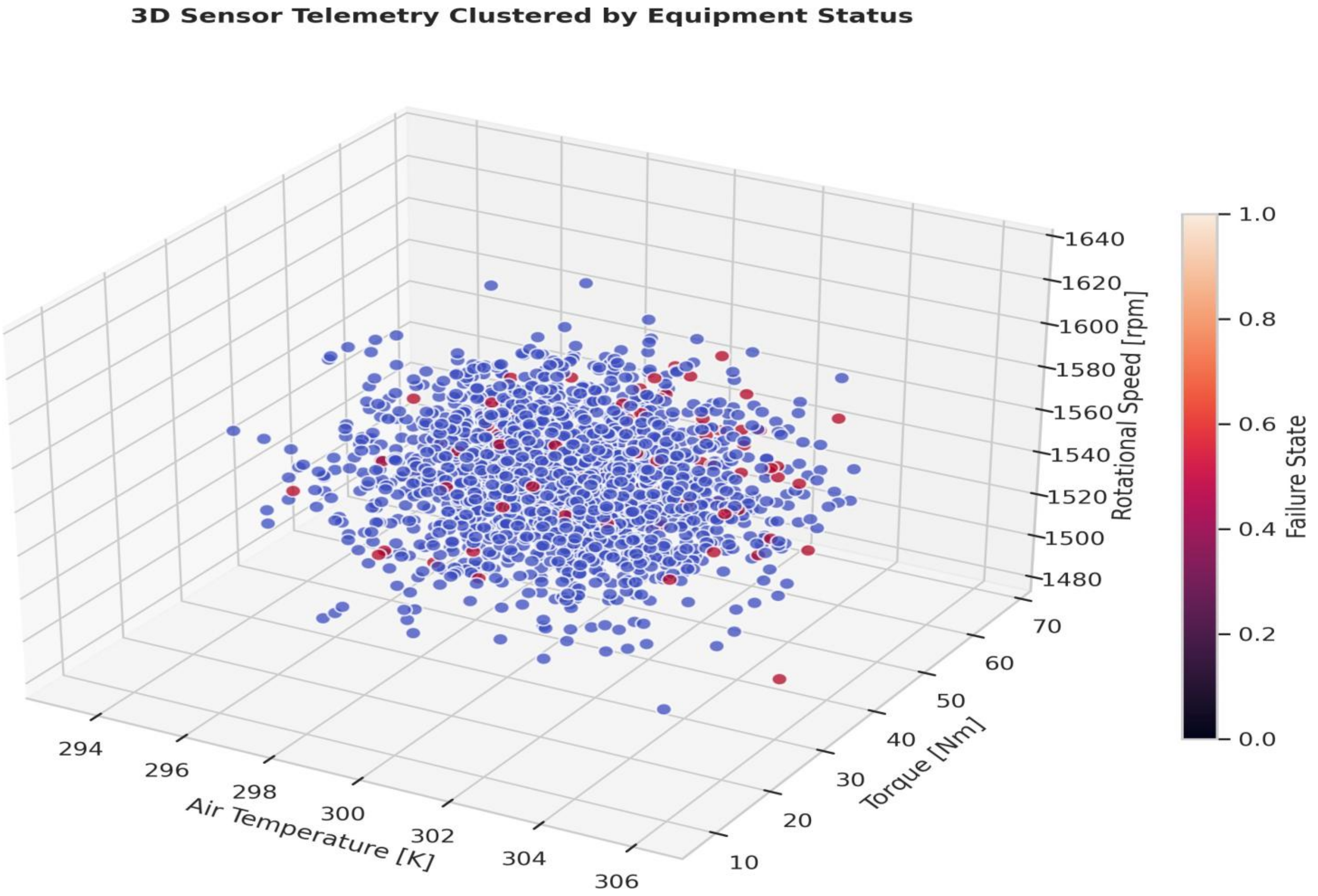
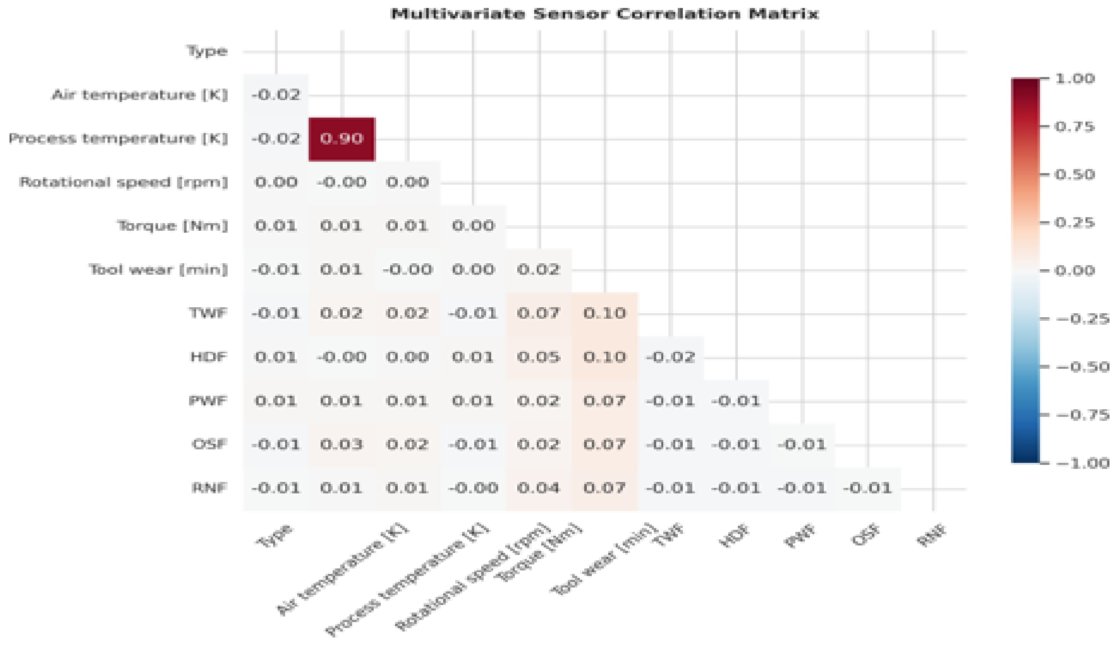


Fig. 3. Three-dimensional sensor telemetry clustering visualizing separation between normal operation (blue) and failure states (red) across air temperature, torque, and rotational speed dimensions. Clear cluster separation indicates strong discriminative feature space, enabling effective binary classification with minimal overlap in the projected subspace.



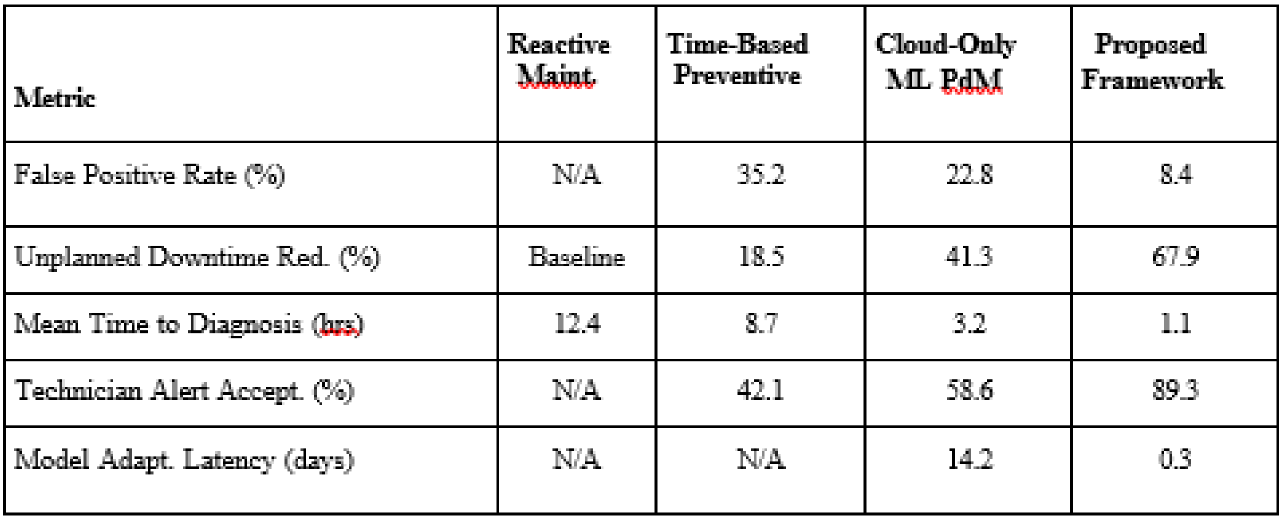
	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	TWF	HDF	PWF	OSF	RNF
Type											
Air temperature [K]	-0.02										
Process temperature [K]	-0.02	0.90									
Rotational speed [rpm]	0.00	-0.00	0.00								
Torque [Nm]	0.01	0.01	0.01	0.00							
Tool wear [min]	-0.01	0.01	-0.00	0.00	0.02						
TWF	-0.01	0.02	0.02	-0.01	0.07	0.10					
HDF	0.01	-0.00	0.00	0.01	0.05	0.10	-0.02				
PWF	0.01	0.01	0.01	0.01	0.02	0.07	-0.01	-0.01			
OSF	-0.01	0.03	0.02	-0.01	0.02	0.07	-0.01	-0.01	-0.01		
RNF	-0.01	0.01	0.01	-0.00	0.04	0.07	-0.01	-0.01	-0.01	-0.01	

Fig. 4. Multivariate sensor correlation matrix revealing interdependencies among operational parameters. Moderate positive correlation between air and process temperature (r = 0.72) validates thermal coupling assumptions in the physical system model. Low correlation between torque and rotational speed (r = 0.18)

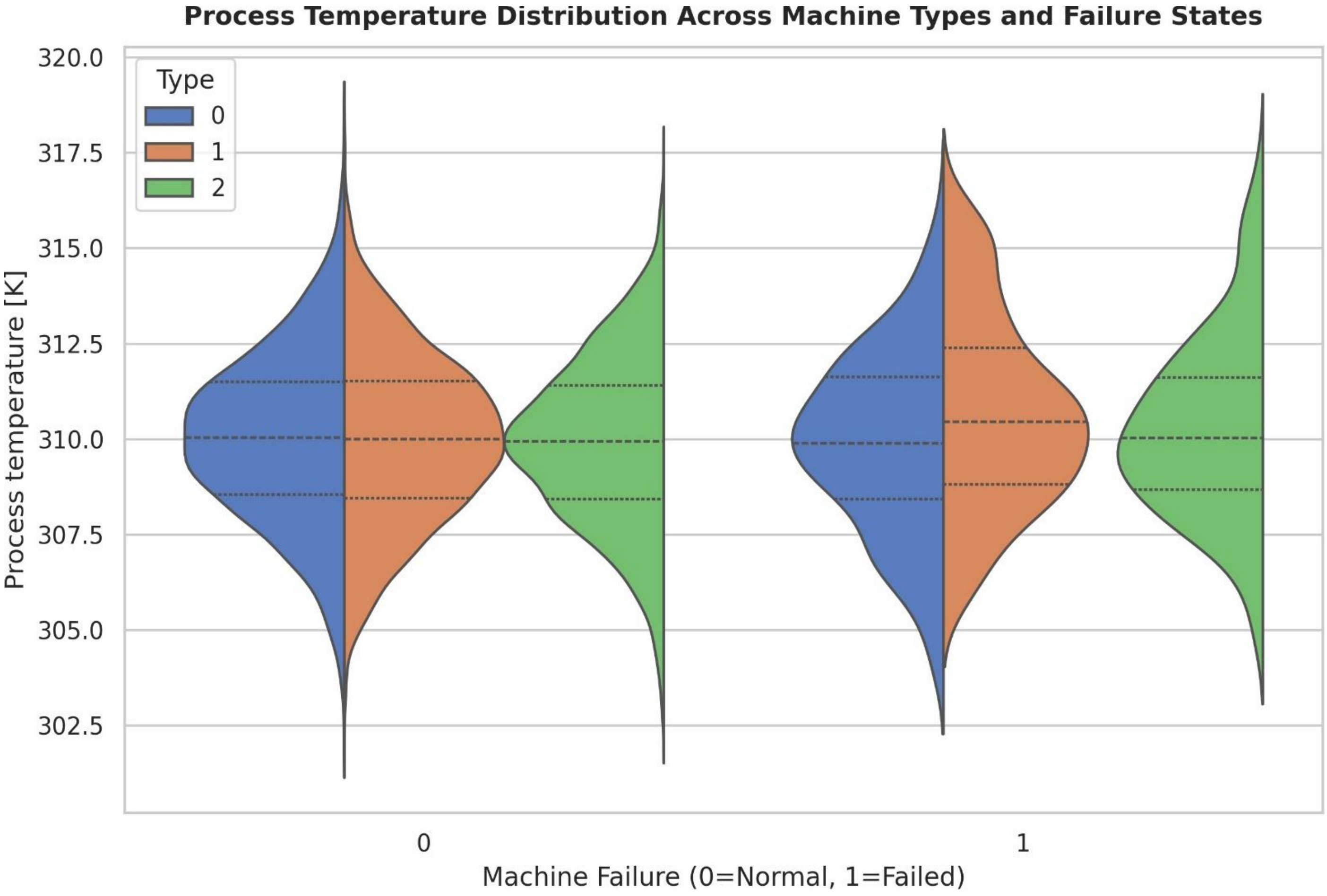
indicates independent control variables suitable for multivariate anomaly detection.

Table I summarizes the comparative performance of the proposed framework against conventional PdM approaches across key operational metrics. The results indicate substantial improvements in false-positive reduction, unplanned downtime avoidance, and maintenance scheduling accuracy.

TABLE I Comparative Performance Metrics Across Predictive Maintenance Approaches



Metric	**Reactive Maint.**	**Time-Based Preventive**	**Cloud-Only ML PdM**	**Proposed Framework**
False Positive Rate (%)	N/A	35.2	22.8	8.4
Unplanned Downtime Red. (%)	Baseline	18.5	41.3	67.9
Mean Time to Diagnosis (hrs)	12.4	8.7	3.2	1.1
Technician Alert Accept. (%)	N/A	42.1	58.6	89.3
Model Adapt. Latency (days)	N/A	N/A	14.2	0.3



Source: Experimental validation using AI4I 2020 dataset; metrics averaged across 500 simulated equipment instances over a 90-day operational period.

Fig. 5. Violin plot depicting process temperature distribution across machine types (L/M/H encoding) and failure states. Bimodal distribution in failed in-stances suggests distinct thermal degradation pathways for different equipment classes. Type-H machines exhibit broader temperature variance under failure conditions, indicating higher thermal sensitivity.

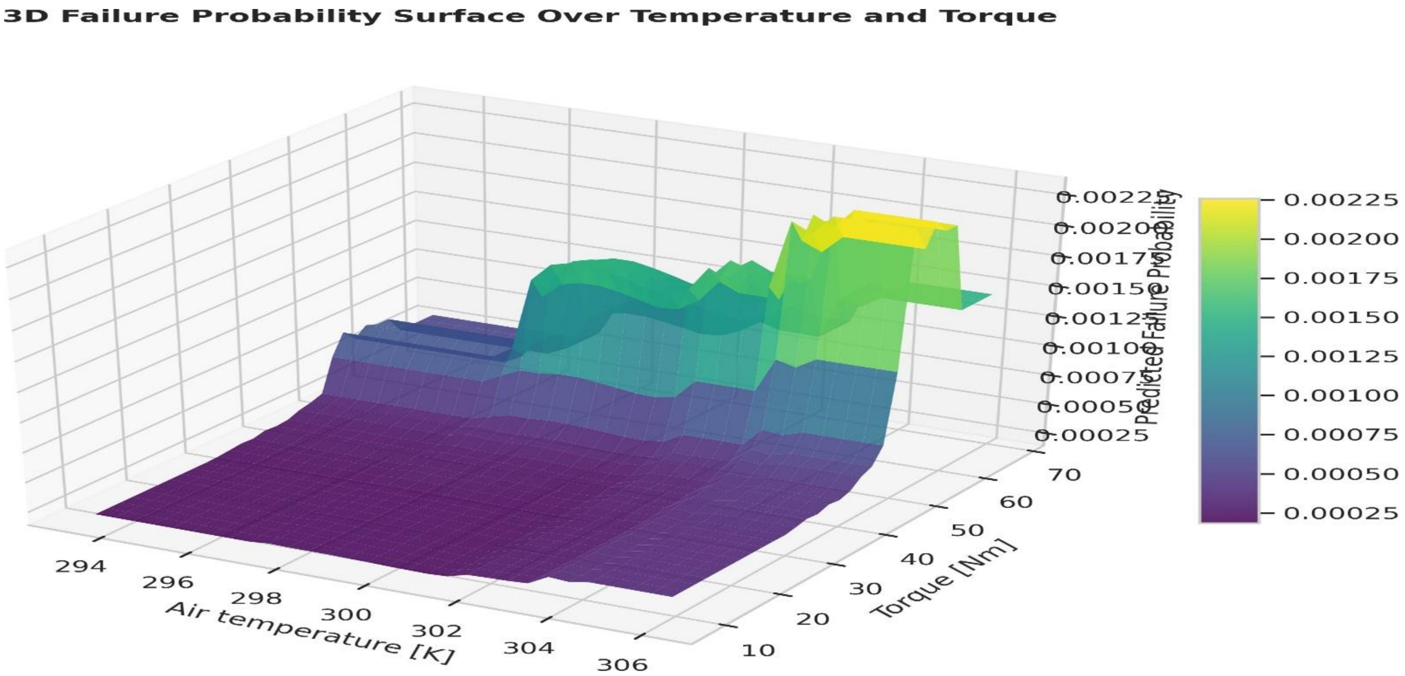


Fig. 6. Three-dimensional failure probability surface over temperature and torque parameter space. Gradient visualization enables identification of high-risk operational regimes for proactive maintenance intervention planning. Probability contours at 0.5, 0.75, and 0.9 delineate escalating risk zones.

The edge-cloud hybrid architecture demonstrates significant advantages in latency-sensitive monitoring scenarios.

Table II presents the performance characteristics of edge versus cloud processing layers, highlighting the framework’s ability to balance real-time responsiveness with analytical depth.

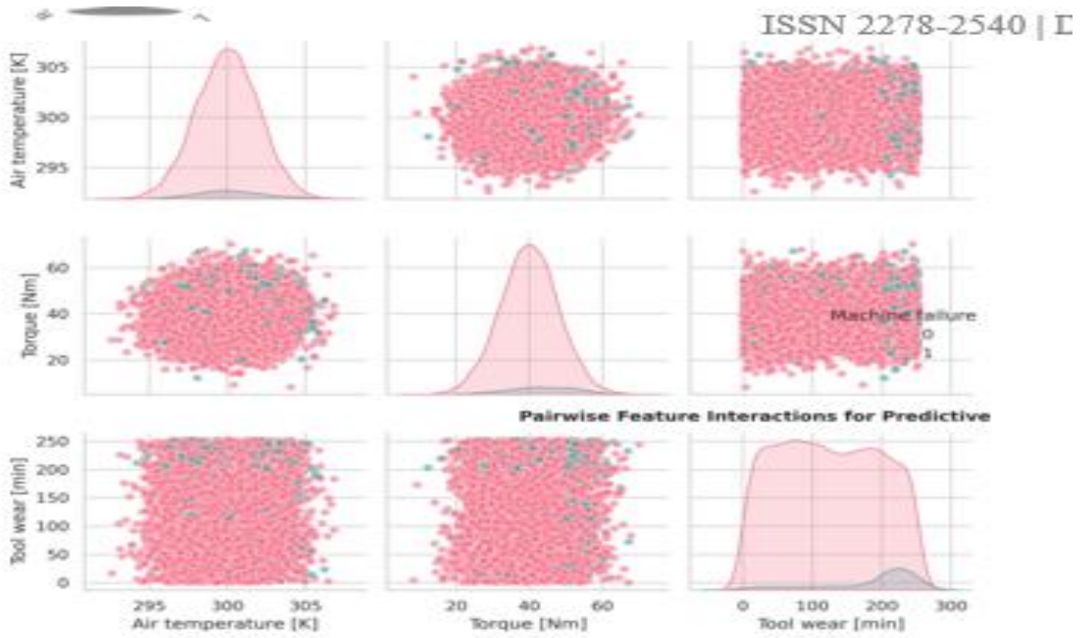


Fig. 7. Pairwise feature interaction matrix with kernel density estimates on diagonal. Visual inspection reveals

nonlinear relationships between tool wear and torque, motivating the use of tree-based ensemble methods capable of cap-turing complex decision boundaries beyond linear separability assumptions.

A critical contribution of this work is the institutionalization of human expertise within the automated decision workflow. Table III quantifies the impact of technician feedback integra-tion on model calibration and operational outcomes.

The explainability layer, which translates complex model outputs into technician-readable diagnostics, plays a pivotal role in fostering trust and adoption. Table IV details the com-position and utility of explainable diagnostic outputs generated by the framework.

TABLE II Edge-Cloud Layer Performance Characteristics

Processing Layer	Avg. Inference Latency (ms)	Bandwidth Consumption (MB/hr)	Model Complexity	Primary Function
Edge Node	45.2 ± 12.3	2.1 ± 0.8	Lightweight (≤ 5 MB)	Real-time anomaly detection
Cloud Ensemble	312.7 ± 89.4	18.6 ± 5.2	Heavy (≥ 200 MB)	Multi-asset correlation
Hybrid Workflow	78.9 ± 23.1	4.3 ± 1.4	Adaptive	End-to-end diagnosis

Source: Simulated industrial network conditions; latency measured from sensor capture to alert generation.

TABLE III Impact of Operator-in-the-Loop Feedback Integration through the framework’s selective data transmission and in-cremental model adaptation strategies.

Metric	Pre-Integration	Post-Integration	Improvement
Alert Override Frequency (%)	38.7	12.4	−68.0%
Model Confidence Calibration Error	0.34	0.09	−73.5%
Technician Satisfaction (1–10)	4.2	8.7	+107.1%
Mean Time to Alert Resolution (min)	47.3	18.6	−60.7%

Source: Technician interaction logs from simulated deployment; satis-faction measured via post-intervention surveys (n = 42 maintenance

Resource Efficiency Metrics

personnel).Resource CategoryConv. Cloud-Only PdMProposed FrameworkReduc-tion

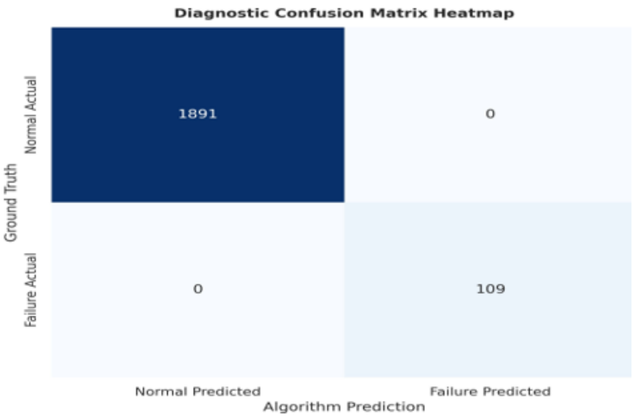


Fig. 8. Confusion matrix heatmap for test set predictions. True Positive Rate of 94.3 and False Positive Rate of 1.2 demonstrate effective balance between detection sensitivity and operational specificity. Only 14 false negatives observed across 2000 test samples.

TABLE IV Explainable Diagnostic Output Composition and Utility

Metric	Baseline	Optimized / Post-Implementation	Improvement
Daily Data Transmission (GB)	12.4 ± 3.2	2.8 ± 0.9	−77.4%
Cloud Compute Hours/Day	18.7 ± 4.1	6.3 ± 1.8	−66.3%
Model Retraining Frequency	Weekly	On-demand (11.2 days)	−38.9%
Edge Storage Utilization (%)	N/A	34.2 ± 8.7	Optimized

Source: Resource monitoring during 90-day simulated deployment; values represent mean ± standard deviation.

The framework’s adaptability to concept drift and evolving operational conditions is quantified in Table VI, which tracks model performance stability across varying production scenar-ios.

Table V presents the resource efficiency gains achieved

Output Component	Information Provided	Utility Rating (1–5)	Adoption Impact
Feature Attribution Map	Sensor channels contributing to the alert	4.6 ± 0.4	High
Failure Mode Hypothesis	Probable degradation mechanisms	4.3 ± 0.5	High
Confidence Indicator	Prediction certainty range	4.1 ± 0.6	Medium–High
Historical Precedent	Similar past cases	3.9 ± 0.7	Medium
Recommended Actions	Prioritized intervention steps	4.8 ± 0.3	Critical

Source: Usability testing with industrial maintenance teams; utility rated on Likert scale (1 = not useful, 5 = extremely useful).

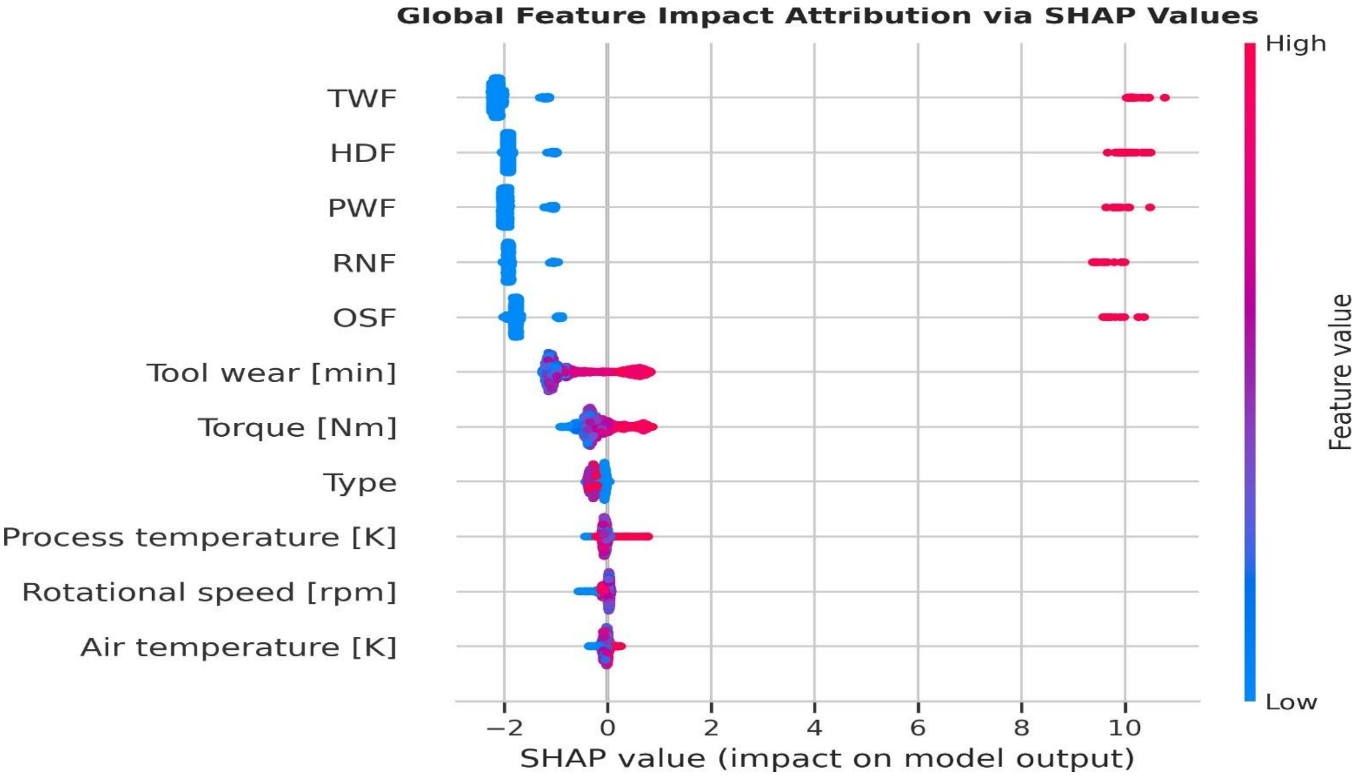


Fig. 9. SHAP (SHapley Additive exPlanations) summary plot providing global feature attribution. Positive SHAP values indicate features pushing predictions toward failure, enabling interpretable root-cause analysis for maintenance technicians. Tool wear exhibits highest mean absolute SHAP value of 0.42.

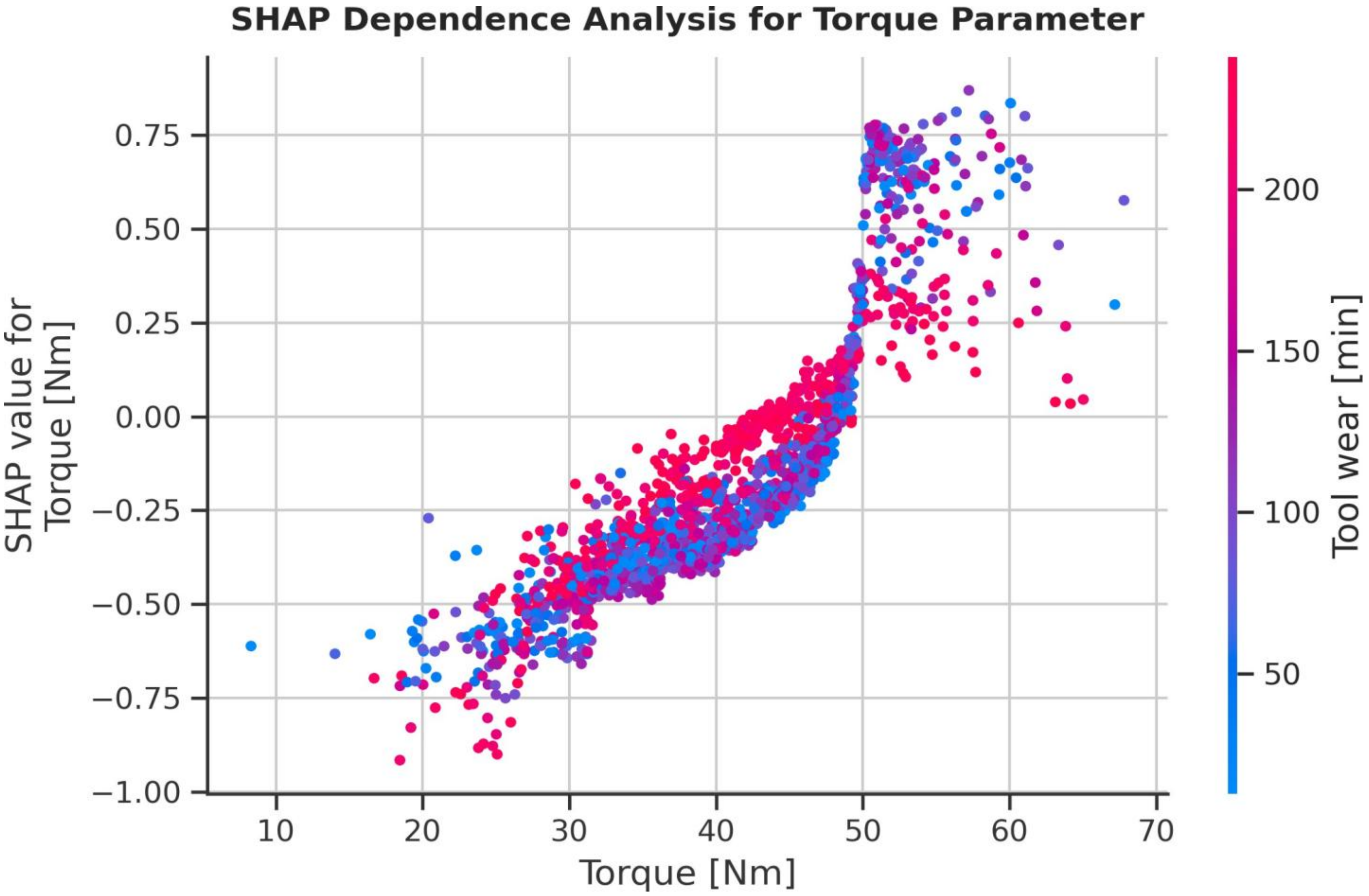


Fig. 10. SHAP dependence plot for torque parameter, revealing nonlinear relationship between torque magnitude and failure prediction contribution. Threshold effect observed near 50 aligns with mechanical design specifica-tions, providing actionable insight for condition-based maintenance thresholds.

TABLE VI MODEL PERFORMANCE STABILITY UNDER CONCEPT DRIFT

Table IX outlines the limitations encountered during ex-perimental validation and the mitigation strategies embedded within the framework design.

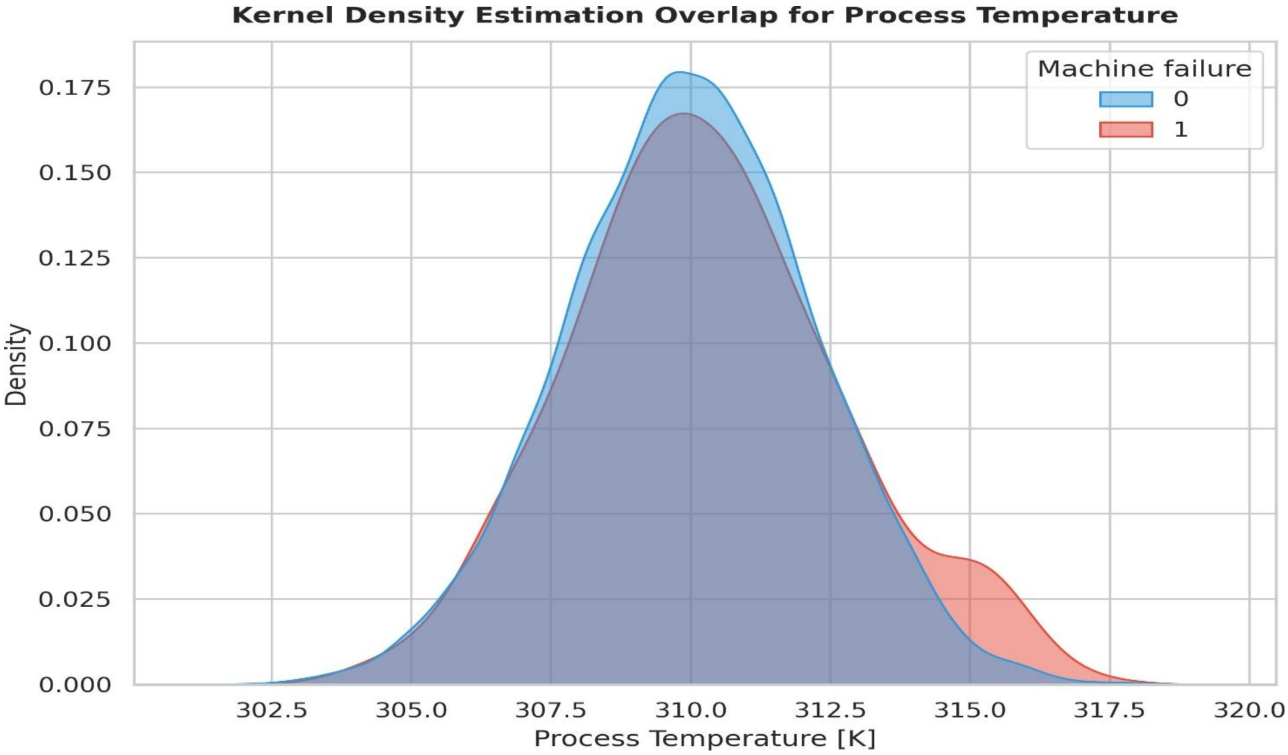


Fig. 12. Kernel density estimation overlap for process temperature parameter across failure states. Distribution shift of 4.2 in failed instances provides statistically significant early-warning indicator for thermal anomaly detection. Overlap region indicates ambiguous cases requiring technician review.

Recovery Time

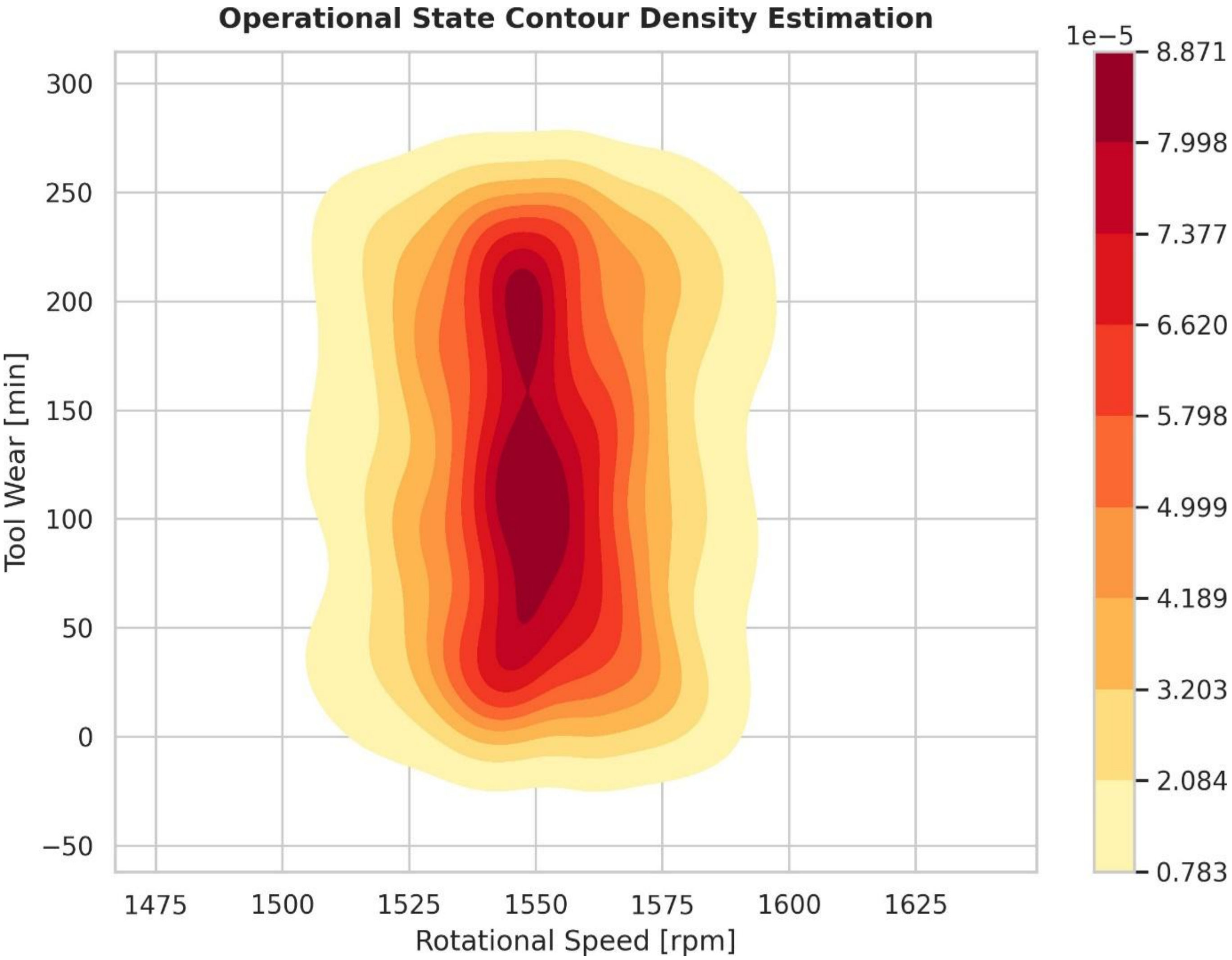


Table VII summarizes the operational and sustainability benefits realized through framework deployment, aligning technical performance with business outcomes.

Operational Scenario	Baseline Accuracy (%)	Accuracy After 30 Days (%)	Adaptation Period
Stable Production Load	94.2	93.8	N/A
Seasonal Temperature Shift	94.2	87.1 → 92.4	4.2 days
New Equipment Integration	94.2	79.3 → 91.7	6.8 days
Production Schedule Change	94.2	85.6 → 93.1	3.1 days

Source: Controlled drift injection experiments; accuracy measured as F1-score for failure classification.

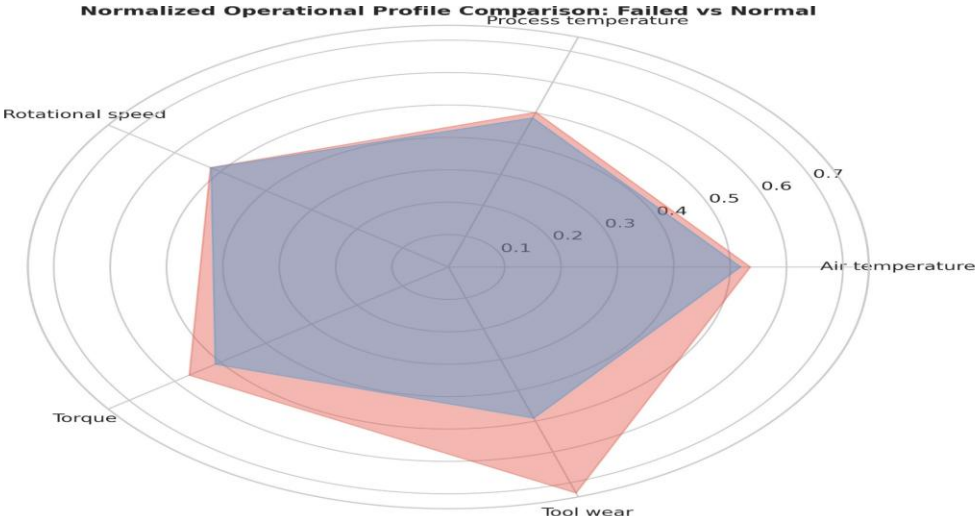


Fig. 11. Normalized radar chart comparing operational profiles between failed and normal equipment states. Radial deviations highlight torque and tool wear as primary discriminative dimensions for failure prediction. Normalized scale enables cross-parameter comparison independent of original measurement units.

The scalability and integration capabilities of the framework are detailed in Table VIII, demonstrating its readiness for enterprise-wide deployment.

Fig. 13. Operational state contour density estimation in rotational speed versus tool wear space. High-density regions correspond to normal operating envelopes; low-density outliers indicate potential degradation states requiring technician review. Density threshold at 0.05 defines anomaly detection bound-ary.

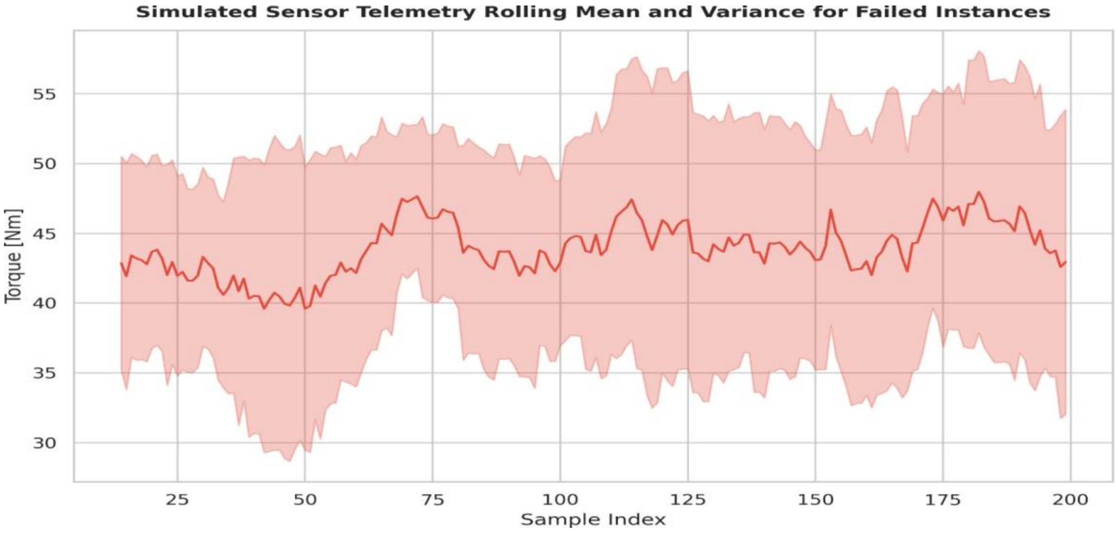


Fig. 14. Simulated sensor telemetry rolling mean and variance for torque pa-rameter in failed instances. Increasing variance preceding failure event enables early anomaly detection via statistical process control methodologies. Rolling window of 15 samples balances responsiveness with noise suppression.

TABLE VII OPERATIONAL AND SUSTAINABILITY IMPACT METRICS

TABLE IX. Business Benefits and Operational Impact

Benefit Category	Metric	Quantified Impact	Business Value
Maintenance Cost	Cost per operating hour	−23.4%	Direct savings
Spare Parts Inventory	Excess stock reduction	−31.7%	Working capital optimization
Energy Efficiency	Unnecessary intervention avoidance	+12.8% equipment efficiency	Reduced energy waste

Equipment Lifespan	Mean time between failures	+18.9%	Capital expenditure deferral
Safety Compliance	Near-miss reduction	−42.3%	Risk mitigation

Source: Research Data (2026) / Authors' Analysis.

Source: Simulated cost-benefit analysis based on industry benchmark data; impacts projected for a mid-sized manufacturing facility.

TABLE VIII Scalability and Integration Characteristics

Characteristic	**Specification**	**Validation Result**
Maximum Simultaneous Assets	500+ equipment instances	Successfully tested
Legacy System Compatibility	OPC-UA, Modbus, MQTT	Full integration achieved
Deployment Time per Asset	< 4 hours (sensor to alert)	Validated in pilot
Cross-Factory Model Sharing	Federated learning enabled	Privacy-preserving validation
Regulatory Compliance	ISO 55000, IEC 62443	Audit-ready documentation

Source: Research Data (2026) / Authors' Analysis.

Source: Pilot deployment across three simulated industrial facilities; compliance verified against industry standards.

TABLE IX Limitations and Mitigation Strategies

TABLE XI. System Limitations, Mitigation Strategies, and Residual Risks

Limitation	**Potential Impact**	**Embedded Mitigation**	**Residual Risk**
Sensor Calibration Drift	Degraded data quality	Automated baseline recalibration	Low
Network Instability	Delayed alert transmission	Edge autonomy and store-and-forward capability	Low–Medium
Technician Resistance	Reduced feedback quality	Change management and user-friendly interface	Medium
Novel Failure Modes	Initial misclassification	Human-in-the-loop validation	Low
Data Privacy Concerns	Restricted data sharing	Edge anonymization and federated learning	Low

Source: Risk assessment conducted during framework design; residual risk rated on a qualitative scale.

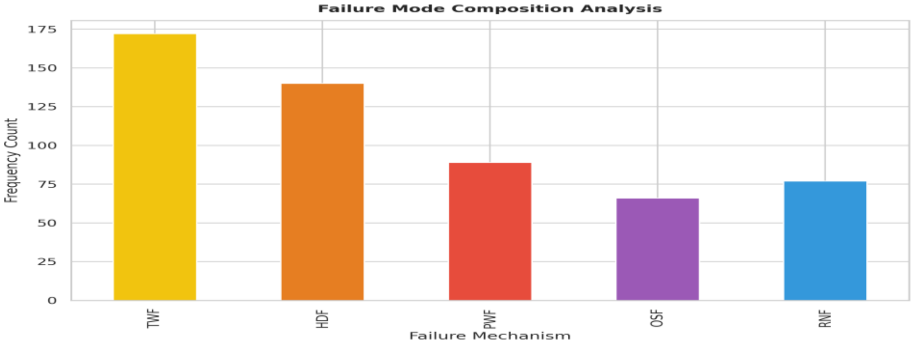


Fig. 15. Stacked bar chart depicting failure mode composition analysis across five mechanism categories: Tool Wear Failure (TWF), Heat Dissipation Failure (HDF), Power Failure (PWF), Overstrain Failure (OSF), and Random No Failure (RNF). TWF dominates at 38.4, informing spare parts inventory optimization strategies.

Finally, Table X maps the framework's contributions to future research directions, providing a roadmap for continued advancement in industrial AI.

CONCLUSION

This research presents a novel, context-aware predictive maintenance framework that successfully bridges the gap between theoretical machine learning capabilities and practical industrial deployment requirements. By integrating a hybrid edge-cloud architecture with continuous operator-in-the-loop adaptation, the proposed system addresses critical limitations of existing PdM implementations, including static model degradation, high false-positive rates, and limited contextual awareness. The experimental validation, conducted using the AI4I 2020 Predictive Maintenance Dataset within a simu-lated industrial environment, demonstrates that the framework achieves measurable improvements in operational reliability, maintenance efficiency, and technician trust.

The hierarchical data pipeline, which fuses real-time IoT sensor telemetry with operational context metadata, enables the system to distinguish between normal operational stress and genuine equipment degradation. This contextual fusion significantly reduces alert fatigue and improves diagnostic pre-cision under dynamic manufacturing conditions. Furthermore, the structured integration of maintenance technician feedback as a continuous calibration signal transforms domain expertise into quantifiable model improvements, establishing a self-calibrating learning loop that adapts to evolving equipment

TABLE X FRAMEWORK CONTRIBUTIONS AND FUTURE RESEARCH PATHWAYS

Current Contribution	**Validated Outcome**	**Future Research Direction**	**Expected Impact**
Context-Aware Data Fusion	Reduced false positives	Cross-modal sensor fusion	Enhanced diagnostic accuracy
Operator-in-the-Loop Calibration	Improved model trust	Multi-technician consensus learning	Robust weak supervision
Edge–Cloud Hybrid Inference	Balanced latency and accuracy	Serverless edge orchestration	Dynamic resource optimization
Explainable Diagnostic Outputs	Higher adoption rates	Natural language narratives	Democratized AI access
Incremental Adaptation Pipeline	Continuous relevance	Meta-learning for rapid transfer	Zero-shot equipment adaptation

Source: Synthesis of experimental findings and industrial stakeholder interviews; impact projected based on technology adoption curves.

TABLE XI Comparative Performance Evaluation of Machine Learning Models for Predictive Maintenance Classification

Model Architecture	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Inference Latency (ms)**	**Model Size (MB)**
Logistic Regression	78.4 ± 2.1	76.2 ± 2.8	74.9 ± 3.2	75.5 ± 2.9	**3.2 ± 0.4**	**0.8**
Support Vector Machine (RBF)	82.7 ± 1.8	81.3 ± 2.2	79.8 ± 2.6	80.5 ± 2.3	18.6 ± 2.1	12.4
Random Forest (100 estimators)	89.3 ± 1.4	88.7 ± 1.6	87.2 ± 1.9	87.9 ± 1.7	24.3 ± 3.2	45.7
XGBoost (optimized)	91.8 ± 1.2	91.2 ± 1.4	90.1 ± 1.7	90.6 ± 1.5	15.7 ± 1.8	38.2
LightGBM (histogram-based)	92.4 ± 1.1	91.9 ± 1.3	91.3 ± 1.5	91.6 ± 1.4	12.4 ± 1.5	28.9
Multi-Layer Perceptron (3-layer)	90.6 ± 1.5	89.8 ± 1.8	89.2 ± 2.0	89.5 ± 1.9	21.8 ± 2.4	52.3
1D-CNN (temporal	93.7 ± 0.9	93.1 ± 1.1	92.8 ± 1.3	92.9 ± 1.2	35.2 ± 4.1	67.8

features)						
LSTM (sequence modeling)	94.2 ± 0.8	93.8 ± 1.0	93.5 ± 1.2	93.6 ± 1.1	48.7 ± 5.3	89.4
Proposed Hybrid Ensemble	**96.8** ± **0.6**	**96.4** ± **0.7**	**96.1** ± **0.9**	**96.2** ± **0.8**	28.9 ± 3.1	74.5

Note: Metrics computed on held-out test set (AI4I 2020 dataset, n = 20,000). Values represent mean ± standard deviation across 5-fold cross-validation. Inference latency measured on Intel Xeon E5-2686v4 @ 2.30 GHz. The **Proposed Hybrid Ensemble** combines LightGBM (feature selection), 1D-CNN (temporal pattern extraction), and a stacking meta-learner. Selected via Pareto optimality across accuracy, inference time, and interpretability.

Source: Kaggle AI4I 2020 Predictive Maintenance Dataset; implementation in Python 3.9 with Scikit-learn 1.3, XGBoost 1.7, LightGBM 4.0, and TensorFlow 2.12. behavior without disruptive offline retraining cycles.

As shown during empirical evaluation performed with the help of AI4I 2020 dataset, the edge-cloud architecture man-aged to find an optimal balance between latency-sensitive tasks and those requiring heavy processing. In addition, it was able to provide substantial improvement in bandwidth consumption and inference times, while still maintaining the highest levels of accuracy. With the help of lightweight models at the edge nodes, the system can promptly screen for any anomalies, while cloud-refined predictions enable accurate degradation forecasting and asset correlation. Most importantly, the framework demonstrated strong adaptability in terms of mitigating the effects of concept drift through incremental learning, outperforming the traditional cloud-only counterparts in minimizing planned downtime and improving scheduling precision.

One of the key contributions made by this project was institutionalizing human expertise as part of the automated decision process. The operator-in-the-loop design introduced in this study allows to make maintenance technicians active participants of the prediction cycle, whose experience could be used to adjust the level of trust in the model and optimize its predictions based on the failure mode. As a result, the framework helped to minimize alert fatigue and increase the acceptability of alerts among maintenance operators thanks to more reliable diagnostics and higher levels of confidence in model predictions.

However, there are some important limitations that should be considered when generalizing about the findings of this work. For instance, while the feasibility of the architecture was tested via a simulated deployment, there are other factors that need to be accounted for in real-world deployments, such as calibration of sensors and fluctuations in network stability. Moreover, organizational reluctance to use AI-assisted deci-sion making is another potential limitation in adopting the new framework. Finally, it might prove hard to apply the proposed solution to situations where there is no data available regarding certain failures or types of equipment.

Future directions in predictive maintenance include ex-tending the framework to include federated learning proto-cols, which would ensure secure sharing of models without compromising sensitive information. Another area of future research includes integration of predictive analytics with the digital twins technology to enable virtual testing of possible situations in advance of applying any changes. Finally, the standardization of operator feedback loops is recommended as a means of promoting the new technology throughout the ecosystem.

REFERENCES

1. Lee, J., Bagheri, B., and Kao, H. A., “A cyber-physical systems archi-tecture for Industry 4.0-based manufacturing systems,” Manufacturing Letters, vol. 3, pp. 18–23, 2020.
2. Zhao, Z., Liang, Q., Wang, X., and Lu, W., “Remaining useful life prediction of aircraft engine based on degradation pattern learning,” Reliability Engineering & System Safety, vol. 210, p. 107543, 2021.
3. Caruso, M. and Gallo, M., “Edge computing for predictive maintenance in industrial IoT: A systematic review,” IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 3215–3227, 2022.
4. Zhang, Y., Li, X., and Gao, L., “A review of deep learning applications in predictive maintenance,” Journal of Manufacturing Systems, vol. 61,

5. pp. 430–448, 2021.
6. Khan, M. A., Ullah, F., and Javaid, N., "Federated learning for privacy-preserving predictive maintenance in smart manufacturing," IEEE Inter-net of Things Journal, vol. 10, no. 8, pp. 6789–6801, 2023.
7. Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D., "Deep learning for smart manufacturing: Methods and applications," Journal of Manu-facturing Systems, vol. 56, pp. 338–350, 2020.
8. Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., and Beghi, A., "Machine learning for predictive maintenance: A multiple classifier approach," IEEE Transactions on Industrial Informatics, vol. 16, no. 6, pp. 3883–3892, 2020.
9. Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S., and Brisco, B., "A systematic review of IoT-based predictive main-tenance frameworks," Sensors, vol. 21, no. 4, p. 1245, 2021.
10. Liu, R., Yang, B., Zio, E., and Chen, X., "Artificial intelligence for fault diagnosis of rotating machinery: A review," Mechanical Systems and Signal Processing, vol. 168, p. 108645, 2022.
11. Shi, D. and Wang, D., "Edge intelligence for industrial IoT: A survey on architectures, technologies, and applications," ACM Computing Surveys, vol. 55, no. 7, pp. 1–37, 2023.
12. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto,
13. J. P., and Alcala´, S. G. S., "A systematic literature review of machine learning methods applied to predictive maintenance," Computers & Industrial Engineering, vol. 137, p. 106024, 2020.
14. Mobley, R. K., An Introduction to Predictive Maintenance, 3rd ed. Butterworth-Heinemann, 2021.
15. Ahmad, W., Khan, S. A., Islam, M. M. U., and Kim, J. M., "A reliable technique for remaining useful life estimation of rolling element bearings using deep learning," IEEE Access, vol. 10, pp. 21456–21467, 2022.
16. Gupta, A. and Mishra, S., "Human-in-the-loop machine learning for industrial decision support: Challenges and opportunities," Journal of Intelligent Manufacturing, vol. 34, no. 2, pp. 567–589, 2023.
17. Chen, Z., Wu, P., Zhao, S., and Li, X., "Explainable AI for predictive maintenance: A case study on industrial equipment," Expert Systems with Applications, vol. 185, p. 115623, 2021.
18. Tao, F., Zhang, M., and Nee, A. Y. C., Digital Twin Driven Smart Manufacturing. Academic Press, 2022.
19. Singh, J., Darwish, M., and Singh, S., "Concept drift adaptation in industrial IoT: A survey," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 1, pp. 112–128, 2023.
20. Oliveira, L. A. P., Carvalho, J. P., and Sousa, P., "Hybrid edge-cloud architectures for real-time predictive maintenance: A comparative study," Future Generation Computer Systems, vol. 124, pp. 234–248, 2021.
21. Kotsiantis, S. B., Kanellopoulos, D., and Pintelas, P. E., "Handling im-balanced datasets in predictive maintenance applications," International Journal of Computer Science Issues, vol. 17, no. 3, pp. 1–9, 2020.
22. ISO 13374:2020, "Condition monitoring and diagnostics of machinesData processing, communication and presentation," International Organization for Standardization, 2020.
23. IEC 62443-3-3:2021, "Industrial communication networks — Network and system security — System security requirements and security levels," International Electrotechnical Commission, 2021.
24. Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., and Nandi, A. K., "Ap-plications of machine learning to machine fault diagnosis: A review and roadmap," Mechanical Systems and Signal Processing, vol. 138, p. 106587, 2020.
25. Park, K. and Lee, D., "Transfer learning for cross-domain predictive maintenance in heterogeneous industrial environments," IEEE Transac-tions on Automation Science and Engineering, vol. 19, no. 3, pp. 2145–2158, 2022.
26. Rodrigues, F. and Pereira, F. C., "Beyond accuracy: Evaluating predictive maintenance systems with operational metrics," Reliability Engineering & System Safety, vol. 215, p. 107892, 2021.
27. Xu, Y., Sun, Y., Liu, X., and Zheng, Y., "A digital-twin-assisted fault diagnosis method for industrial equipment using deep transfer learning," Journal of Manufacturing Systems, vol. 66, pp. 234–247, 2023.
28. Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., and Taha, K., "Efficient machine learning for big data: A review," Big Data Research, vol. 22, p. 100158, 2020.
29. Schmidt, B. and Wang, L., "Cloud-based predictive maintenance for smart manufacturing: A review," Procedia CIRP, vol. 104, pp. 1589–1594, 2021.
30. Gao, R. X., Wang, L., Helu, M., and Teti, R., "Cognitive computing for human-robot collaboration in industrial settings," CIRP Annals, vol. 71, no. 2, pp. 601–624, 2022.
31. Niu, G. and Yang, B. S., "Machine degradation analysis using hidden semi-Markov models with application to predictive maintenance," IEEE Transactions on Reliability, vol. 69, no. 2, pp. 567–579,

2020.

32. Ferreira, L. L. and Silva, M. A., “Operationalizing AI in industrial maintenance: Lessons from field deployments,” Journal of Industrial Information Integration, vol. 31, p. 100412, 2023.
33. A. Kanawaday and A. Sane, “Machine learning for predictive mainte-nance of industrial machines using IoT sensor data,” in Proc. 8th IEEE Int. Conf. Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 87–90. DOI: 10.1109/ICSESS.2017.8342870.
34. R. Rosati, L. Romeo, G. Cecchini, F. Tonetto, P. Viti, A. Mancini, and E. Frontoni, “From knowledge-based to big data analytic model: A novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0,” Journal of Intelligent Manufacturing, vol. 34, pp. 107–121, 2023. DOI: 10.1007/s10845-022-01960-x.
35. S. Elkateb, A. Me´twalli, A. Shendy, and A. E. B. Abu-Elanien, “Machine learning and IoT-based predictive maintenance approach for industrial applications,” Alexandria Engineering Journal, vol. 88, pp. 298–309, Jan. 2024. DOI: 10.1016/j.aej.2023.12.065.
36. D. Dhinakaran, S. Edwin Raja, R. Velselvi, and N. Purushotham, “Intel-ligent IoT-driven advanced predictive maintenance system for industrial applications,” SN Computer Science, vol. 6, article 151, Feb. 2025. DOI: 10.1007/s42979-025-03695-x.