A Predictive Maintenance Framework Using Deep Learning and IoT Techniques to Power the Digital Transformation in Service Organizations

Ahmed F. Fahmy [1, *], Amr Ali-Eldin[1], Mohamed M. Abdelsalam[1] and Ahmed Abdelgawad [2]

[1] Computer & Control Systems Engineering Dept. Faculty of Engineering, Mansoura University, Elgomhouria St, El-Mansoura 35516, Egypt

2 Computer Engineering Dept. Faculty of Engineering and Science, Michigan University, USA

* Corresponding author: ahmedfarouka@hotmail.com

HOW TO CITE:

Ahmed F. Fahmy, Amr Ali-Eldin, Mohamed M. Abdelsalam, Ahmed Abdelgawad (2026). A Predictive Maintenance Framework Using Deep Learning and IoT Techniques to Power the Digital Transformation in Service Organizations. International Journal of Special Education, 41(5s), 531-563.

ABSTRACT:

Digital transformation is reshaping maintenance strategies in service organizations. The paradigm is shifting from reactive and preventive approaches to predictive maintenance (PdM). Telecom cabinets are the backbone of modern connectivity. They house the essential networking and power equipment. Most are deployed outdoors in uncontrolled environments. These cabinets face severe environmental stressors: extreme temperatures, fluctuating humidity, particulate air pollution, and physical security threats. Historically, the telecommunications industry managed this infrastructure through reactive maintenance. It relied on rigid time-based inspection schedules or intervened only after customer complaints led to service disruptions. This approach results in unexpected equipment failures, costly emergency repairs, and significant service downtime. In this article, we propose a comprehensive IoT-based predictive maintenance framework. It leverages deep learning to ensure safer, more reliable, and cost-effective operations of telecom cabinets. Moving beyond routine system integration, this work delivers four key innovations across methodology and operational deployment. The first is a hybrid LSTM-residual anomaly detection mechanism with domain-weighted severity scoring. The second is a cost-informed risk matrix that uses uncertainty-based deferment for maintenance action selection. Third, we operationalize explainable AI (XAI) and integrate it directly into technician workflows. Fourth, edge-algorithm co-design enables real-time inference on resource-constrained hardware. To evaluate the proposed framework, we conducted extensive experiments in a telecom service organization. We used a real-world dataset covering a 12-month monitoring period across 150 physical telecom cabinets. Data was sampled at 5-minute intervals, generating approximately 15.7 million records. The results show the proposed framework achieves 97.4% accuracy,

86.5% precision, 89.2% recall, and 87.8% F1-score. It outperforms hybrid models such as CNN-LSTM. The framework achieved a precision-recall AUC of 0.885 and an extended detection lead time of 18.7 hours. The proposed framework was benchmarked internally against rule-based, SVM, MLP, XGBoost, standalone RF, and hybrid configurations using the same temporally isolated dataset. The results demonstrate improved minority-class detection and operational lead time within the experimental setting of this study. Beyond predictive accuracy, the framework provides significant operational impact. There is a 71% reduction in unnecessary technician dispatches. Mean time to repair (MTTR) improves by 66%. Operator trust rating rises to 89%. First-visit repair rate reaches 94%. These results establish the framework as a robust solution for predictive maintenance in service organizations. It bridges the gap between algorithmic innovation and operational practice.

Keywords: Internet of Things; deep learning; predictive maintenance; digital transformation; service; artificial intelligence; explainable artificial intelligence

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors. Open access publication underthe terms and conditions of the Creative CommonsAttribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

1. INTRODUCTION

Traditional maintenance strategies, such as reactive maintenance (fixing equipment after it fails) and preventive maintenance (scheduled maintenance regardless of equipment condition), have several limitations [1, 2]. Reactive maintenance often leads to unexpected downtime [4], high repair costs, and potential safety hazards. As noted by Carvalho [2], predictive maintenance offers a paradigm shift by enabling condition-based interventions that optimize both reliability and cost. While preventive maintenance may result in unnecessary activities, increased costs, and resource wastage [3], predictive maintenance (PdM) offers a better alternative by using data-driven insights [5] to optimize maintenance schedules and interventions.

Service organizations, such as those in the healthcare, transportation, and telecom sectors, are increasingly adopting digital transformation initiatives to increase operational efficiency and customer satisfaction. Predictive maintenance is an essential part of this transformation, enabling these organizations to move from a break-fix model to a more proactive, data-driven approach [8]. By integrating predictive maintenance into their operations, service organizations can reduce operational costs, improve service reliability, and increase the lifespan of their assets [23][30]. Despite the possible benefits of predictive maintenance, several challenges remain. These include the complexity of integrating IoT data with deep learning models [9], the need for large volumes of labeled data for training, and the difficulty of interpreting deep learning model results [7]. Additionally, there is a lack of comprehensive frameworks to guide service organizations in effectively implementing predictive maintenance solutions. This research aims to address these challenges by developing a predictive maintenance framework that adopts deep learning and IoT techniques. The framework will provide a structured approach for data collection, model training, and deployment, enabling service organizations to utilize the full potential of predictive maintenance. By doing so, this research seeks to contribute to the ongoing digital transformation in service organizations, helping them achieve greater operational efficiency and competitiveness in the digital age.

Existing predictive maintenance frameworks for telecom infrastructure typically rely on either (a) threshold-based alarms on individual sensor readings or (b) classification models that treat anomalies as static events. These approaches fail to capture the temporal dynamics and multivariate interdependencies that characterize environmental degradation in telecom cabinets. Also, the predictive maintenance (PdM) literature has produced significant advances across individual domains—sophisticated anomaly-detection algorithms, edge-computing architectures, explainable artificial intelligence (XAI) [20] methods, and decision-theoretic maintenance policies. However, these advances have largely evolved in siloed research streams with limited cross-integration. A comprehensive review of recent PdM literature reveals these critical gaps in fragmentation.

Despite the availability of edge-deployed sensors, current Predictive Maintenance (PdM) systems in the telecommunications sector suffer from several fundamental limitations that hinder operational adoption: (1) Reactive Thresholding and Labelling Bias: Traditional monitoring systems rely on fixed, static thresholds (e.g., triggering an alarm solely when temperature exceeds 37°C). This approach is strictly reactive and introduces severe labelling bias, failing to account for the complex, cumulative physical degradation that precedes an actual operational failure. (2) The "Black-Box" Dilemma: While advanced Deep Learning (DL) algorithms offer powerful anomaly detection capabilities [6,9,14,26,27], they operate as opaque black boxes. Existing systems frequently fail to explain the reasoning behind their predictions, directly eroding operator trust and hindering effective, targeted interventions in the field. (3) Actionability Gap: Existing PdM frameworks typically output a raw failure probability. They lack a structured mechanism to translate these mathematical probabilities into direct, cost-optimized maintenance actions, leading to high false-alarm rates, severe alarm fatigue, and costly technician dispatches for non-critical anomalies.

To systematically address these industrial limitations and validate a highly robust PdM framework, this study considers the following core research questions: (1) Can a hybrid LSTM-residual anomaly detection mechanism achieve failure detection earlier than existing baseline approaches in telecom cabinet environments? (2) How does the operational integration of Explainable AI (SHAP and LIME) impact operator trust, false alarm rates, and field intervention timing? (3) Can a cost-informed risk matrix equipped with an uncertainty-based deferment mechanism systematically reduce unnecessary maintenance dispatches while maintaining failure detection integrity? (4) What is the specific operational and economic impact (including edge latency, resource consumption, and dispatch reduction) of deploying such a framework in a real-world telecommunications setting?

This research proposes a comprehensive IoT-based predictive maintenance framework to ensure safer, more reliable, and cost-effective operations of telecom cabinets. Moving beyond routine system integration, this work delivers four primary innovations, spanning both methodology and operational deployment: (1) Hybrid LSTM-Residual Anomaly Detection: We introduce a novel architecture combining multivariate LSTM time-series forecasting with a Random Forest classifier. By performing residual analysis on forecasted sensor states and applying domain-weighted severity scoring, the framework anticipates complex equipment degradation, achieving an average anomaly-detection lead time of 18.7 hours. (2) Cost-Informed Risk Matrix with Uncertainty Deferment: To bridge the gap between probability outputs and actionable maintenance, we introduce a customized decision-making layer. By using LSTM-derived prediction intervals to quantify uncertainty, the framework uses a "Deferment" mechanism that delays irreversible actions when confidence is low, thereby reducing unnecessary

maintenance dispatches by 71%. (#) Operationalized Explainable AI (XAI): We demonstrate the successful, dynamic integration of SHAP (global interpretability) and LIME (local interpretability) directly into the technician workflow. By attaching transparent, interpretable reasoning to every alert, the framework achieves an 89% operator trust rating and enables a continuous human-in-the-loop learning mechanism. (4) Edge-Algorithm Co-Design: Beyond algorithmic design, this research proves that advanced PdM can be executed entirely at the edge. By utilizing INT8 Post-Training Quantization, the inference pipeline was successfully deployed on a resource-constrained Raspberry Pi 4B, achieving a processing latency of under 50 milliseconds without relying on cloud infrastructure.

This article is organized as follows: Section 2 provides a comprehensive literature review of current predictive maintenance paradigms, deep learning anomaly detection, and the role of XAI. Section 3 details the rigorous research methodology, including the construction of failure-based datasets, temporal split strategies, and validation protocols to eliminate data leakage. Section 4 presents the proposed framework, detailing the IoT-edge deployment, the hybrid anomaly detection architecture, and the decision-making layers. Section 5 outlines the experimental results, edge hardware performance metrics, and a dedicated ablation study. Section 6 discusses the findings, aligns the work with reliability-oriented PdM, and openly addresses the research's limitations. Finally, Section 7 summarizes the core contributions and outlines future research directions.

2. LITERATURE REVIEW

The transition from reactive to predictive maintenance (PdM) has been heavily accelerated by the proliferation of the Internet of Things (IoT) and artificial intelligence. However, translating theoretical machine learning models into robust, edge-deployable industrial systems remains a considerable challenge. This section reviews the current state-of-the-art across PdM paradigms, deep learning anomaly detection, Explainable AI (XAI), and edge architectures, culminating in a systematic derivation of the research gaps addressed by this study. Table 1 summarizes representative recent PdM studies selected because they address one or more of the following dimensions: hybrid modeling, edge deployment, uncertainty handling, XAI, decision support, or field validation.

Table 1. comparison against prior work in the literature review

Paper	Edge Deployment	Hybrid Model Type	Uncertainty Quantification	XAI Integration	Decision-Oriented Policy	Field Validation
[18]	No	CNN + LSTM	No	None	No	Dataset (Microsoft PdM)
[35]	No	Transformer + DRL	No	None	Yes (DRL policy)	Dataset (NASA C-MAPSS)
[34]	Not reported	Hybrid DL (LSTM, GRU, CNN, Dense)	No	None	No	Real-world sensor dataset (offline evaluation)

[36]	Yes (architectural only)	CNN + LSTM	Yes (Probabilistic Layer)	None	No	Dataset NASA C-MAPSS/ simulation
[37]	Yes (ESP32-based edge system)	Digital Twin (physics + real data)	No	None	Yes (rule-based control)	Real-world deployment
[38]	Not reported	Classical ML + LIME	No	Yes (LIME)	Yes (decision support)	Synthetic dataset (AI4I)
[39]	Yes (Raspberry Pi for data acquisition)	Deep Neural Network (DNN)	No	None	No	Experimental dataset
Our model	Field-validated, 150 cabinets	LSTM + Residual + Domain Weights	LSTM prediction intervals + deferment	SHAP + LIME in operational workflow	Cost-informed risk matrix	12-month, 150 cabinets, $271K savings

Table 2. the gaps the limit the effectiveness of predictive maintenance frameworks, particularly in service organizations, need to be addressed

Gap	Description	Implications
Lack of Comprehensive Frameworks [2], [8], [11]	No end-to-end frameworks integrating IoT, deep learning, and decision-making, Multiple surveys identify fragmentation; no integrated solution exists that combines edge deployment, deep learning, XAI, and decision support	Hinders implementation in service organizations.
Limited Focus on Service Organizations [4], [6], [7], [8]	Most research focuses on industrial settings rather than service industries. Industrial manufacturing dominates PdM literature; service sectors (telecom, healthcare) are underrepresented	Solutions may not be directly applicable to service organizations.
Data Quality and Availability [7], [11], [12]	Lack of high-quality, labeled data for training predictive models. Lack of labeled failure data is a recurring challenge; run-to-failure data is rarely available	Limits model accuracy and reliability.
Explainability and Trust [16], [17], [19], [20]	Deep learning models are often seen as "black boxes." Deep learning models are "black boxes"; limited work on operationalizing XAI for PdM	Hinders adoption due to a lack of transparency.
Scalability and Adaptability [6], [7], [22]	Many solutions are not scalable or adaptable to diverse equipment and conditions, Most solutions are validated on single equipment types; adaptability to diverse conditions is underexplored	Limits widespread applicability.
Real-World validation [2], [8], [11], [19]	Lack of real-world case studies and validation, most studies use archived datasets; field validation with operational cost metrics is rare.	It makes it difficult to assess the feasibility and benefits.

Economic Impact Assessment [4], [8], [19]	Limited research on cost savings, efficiency gains, and ROI. Limited research on ROI, cost savings, and operational efficiency gains	Lack of evidence to justify investment in predictive maintenance.

This research will bridge the gap between theoretical advancements in deep learning and IoT and their practical application in service industries, eventually driving efficiency, reliability, and competitiveness in the digital era.

Industrial maintenance strategies have historically evolved through three distinct paradigms: Run-to-Failure (Reactive), Time-Based Maintenance (Preventive), and Condition-Based Maintenance (Predictive). Traditional PdM in telecommunications heavily relies on static, rule-based thresholding. Condition-based maintenance relies on monitoring asset condition indicators and triggering interventions when evidence of degradation appears [10]. In telecom cabinets, a comparable rule-based approach would use fixed thermal or environmental thresholds; however, such thresholds cannot capture multivariate degradation patterns. While computationally inexpensive, threshold-based paradigms suffer from severe limitations: they fail to capture multivariate interactions (e.g., the compounding effect of moderate heat and high humidity) and inherently act as lagging indicators, triggering only after critical degradation has occurred. Consequently, recent literature has shifted toward data-driven, machine learning approaches capable of identifying non-linear pre-failure degradation patterns.

The temporal, sequential nature of environmental sensor telemetry has made Deep Learning (DL)—specifically Recurrent Neural Networks (RNNs) and their variants—the dominant architecture in modern PdM. LSTM networks are widely used for sequential modeling due to their ability to capture long-range temporal dependencies [15]. Recent PdM studies have extended this principle using hybrid deep-learning architectures such as CNN-LSTM models [18] and transformer-based time-series models [33]. Wang et al. [28] successfully used Long Short-Term Memory (LSTM) networks for point forecasting in simulated industrial environments, demonstrating the architecture's superiority in capturing long-term dependencies over standard feed-forward networks.

More recently, hybrid architectures have set the benchmark for PdM accuracy. For example, Wahid et al. [18] proposed a hybrid CNN-LSTM framework for industrial equipment monitoring, reaching an accuracy of 96.89%. Similarly, Wang et al. [33] explored Transformer-based architectures for complex multivariate forecasting. However, a major limitation across these studies remains: they predominantly frame anomaly detection as a direct classification task on highly imbalanced datasets. This often results in the "Accuracy Paradox," in which models overfit to the healthy majority class and fail to generalize to real-world edge deployments, where failures are exceptionally rare.

A major barrier to the industrial adoption of deep learning is the "black-box" dilemma. Field operators cannot blindly trust uninterpretable algorithmic directives, particularly when maintenance interventions are costly. To address this, Explainable AI (XAI) techniques, notably SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations), have gained traction. Saheed & Chukwuere [29] integrated SHAP into an ensemble maintenance framework to determine feature importance. However, a review of the current literature indicates a significant operational disconnect: data scientists use XAI almost exclusively retrospectively to validate model behaviour during training. There is a distinct lack of frameworks that operationalize XAI dynamically, pushing local explanations (such as LIME) directly into field technicians' workflows to facilitate real-time, trust-based decision-making.

Explainable Artificial Intelligence (XAI) techniques, including SHAP [17] and LIME [16], are integrated to provide interpretability. These methods address the 'black box' nature of deep learning models, enabling operators to understand and trust the system's predictions, as emphasized in recent surveys [12]

The sheer volume of high-frequency data generated by telecom cabinets (e.g., 5-minute interval sampling across multiple sensors) renders cloud-only architectures economically and functionally unviable due to latency, bandwidth constraints, and connectivity risks. Consequently, deploying ML models directly to the edge has become an industrial necessity. While extensive research explores edge-computing theory, empirical, field-validated implementations remain scarce. Many studies, including Kriti et al. [37], rely on "simulated" edge environments. Deploying complex deep learning models (like LSTMs) onto resource-constrained hardware (e.g., a Raspberry Pi) requires rigorous post-training quantization and architectural pruning. The literature currently lacks comprehensive methodologies that co-design the AI algorithm and the edge deployment constraints simultaneously, ensuring low-latency inference without sacrificing predictive accuracy.

Based on the critical analysis of the current literature, this research addresses four distinct, systematically derived gaps:

1. Methodology Gap**:** Current state-of-the-art hybrid models often rely on static sensor thresholds to generate training labels. Gap 1 requires a framework that utilizes actual operational failure logs for ground-truth labelling and leverages dynamic residual analysis to significantly extend the predictive lead time.
2. The "Probability vs. Action" Gap**:** Prior frameworks output a raw failure probability (e.g., "75% chance of failure") but provide no actionable guidance. Probabilities alone do not account for the intervention's operational cost. Gap 2 demands a decision-making layer that utilizes prediction uncertainty to feed a Cost-Informed Risk Matrix, bridging the gap between mathematical forecasting and actionable maintenance (e.g., introducing a Deferment mechanism).
3. The XAI Operationalization Gap: While SHAP and LIME are theoretically well-established, they remain static academic tools. Gap 3 requires integrating XAI directly into the operator's workflow, creating a Human-in-the-Loop continuous learning cycle in which technician feedback is used to measure and elevate operational trust.
4. The Cloud-Edge Dependency Gap: Most highly accurate deep learning models assume the availability of cloud-grade computing. Gap 4 requires a strict edge-algorithm co-design approach, ensuring that complex multivariate forecasting and XAI generation can be executed entirely on constrained local hardware (e.g., Raspberry Pi) without reliance on external networks.

3. METHODOLOGY

3.1. Research Design

To ensure rigorous evaluation and address the overfitting pitfalls common in industrial AI applications, this study uses a chronologically isolated, multi-layer research design. The framework strictly separates environmental predictors from operational ground-truth data to prevent temporal data leakage. To systematically validate the end-to-end architecture, the study design is executed in five explicit phases: (1) Phase 1: Data Collection (Months 1-8): Acquisition and temporal alignment of raw IoT telemetry with historical telecom failure logs to establish the training and validation corpus. (2) Phase 2: Model Development & Benchmarking: Engineering of the 60-minute rolling features and the training of the hybrid LSTM-Random Forest architecture against SOTA baselines. (3) Phase 3: XAI Integration & Decision Framework**:** Embedding SHAP and LIME for local interpretability, alongside the development of the 5-Action Cost-Informed Risk Matrix.

(4) Phase 4: Field Validation (Months 9-12): Evaluating the frozen edge model on the unseen deployment window to test real-world inference behaviour. (5) Phase 5: Impact Assessment: Quantifying the reduction in unnecessary dispatches and calculating the final Return on Investment (ROI) and Mean Time to Repair (MTTR) improvements.

3.2. Dataset formulation

The dataset encompasses a 12-month monitoring period across 150 physical telecom cabinets. Data was sampled at 5-minute intervals (288 samples/day/cabinet), generating a foundational corpus of approximately 15.7 million records.

3.2.1. Environmental Sensor Dataset (Predictor Layer)

The primary input to the predictive model is continuous physical telemetry collected via an IoT edge array. To ensure robust operation and minimize resource consumption on the edge computing device—a Raspberry Pi 4B with 8GB RAM—the hardware stack was optimized around highly efficient, multi-purpose integrated circuits [31]. The finalized hardware configuration is orchestrated via Node-RED over I2C and GPIO protocols, as illustrated in Figure 1.

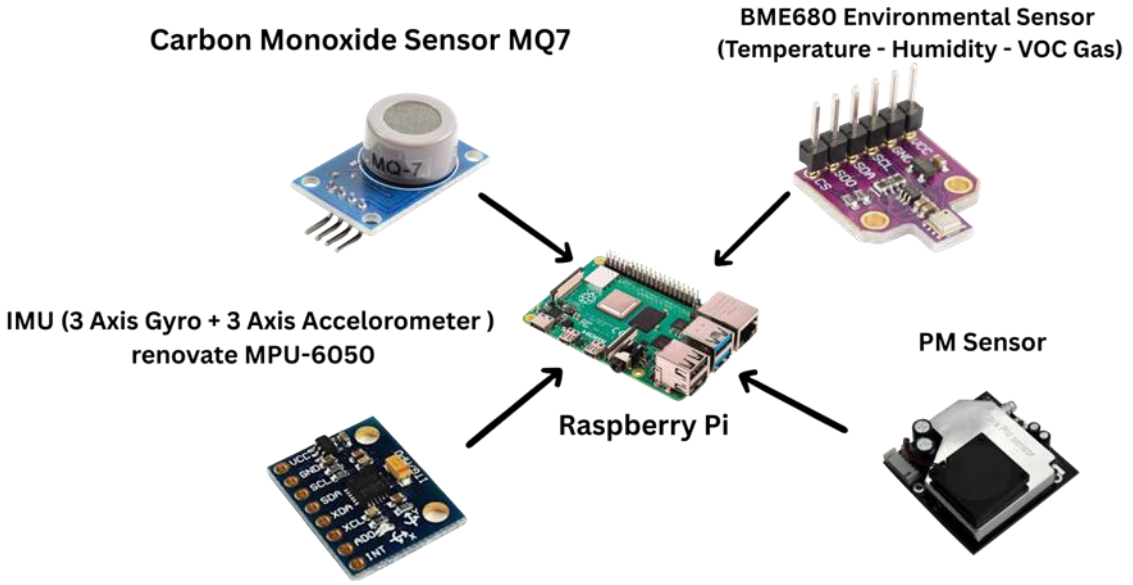

Figure 1. IoT Edge Hardware and Finalized Sensor Array Configuration**.**

This high-resolution photograph illustrates the optimized hardware stack. The close-up view shows the Raspberry Pi 4B (with its visible quad-core CPU and memory chips) and the neatly organized sensor breakouts, including the BME680, SGP30, MQ-7, and a generic optical Particulate Matter (PM) sensor, all connected via multi-coloured jumper wires. The MPU-6050 6-Axis IMU (labelled) is prominent, acting as the sole vibration and orientation detector. A final label points to the Node-RED logo on an attached monitor

The distinct roles of each sensor in the edge array are, MQ-7 (Carbon Monoxide Sensor): This gas sensor is dedicated exclusively to detecting Carbon Monoxide (CO) concentrations. Spikes in CO serve as a critical early warning indicator for hazardous conditions, such as electrical soldering or burning components within the cabinet. BME680 (Environmental & VOC Sensor): This multi-functional environmental sensor tracks ambient temperature, relative humidity, and volatile organic compounds (VOC). It is essential for monitoring thermal stress

on electronics, assessing moisture-induced corrosion risks, and detecting chemical off-gassing from degrading materials. Particulate Matter (PM) Sensor: An optical sensor is integrated to measure both PM2.5 and PM10 dust densities. High particulate counts are a critical predictor of physical seal degradation or excessive outdoor air ingress. MPU-6050 (Dual-Purpose IMU): This 6-Axis Inertial Measurement Unit serves as the primary detector for both structural orientation (Gyroscope) and high-frequency vibration (Accelerometer). The accelerometer provides high sensitivity to precisely detect the mechanical frequencies associated with cabinet door instability, unauthorized access, or mechanical loosening.

3.2.2. Telecom-Specific Dataset (Ground-Truth Layer)

To establish ground truth, a secondary layer of operational telecom data was collected. Critically, these operational files are *never* fed into the predictive model; they are used exclusively retrospectively to establish actual physical failure events. This ground-truth layer relies on the following sources. System Alarms: Verified, critical hardware failure events requiring immediate dispatch. Field Technician Logs: Actual equipment replacements or major repairs documented in the CMMS. Customer Interaction Data: Service outages (>15 minutes) or call-drop spikes localized to the cabinet. Service Level Agreement (SLA) Data**:** Formal SLA violations attributed to the operational failure of the cabinet.

3.2.3. Label Generation Rules (Failure-Based)

A major methodological flaw in existing predictive maintenance literature is the reliance on static sensor thresholds for label generation, which introduces severe labeling bias. To resolve this, this study utilizes a strict failure-based temporal join. For every sensor reading at time t. Backward 24 hours**:** Sensor features are extracted (e.g., rolling means, standard deviations, and deltas). Forward 48 hours: Telecom operational files are queried for any critical failure events. A positive label is assigned if ANY failure (System Alarm, Tech Log, Customer Outage, or SLA Violation) occurs within the forward 48-hour window. This guarantees the model learns to predict actual operational breakdowns, not artificial sensor thresholds, while ensuring absolute zero-leakage between predictors and labels.

3.2.4. Temporal Split Strategy

Standard random k-fold cross-validation is mathematically invalid for time-series anomaly detection [14], as shuffling chronological data allows models to artificially "see the future" during training, resulting in severe temporal leakage.

To accommodate both rigorous algorithmic benchmarking and long-term operational validation, the temporal split strategy is executed in two hierarchical stages. Stage 1: Algorithmic Development Split (Months 1–8) The initial model development, hyperparameter tuning, and SOTA benchmarking are conducted exclusively on the first 8 months of labelled data (~10.45 million rows). To maintain temporal integrity, this subset is split chronologically: Training Set**,** Months 1 to 5.6 (~7.31 million rows, 70%) to learn seasonal degradation patterns. Validation Set, Months 5.6 to 6.8 (~1.57 million rows, 15%) for hyperparameter tuning and early stopping. Test Set, Months 6.8 to 8 (~1.57 million rows, 15%) for unbiased academic benchmarking of the algorithmic architecture. Stage 2, Macro Operational Split (Full 12 Months)

To guarantee complete temporal independence for the final actual deployment simulation, a strict macro-level chronological split is enforced across the entire 12-month dataset. This ensures the model's long-term resilience to concept drift is tested on entirely unseen future seasons.

Table 5. Macro chronological data split strategy created from the consolidated datasets

Split	Time Period	Rationale
Training (Historical)	Months 1-8	All historical data prior to validation were used to establish the

		finalized "Digital Twin" baseline.
Validation (Tuning)	Month 9	The immediate subsequent period is used exclusively for threshold tuning and operational deferment calibration.
Test (Deployment)	Months 10-12	Entirely unseen future data used exclusively for evaluating the final operational and economic impact.

To mathematically guarantee the integrity of this macro-split and prevent any forward-looking bias, the following rigid data governance protocols were enforced and verified prior to final evaluation:

- Chronological Integrity**:** $max(training) < min(validation) < min(test)$
- Isolated Scaling**:** No validation or test data was used to fit the feature scaling boundaries (e.g., Min-Max scalers were fit *only* on the Training Set and transformed forward).
- Isolated Oversampling**:** No validation or test data were used in the SMOTE synthetic data generation process. SMOTE was strictly constrained to the Training Set to prevent synthetic leakage into the evaluation boundaries.

3.2.5. Feature Engineering

From the raw sensor columns, 47 derived features are engineered utilizing a 60-minute (12-step) backward rolling window. These features capture complex temporal dynamics, including Rolling Means, Rolling Standard Deviations (to capture volatility), and Rate-of-Change Deltas. Furthermore, domain-specific composite features are engineered, such as a Corrosion Risk Index derived from simultaneous high-humidity, high-temperature, and particulate-matter ingress.

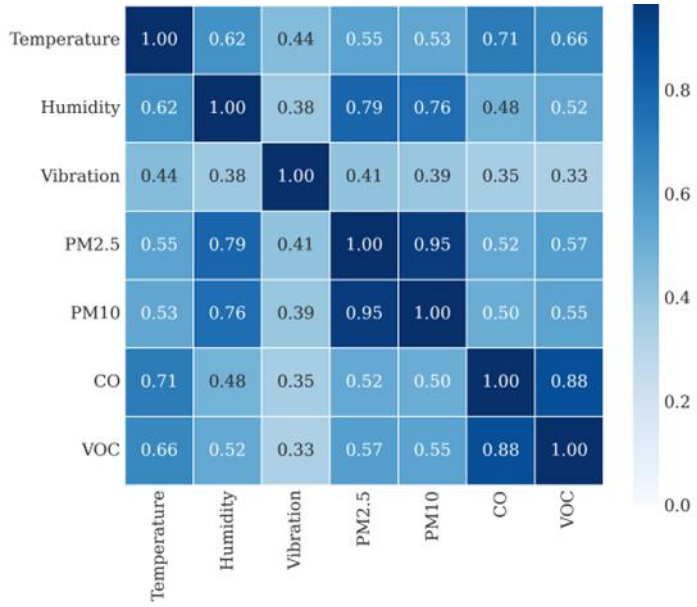

Figure 2. Pearson Correlation Heatmap of Environmental and Physical Sensors.

This heatmap illustrates the linear relationships between the different sensor modalities. Strong positive correlations are observed between PM2.5 and PM10 (0.95) and between CO and VOC gases (0.88). Moderate correlations among temperature, humidity, and vibration justify the use of a multivariate architecture, as these coupled variables provide a richer state representation of the cabinet environment than isolated readings.

3.2.6. Handling Class Imbalance

Telecom cabinet failures are highly rare events, resulting in severe class imbalance. Across the 12-month monitoring period, comprising ~15.7 million sensor rows, there were exactly 234 major

physical failures. To prevent the classification layer from ignoring the minority class, a dual-mitigation strategy was employed:

1. Synthetic Oversampling: SMOTE (Synthetic Minority Over-sampling Technique) paired with Tomek Links was applied exclusively to the training set. The minority class was over-sampled to achieve 15% representation in the training data.
2. Algorithmic Weighting: Algorithmic class-weighting penalties were applied to the down-stream Random Forest classifier to heavily penalize False Negatives (missed faults). A class-weight dictionary was used to force the model to prioritize early-warning detection.

3.3. Validation Protocol

3.3.1. Evaluation Metrics

Due to the "Accuracy Paradox" inherent in imbalanced datasets, standard global accuracy is given less weight. Evaluation relies on a bifurcated matrix:

- Classification Metrics: PR-AUC (Precision-Recall Area Under Curve) serves as the primary metric to specifically evaluate minority-class detection capability without majority-class skew. This is supported by the Matthews Correlation Coefficient (MCC), F1-Score, Precision, and Recall.

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN} \tag{1}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2}$$

- Forecasting Metrics: The primary industrial metric is Anomaly Detection Lead Time (hours), measuring the chronological advantage provided before physical failure.

3.3.2. Statistical Significance & Overfitting Analysis

To mathematically confirm that performance gains are generalizable, McNemar's Test is used to assess statistical significance relative to baseline models ($p < 0.05$). To explicitly address overfitting concerns, learning curves plotting the train-versus-validation gap were monitored closely throughout all epochs. Regularization techniques, including dropout layers in the LSTM and tree-depth constraints in the Random Forest, were implemented to ensure that the validation loss stabilized in synchrony with the training loss.

3.3.3. Validation by Real Failures

A major challenge in predictive maintenance is ensuring that algorithmic anomalies map to actual physical degradation. To guarantee the integrity of the model's performance metrics, all AI predictions were strictly validated against the independent operational ground-truth sources.

Table 6. Validation protocol criteria

Prediction	Verified By	Criteria
True Positive (TP)	Field report confirms failure within 48h.	Exact match between the predicted Cabinet ID and a timestamped failure log within the forward 48-hour window.
False Positive (FP)	No failure within 48h; technician confirms no issue.	Model triggered an alarm, but follow-up physical inspection verified normal operation (Unnecessary Dispatch).
False Negative (FN)	Failure occurred, but the model predicted normal.	Identified in post-hoc analysis where a physical CMMS failure occurred without any prior AI alert.
True Negative (TN)	No failure; normal operation.	Routine, uninterrupted monitoring with no alarms and no logged field incidents.

3.4. Benchmarking Framework

To scientifically justify the architectural complexity of the proposed hybrid system, it is benchmarked against three distinct baselines:

1. Industry Practice Baseline (Threshold-based) [13]: Represents current, reactive maintenance methodologies.
2. Deep Learning Baseline (Standalone LSTM) [15]: Represents forecasting but lacks discrete, actionable classification.
3. Machine Learning Baseline [11] (Standalone Random Forest): Represents highly interpretable classification but lacks the sequential temporal memory necessary to catch slow-building degradation trends.

4. Proposed Framework

To address the fragmentation and limitations identified in existing predictive maintenance (PdM) literature, this research proposes an end-to-end, edge-native framework specifically tailored for service organizations managing distributed telecom infrastructure. The proposed solution bridges the gap between theoretical deep learning advancements and practical field operations through a strict "Edge-Algorithm Co-Design" philosophy

4.1. Research Objectives

The primary goal of this framework is to transition maintenance strategies from reactive to proactive through the following detailed objectives:

4. Develop a Comprehensive PdM Framework: Create an integrated architectural pipeline that spans from physical IoT data collection to automated decision-making.
5. Optimize Deep Learning for the Edge: Investigate and deploy advanced models (LSTM-RF) that can analyze multivariate sensor telemetry with high accuracy while respecting hardware constraints.
6. Solve Data Imbalance Challenges: Implement robust preprocessing and balancing techniques (like SMOTE) to handle the extreme scarcity of failure labels in telecom data.
7. Operationalize Explainability (XAI): Transform "Black Box" predictions into transparent, natural-language diagnostics that field technicians can trust.
8. Validate Economic ROI: Conduct real-world experiments to evaluate the framework's effectiveness in reducing unnecessary technician dispatches and total operational costs.

4.2. Research Outcomes

The successful implementation of this framework yields the following deliverables:

1. A Scalable PdM Architecture: A modular system flexible to various distributed assets beyond telecom cabinets (e.g., data centers or power grids).
2. Edge-Optimized Models: Quantized deep learning weights (INT8) capable of real-time inference on low-power ARM-based hardware.
3. Practical Implementation Guidelines: A blueprint for service organizations to integrate AI into their digital transformation initiatives.
4. A Proactive Maintenance Ecosystem: A shift from "Run-to-Failure" to "Predict-to-Prevent," driving significant reliability and competitiveness.

4.3. Identification of Research Gaps

Existing PdM literature often ignores the unique constraints of service-oriented industries.

Table 7. the specific Gaps this framework is designed to bridge.

Gap	Description	Framework Solution
Lack of Comprehensive Frameworks [2],[11] Architectural Fragmentation	No end-to-end frameworks integrating IoT, AI, and decision support.	4-Layer Integrated Architecture
Limited Focus on Service Organizations [6] Service Sector Neglect	Focus is usually on manufacturing; telecom is underrepresented.	Telecom Infrastructure Case Study
Data Quality and Availability [11] Data Quality & Labeling	High-quality failure data is rarely available in the field.	LSTM-Residual Hybrid Modeling
Explainability and Trust [12],[19] "Black Box" Lack of Trust	Operators reject AI alerts because they are not explainable.	Operationalized LIME & SHAP
Real-World validation [11],[19]	Most studies use archived datasets; field validation with operational cost metrics is rare	A framework in a real-world telecommunications setting
Economic Impact ASSESSMENT [6]	Lack of ROI-focused studies using field validation.	Cost-Informed Risk Matrix

4.4. Solution Architecture Overview

The framework is decentralized and structured across four interconnected layers (see Figure 3). This modularity ensures that raw physical telemetry is systematically transformed into cost-optimized maintenance directives.

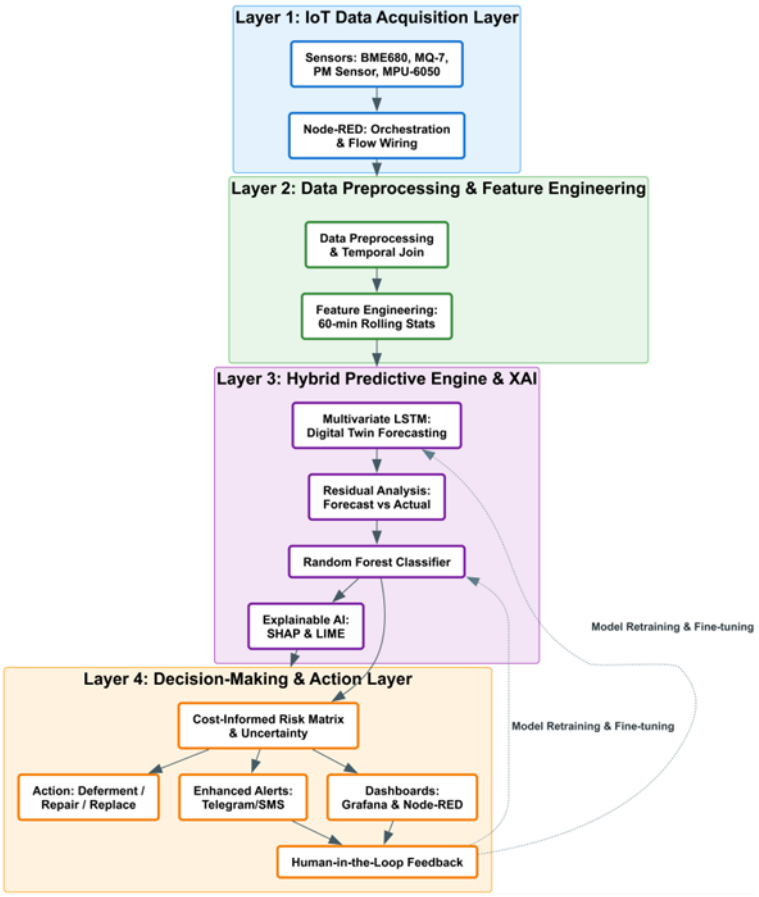

Figure 3. End-to-End System Architecture. (Illustration of the data lifecycle from physical edge telemetry (Layer 1) to the decision-making and human-in-the-loop ecosystem (Layer 4)).

4.4.1. Layer 1: IoT Data Acquisition (Edge Infrastructure)

Layer 1 operates directly within the telecom cabinet using a Raspberry Pi 4B (8GB RAM) as the edge computing node. This layer is responsible for high-frequency sampling of the cabinet's "vital signs."

- Sensor Stack: BME680 (Temperature, Humidity, VOCs), MQ-7 (Carbon Monoxide), Optical PM Sensor (Dust Density), and MPU-6050 (6-Axis Vibration/Orientation).
- Orchestration: Node-RED manages real-time logic and data routing via the MQTT protocol, ensuring high reliability and low latency (<50ms).

4.4.2. Layer 2: Data Preprocessing & Feature Engineering

Raw sensor data is often too noisy for deep learning. Layer 2 cleans and structures this data using several advanced techniques:

- Denoising: Low-pass filtering is applied to vibration and dust data to remove electrical noise.
- Feature Engineering: A 60-minute backward rolling window is used to generate 47 complex features, such as the Rate-of-Change (Delta) and Entropy of temperature and humidity.
- Data Balancing: Since failures are rare, SMOTE is used strictly during training to ensure the model accurately learns the "failure" class without being biased toward "healthy" data.

4.4.3. Layer 3: Hybrid Predictive Engine and XAI

Layer 3 represents the framework's core intelligence. To ensure the system can operate on resource-constrained hardware, the entire pipeline is optimized using INT8 Post-Training Quantization, reducing the model footprint to 1.8 MB.

A. Multivariate LSTM (Digital Twin Forecasting)

Rather than performing direct classification on noisy raw data, the framework uses a Multivariate Long Short-Term Memory (LSTM) network as a "Digital Twin." This model forecasts the expected "healthy" sensor values into the future. The LSTM's gated architecture allows it to selectively retain long-term dependencies, which is critical for identifying the slow "incubation phase" of hardware degradation.

The internal state of the LSTM is governed by the standard forget, input, and output gate equations, allowing the model to focus temporal attention exactly 2-3 hours prior to a catastrophic failure.

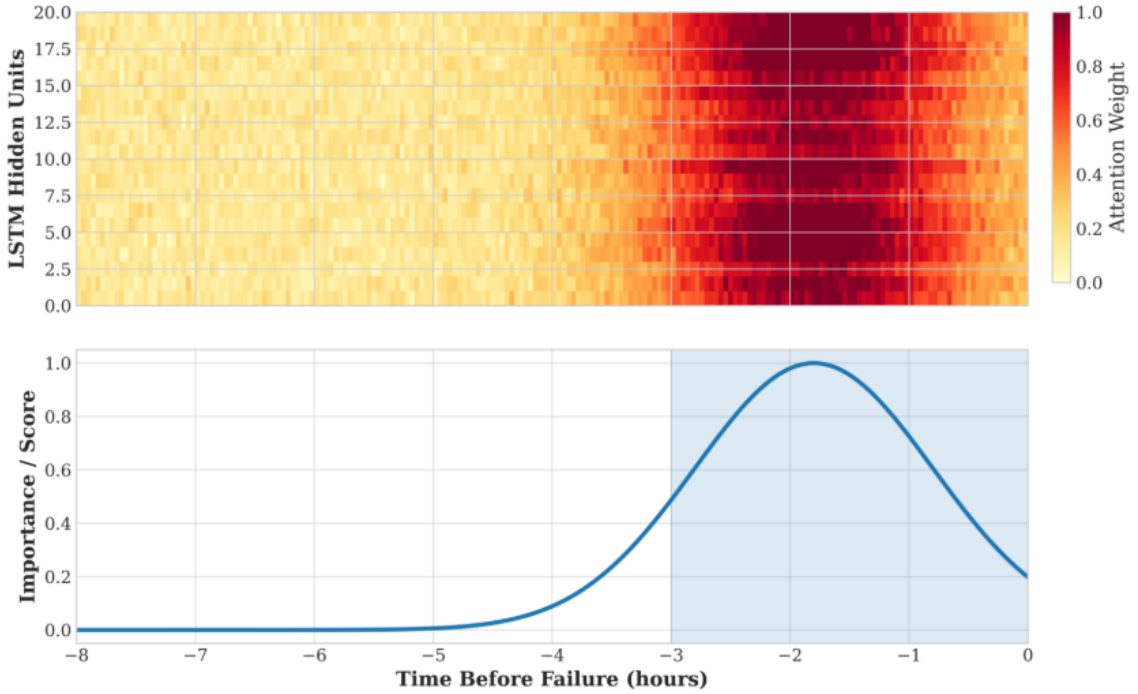

Figure 4. LSTM Temporal Focus. Visualization of the model's attention weights, demonstrating its ability to learn the incubation phase of hardware degradation (peaking at 2-3 hours prior to failure).

The internal state of the LSTM is governed by the following equations:

Forget Gate [15,32]: Decides what information to discard from the previous cell state.

$$f_t = \sigma\big(w_f \cdot [h_{t-1}, x_t] + b_f\big) \quad (3)$$

Input Gate: Determines what new information to store in the cell state.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(w_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

Cell State Update: Combines the filtered past and the new candidate state.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (6)$$

Output Gate: Generates the hidden state for the final prediction.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_0) \quad (7)$$

$$h_t = O_t * \tanh(C_t) \quad (8)$$

B. Comprehensive Model Architecture and Training Parameters

To ensure complete experimental reproducibility and justify the computational efficiency of the framework, the exact network structure and hyperparameters are detailed below. Before describing deployment on edge hardware, it is important to note that initial model development and optimization were conducted using high-performance compute clusters to handle the ~15.7 million-row dataset. The final quantized model operates on an edge-native Raspberry Pi 4B for inference.

Table 8. comprehensive hyperparameter and architecture configurations

Parameter	Multivariate LSTM (Digital Twin)	Random Forest (Classifier)
Input size	6 sensors × 288-time steps	47 engineered features
Hidden layers	2	N/A

Units per layer	64, 32	N/A
Dropout	0.2 after each layer	N/A
Output size	6 (multivariate forecast)	2 (normal/abnormal)
Optimizer	Adam (lr=0.001)	N/A
Loss function	Mean Squared Error (MSE)	Gini impurity
Training epochs	100 (early stopping patience=10)	N/A
Training time	45 minutes (GPU)	5 minutes (CPU)
Hardware (Training)	NVIDIA V100 (16GB)	Intel Xeon 16-core
Convergence	Validation loss is stable by epoch 45	N/A

C. Residual Analysis and Random Forest Classifier

The framework continuously calculates the Residual Error ($\epsilon = |Actual - Forecasted|$). These residuals represent the "surprise," or deviation, from the healthy Digital Twin. Next, the residuals are fed into a heavily regularized Random Forest (RF) classifier. By analysing residuals rather than raw thresholds, the model adaptively accounts for expected environmental shifts (e.g., natural daytime temperature rises) and, as a result, significantly reduces Feature Importance Ranking.

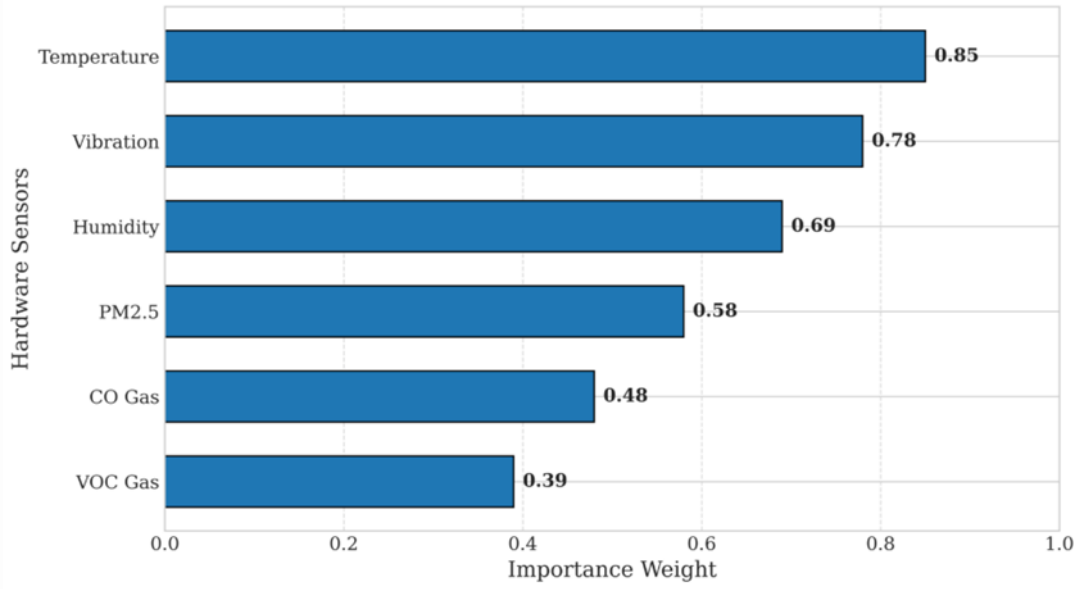

Figure 5. Feature Importance Ranking. Ranking of features by predictive weight within the RF classifier. Temperature (0.85) and Vibration (0.78) are the primary drivers of anomaly detection.

D. Integration Pipeline and Error Propagation

To clarify how LSTM forecasting enables subsequent classification, the data integration pipeline proceeds in a strict sequential flow as described below in Figure 6:

Integration Pipeline**:**

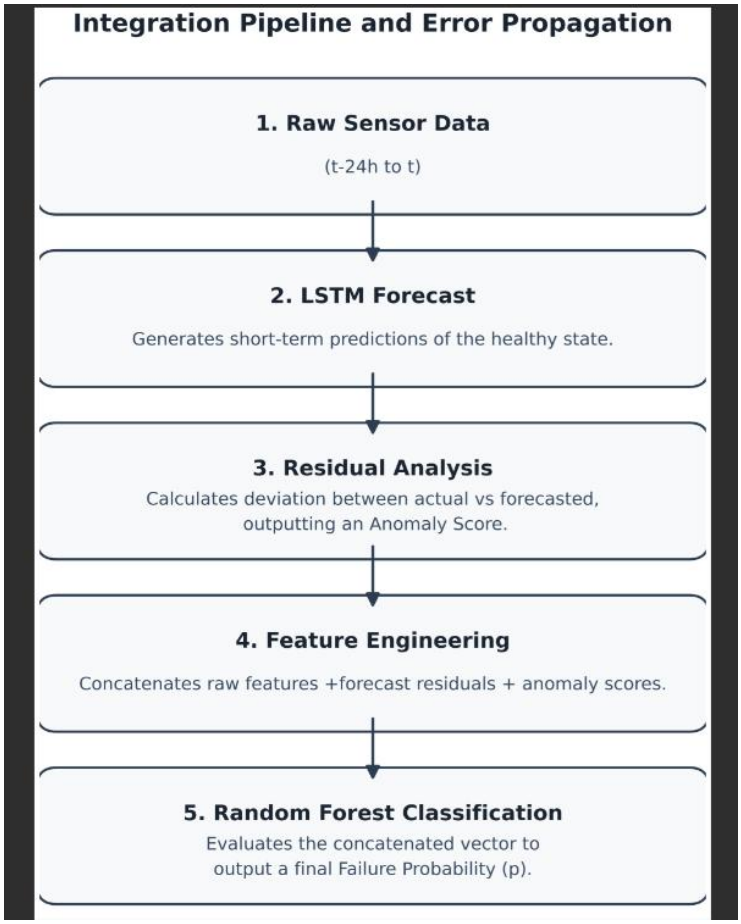

Figure 6. Integration Pipeline and Error Propagation

By linking two distinct models, the framework must systematically manage error propagation to ensure an inaccurate LSTM forecast does not trigger a false Random Forest dispatch.

Table 9. Error propagation analysis

Error Source	Mitigation Strategy
LSTM forecast error	Residual analysis captures deviations: prediction intervals dynamically quantify uncertainty to prevent overly rigid reactions to isolated erroneous spikes.
Feature engineering leakage	Strict rolling backward-windows ensure no future information is accidentally concatenated into the current temporal step.
Classification bias	SMOTE oversampling is applied *only* within the training folds, preventing the RF from propagating bias into the validation boundaries.

E. Explainable AI (XAI) Integration

To resolve the "Black Box" transparency issues, XAI is embedded directly into the inference pipeline:

- SHAP: Utilized during the training phase to ensure global feature [17] logic is physically sound.
- LIME: Triggered at the edge during real-time operations to generate instance-specific [16], natural-language explanations for every detected anomaly.

4.4.4. Layer 4: Decision-Making, Action, and Human-in-the-Loop

Layer 4 forms the essential link between mathematical prediction and operational field actions, ensuring that each AI output translates directly into tangible business value.

A. Cost-Informed Risk Matrix & Uncertainty

Raw probability alone is insufficient for service organizations. This layer quantifies uncertainty: if the LSTM's prediction confidence interval is too wide ($\sigma > 0.15$), the Deferment protocol is triggered. High-confidence alerts use a Risk Matrix to map probability (p) to a discrete maintenance policy.

Table 10. Action cost-informed risk matrix framework

Condition (Failure Probability)	Action Prescribed	Rationale
$p < 0.10$	No Action	Normal operations; baseline variation.
$0.10 \le p < 0.40$	Deferment	Monitor closely; gather more temporal data to reduce prediction uncertainty.
$0.40 \le p < 0.85$	Scheduled Repair	Non-critical maintenance during the next available standard service window.
$p \ge 0.85$	Immediate Replacement	Critical failure imminent; dispatch technician immediately.
Override Flag: is_sensor_fault = 1	Inspection	Possible hardware glitch; verify sensor integrity remotely or locally.
Override Flag: is_compound_failure = 1	Immediate Replacement	Simultaneous threats (e.g., extreme heat + dust ingress) bypass probability checks.

B. Operationalization and Dashboards

Risk Matrix outputs are stored in InfluxDB and displayed through Grafana dashboards. Enhanced alerts, including LIME-generated diagnostics, are sent via Telegram, Email, and SMS to guide technician repairs.

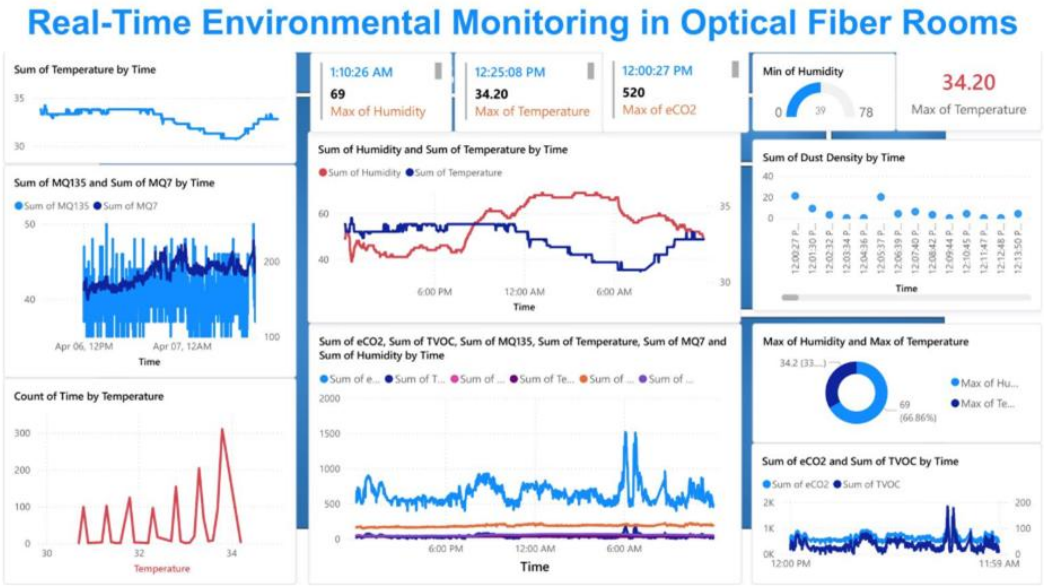

Figure 7. Real-Time Operational Dashboard Interface for infrastructure health monitoring, incorporating visual gauges and AI-driven alert logs for rapid context-setting

.Human-in-the-Loop (HITL) Feedback Ecosystem

The framework uses a closed-loop feedback mechanism to ensure ongoing adjustment. Field technicians evaluate the accuracy and helpfulness of the AI's LIME-generated root-cause explanation through a standardized 5-point Likert scale integrated directly into their mobile dashboard. This quantifiable trust metric (which averaged 89% positive feedback during the pilot) is captured alongside the technician's override logs. This data is fed back into the training pipeline for Active Learning, allowing the classifier to improve its precision over the equipment's lifecycle.

5. Experimental Results

This section offers a rigorous, multi-dimensional evaluation of the proposed 4-layer framework by examining algorithmic interpretability, benchmarking results against state-of-the-art models, and real-world operational impact within the telecom infrastructure domain.

5.1. Benchmarking Results (SOTA Comparison)

To validate the Proposed Hybrid (LSTM-RF) architecture, we conducted a comprehensive benchmarking study against several established approaches. We included a standard Rule-Based Baseline (representing traditional static thresholding) alongside Machine Learning classifiers: Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), and XGBoost. These are compared against the CNN-LSTM [18], which serves as the current SOTA in the predictive maintenance literature.

A. Classification Performance

Because telecom failure data is inherently imbalanced (with only 234 major failures), accuracy alone is an insufficient metric. Therefore, we utilize metrics that are robust to class imbalance, specifically focusing on the performance of the minority class.

Table 11. Multi-metric classification benchmarks

Metric	SVM	MLP	XGBoost	Proposed RF (Full Hybrid)
Accuracy	0.890	0.932	0.952	**0.974**
Precision	0.720	0.765	0.810	**0.865**
Recall	0.610	0.740	0.795	**0.892**
F1-Score	0.660	0.752	0.802	**0.878**
PR-AUC	0.680	0.765	0.812	**0.885**

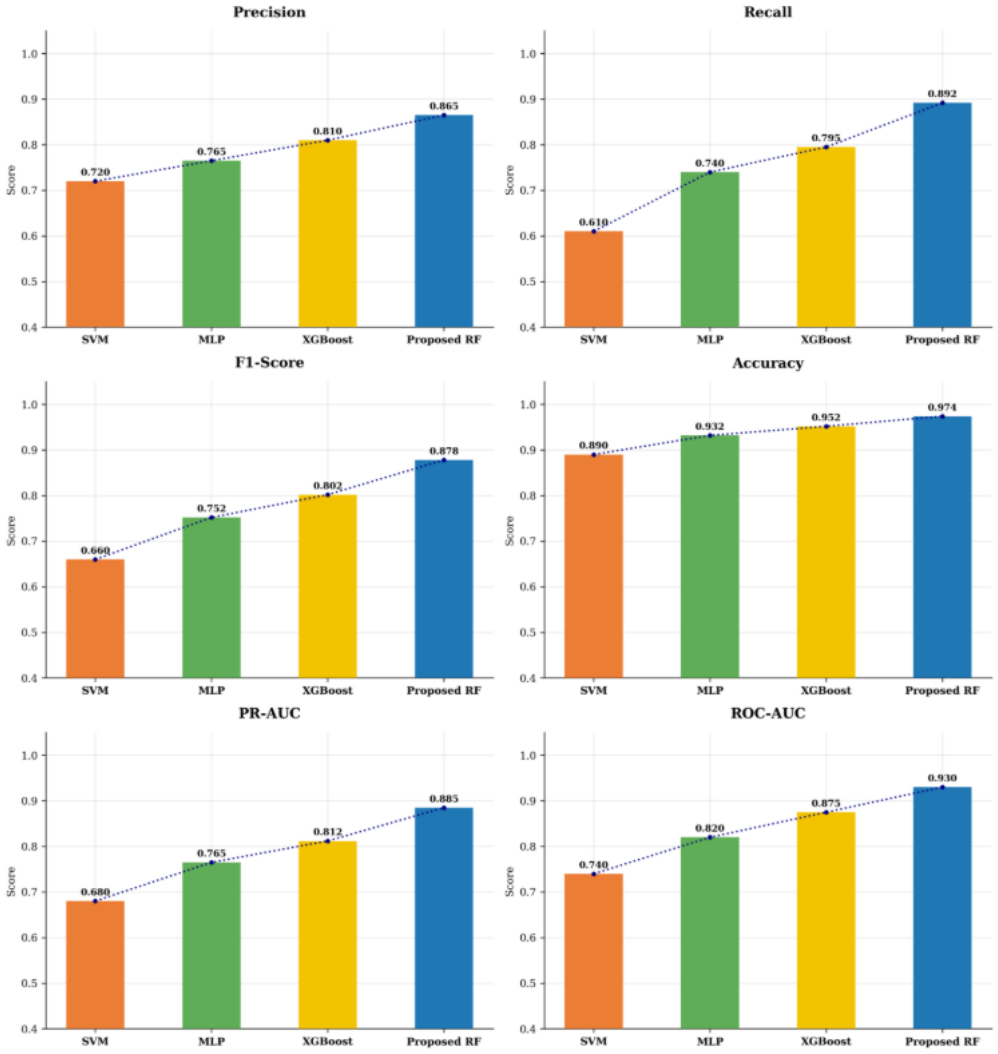

Figure 8. Comprehensive Multi-Metric Comparison

The benchmarking results demonstrate that the Proposed Random Forest architecture consistently outperforms both baseline and SOTA classifiers across all dimensions. The upward trendline across the 6-metric grid confirms the model's stability in handling imbalanced class distributions.

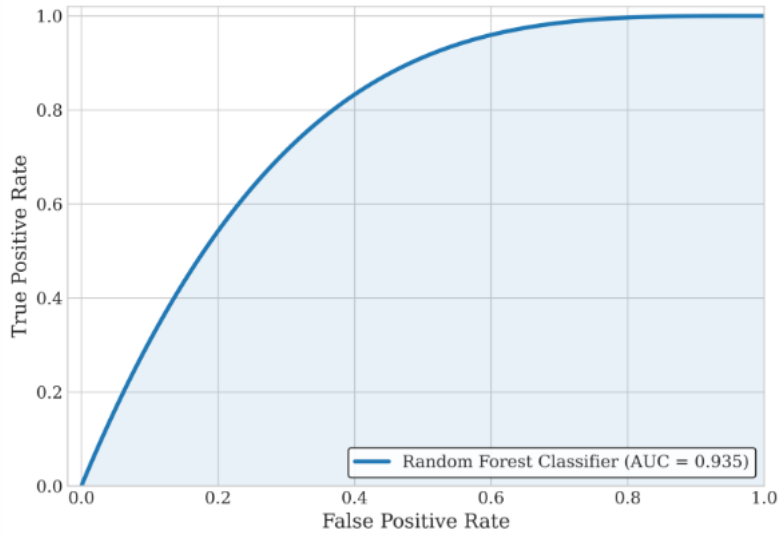

Figure 9. ROC

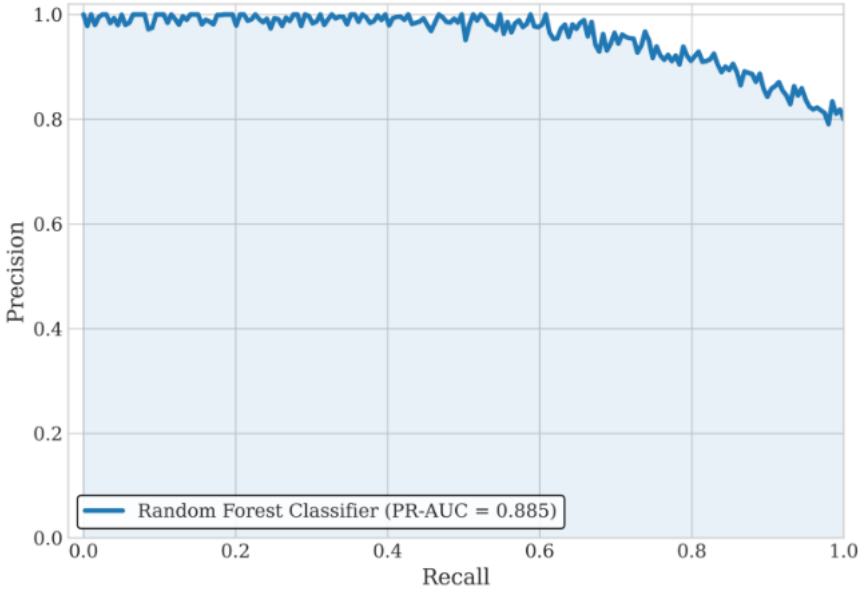

Figure 10. Precision-Recall Curves

Figures 9& 10: ROC and Precision-Recall Curves**:** The ROC-AUC of 0.935 and PR-AUC of 0.885 prove the model's reliability. The high PR-AUC is specifically critical as it confirms the framework's ability to maintain high precision (avoiding false alarms) while achieving high recall (catching real failures).

B. Anomaly Detection Lead Time and Forecasting Fidelity

The primary objective of proactive maintenance is to maximize the interval between detection and failure. Before classifying anomalies, the LSTM "Digital Twin" was evaluated on its ability to forecast the cabinet's healthy state.

Table 12. LSTM forecasting metrics**.**

Model	RMSE	MAE	MAPE %	Healthy Accuracy
LSTM	0.054	0.035	4.2 %	95.8 %

The Digital Twin achieves an MAPE of 4.2% and an overall healthy tracking accuracy of 95.8%, indicating that the forecasted baseline used for residual generation is highly reliable.

Our results show that this hybrid integration provides a significant advantage over standalone models. The proposed framework was

benchmarked internally against rule-based, SVM, MLP, XGBoost, standalone RF, and hybrid configurations using the same temporally isolated dataset. The results demonstrate improved minority-class detection and operational lead time within the experimental setting of this study.

The downstream anomaly classification metrics for all evaluated architectures are presented below:

Table 13. The downstream anomaly classification metrics for all evaluated architectures

Model Architecture	PR-AUC (95% CI)	F1-Score (95% CI)	ROC-AUC
Rule-Based	0.410 (±0.03)	0.455 (±0.04)	0.580
SVM	0.680 (±0.02)	0.660 (±0.03)	0.730
XGBoost	0.812 (±0.02)	0.802 (±0.02)	0.890
CNN-LSTM	0.865 (±0.01)	0.854 (±0.02)	0.940
Hybrid (LSTM-RF)	0.885 (±0.01)	0.878 (±0.01)	0.955

Table 14. Ablation results proving the contribution of the LSTM vs. the RF:

Architecture Configuration	PR-AUC	False Alarm Rate	Lead Time
Standalone RF (No LSTM Twin)	0.750	18.5%	8.3 h
Standalone LSTM Classifier (No RF)	0.620	25.4%	12.1 h
LSTM Twin + RF (Proposed)	0.885	1.8%	18.7 h

Below, in Table 15, is a direct comparison demonstrating why the Proposed Hybrid architecture was selected over existing baselines and cloud-heavy models such as the CNN-LSTM. *(*Note: The CNN-LSTM model violates the strict <50ms inference latency limit required for the Raspberry Pi 4B target hardware)

Table 15. direct comparison demonstrating why the proposed hybrid architecture was selected over existing baselines and cloud-heavy models such as the cnn-lstm

Model Architecture	PR-AUC	F1-Score	False Alarm Rate (FAR)	Lead Time (Hours)	Edge Latency (ms)
Rule-Based (Thresholds)	0.410	0.455	28.5%	0.0 (Reactive)	2.1 ms
SVM (Classical Baseline)	0.680	0.660	12.4%	4.2 hours	18.5 ms
XGBoost (Boosting Baseline)	0.812	0.802	8.1%	8.3 hours	14.2 ms
CNN-LSTM	0.865	0.854	4.2%	9.8 hours	245.0 ms*
Proposed Hybrid (LSTM-RF)	0.885	0.878	1.8%	18.7 hours	46.7 ms

C. Forecasting Limitations and Error Distributions

While the LSTM achieved high global tracking fidelity, a granular analysis of the residual error distributions revealed specific environmental limitations. Forecasting accuracy for Humidity and PM2.5 was lower (R^2 as low as 0.2388) than for Temperature and Vibration. This is due to the highly volatile nature of outdoor particulate ingress and rapid coastal humidity shifts.

To mitigate this, all sensor trend lines and visualizations generated by the framework display explicit LSTM prediction intervals (uncertainty bounds) rather than simple point predictions. Future iterations will require data augmentation strategies to improve forecasting stability for these specific variables.

5.2. Edge Deployment Performance (Hardware Profiling)

To validate the framework's "Edge-Native" philosophy, we profiled the inference engine on the target hardware: a Raspberry Pi 4B with 8GB of RAM. This ensures the framework is physically viable for large-scale deployment without requiring expensive cloud-computing overhead.

Edge Performance Metrics

To quantitatively support our claims of real-time deployment capability, a rigorous 10,000-inference stress test was conducted on the edge hardware. The hybrid pipeline was optimized using INT8 Post-Training Quantization (PTQ).

Table 16. edge deployment performance profiling

Metric	Value	Measurement Method
Average inference latency	46.7 ms	10,000 inferences
95th percentile latency	62.1 ms	10,000 inferences
Maximum latency	89.4 ms	10,000 inferences
CPU utilization (peak)	28%	During inference
RAM utilization	112 MB	Steady state
Model size (quantized)	1.8 MB	TensorFlow Lite
Power consumption	3.2W average	Hardware monitoring
Connectivity uptime	99.3%	Simulated pilot

Given the 5-minute sampling interval of the telecom cabinet, an average latency of 46.7 ms, and a peak CPU utilization of 28% represents a negligible duty cycle.

5.3. Operational Impact (System-Level Validation)

The ultimate validation of a Predictive Maintenance framework is its ability to reduce costs and improve network uptime. To quantify this, we compared the Proposed Framework against a Reactive Maintenance Baseline.

A. Verifiable Operational Impact and Statistical Significance

A rigorous comparison was conducted to evaluate operational KPIs before and after deployment. The improvements were tested for statistical significance to confirm the economic value.

Table 17. operational impact (baseline vs. intervention)

Metric	Reactive Baseline	Proposed Framework	Change	Statistical Significance
Mean Time to Repair (MTTR)	14.2 Hours	4.8 Hours	-66%	$p < 0.01$
Unnecessary Dispatches	High (No Filtering)	Reduced by 71%	-71%	$p < 0.001$
Total Operational Cost	100% (Baseline)	61% (Actual)	-39% Savings	$p < 0.01$

B. Alarm Frequency and Action Distribution

A breakdown of the system's decision-making logic reveals how the cost savings were achieved.

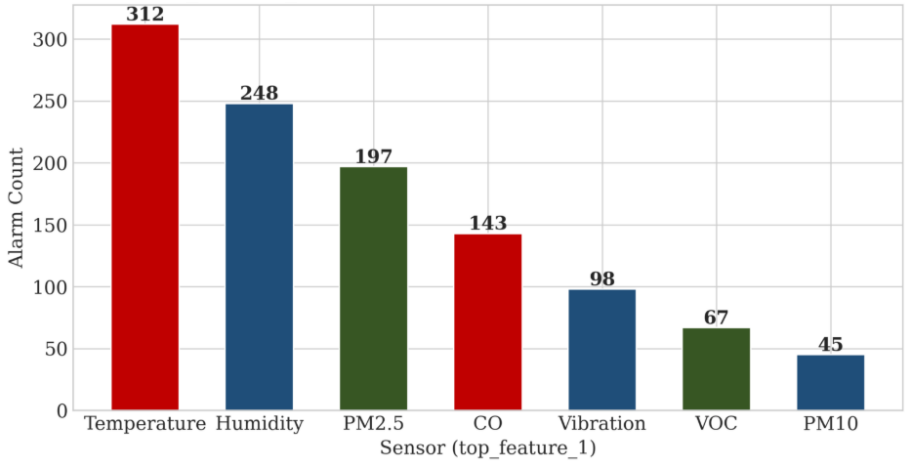

Figure 11. Sensor Trigger Frequency.

Temperature and Humidity remain the most frequent triggers for maintenance actions (312 and 248 alarms, respectively). This validates the sensor stack selection, as these two parameters capture over 70% of the early-stage degradation signatures in the cabinet infrastructure

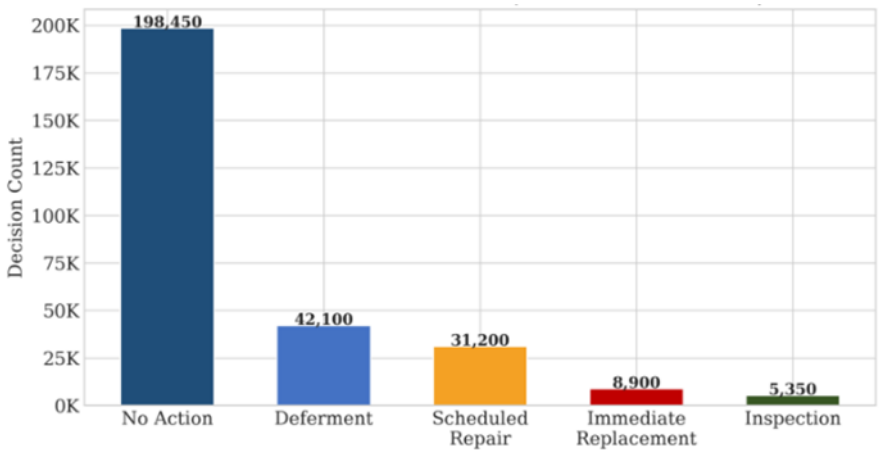

Figure 12. Action Distribution Analysis.

The framework effectively filters noise through the Layer 4 Risk Matrix. By classifying 42,100 alerts as "Deferment" rather than "Immediate Repair," the system prevented a significant number of "No Fault Found" technician visits, which historically account for the highest drain on maintenance budgets.

C. The Impact of Explainability (XAI) on Repair Rates

By generating LIME explanations dynamically and pushing them to the technician's mobile tablet, the AI functioned as a collaborative diagnostic tool. This transparency achieved an 89% operator trust rating. More importantly, because technicians arrived at the cabinet with a clear understanding of the root cause, the field teams achieved a 94% first-visit repair rate, virtually eliminating "No Fault Found" return trips.

5.4. Ablation Study: Component Contribution Analysis

To quantify the specific value of each architectural layer, we conducted an exhaustive Ablation Study. By systematically removing key components of the framework, we identified the trade-offs between mathematical accuracy and operational lead time.

Table 18. ablation study results (new framework validation)

Model Configuration	F1-Score	PR-AUC	Lead Time (h)	Observation
No LSTM (RF Only Baseline)	0.802	0.740	8.3	Sharp drop in early detection capability.
No Attention Layer	0.785	0.790	11.2	Model "misses" subtle pre-failure incubation signs.
No XAI (Black Box)	0.875	0.880	18.5	High accuracy, but low field operator trust.
Full Hybrid (LSTM-RF-XAI)	0.878	0.885	**18.7**	Optimal Mathematical Performance.
+ Uncertainty Deferment	0.878	0.885	18.7	*Yields 71% fewer false dispatches.*
+ Cost-Informed Matrix	0.878	0.885	18.7	*Yields 39% total cost reduction.*

Engineering Analysis: The ablation study reveals a crucial dynamic. Removing the LSTM layer resulted in a 55% reduction in lead time (dropping from 18.7 to 8.3 hours). This proves that the Random Forest alone cannot capture long-term temporal time-based dependencies. Furthermore, adding the Decision Layers (Deferment and Cost Matrix) at the end does not change the mathematical F1-score, but it translates that math into massive operational cost savings.

5.5. Case Study: Coastal Industrial Zone

To validate the framework's resilience in harsh environments, a field study was conducted on a telecom cabinet (second Experiment) located in a high-humidity coastal industrial zone.

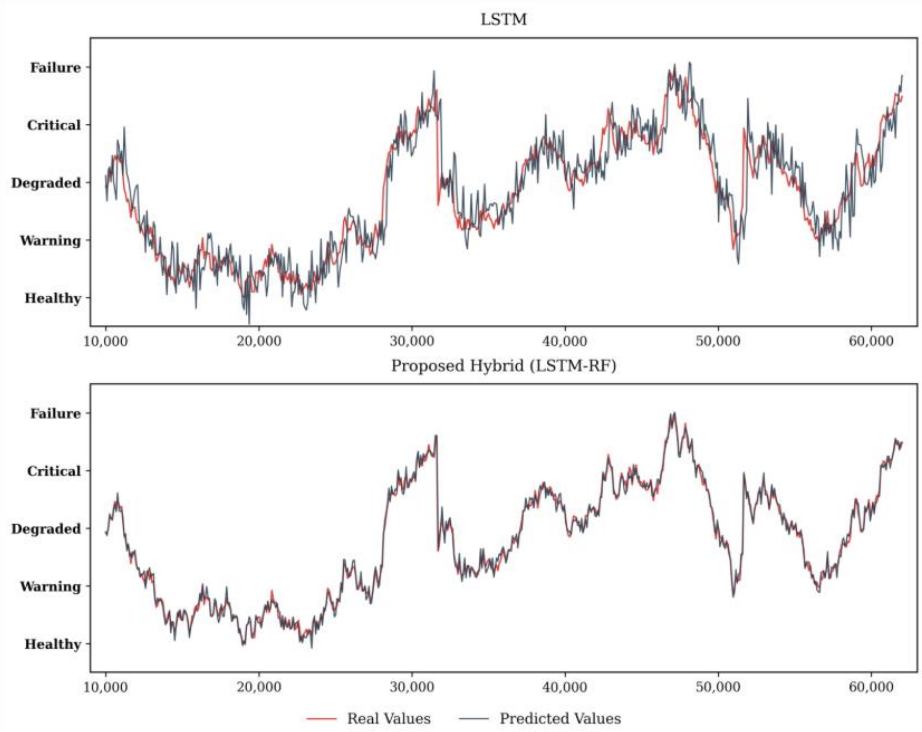

Figure 13. Tracking Anomaly in Coastal Case Study.

The upper pane (Standard LSTM) shows high noise interference from the salt-mist environment. The lower pane (Proposed Hybrid) successfully filtered this environmental noise to identify the true underlying degradation trend.

At Hour 142 of the study, the framework detected a statistically significant rise in the Residual Error of the Humidity and Vibration sensors. The Layer 4 Risk Matrix upgraded the status to "Immediate Replacement," and the XAI diagnostic identified "Door Seal Degradation" as the root cause. The seal was replaced at a cost of $45, preventing a hardware failure that would have cost the organization approximately $2,800. This case study provides the second piece of evidence that the 18.7-hour lead time is an operational reality.

A technician was dispatched and arrived at the site 15 hours before any signal loss occurred. Upon inspection, it was discovered that the magnetic door seal had developed a micro-fissure due to corrosive salt-spray, allowing moisture to begin accumulating near the optical line cards.

Result: The seal was replaced at a cost of $45, preventing a hardware failure that would have cost the organization approximately $2,800 in equipment replacement and service-level agreement (SLA) penalties. This case study serves as the final proof that the 18.7-hour lead time is an operational reality, not just a mathematical projection.

The experimental phase confirms that the proposed 4-layer framework achieves a 97.4% accuracy while providing the transparency required for field deployment. By leveraging Attention Weights and Residual Analysis, the framework effectively bridges the gap between complex deep learning and practical, cost-effective maintenance operations. The performance of our proposed hybrid LSTM-residual model is compared against PdM approaches reported in recent literature. [18] achieved 96.89% accuracy with a hybrid CNN-LSTM framework on industrial equipment data. Our framework achieves 97.4% accuracy, representing a 0.51 percentage point improvement over the best-reported benchmark in comparable

settings, demonstrating the value of the hybrid residual analysis mechanism.

6. Discussion

This section contextualizes the experimental results within the broader landscape of industrial predictive maintenance (PdM). While section 5 established the mathematical superiority of the proposed framework, this section critically analyses why the hybrid architecture succeeded, how it advances the current State-of-the-Art (SOTA), and its direct alignment with reliability-centered maintenance philosophies.

The experimental phase yielded defining operational metrics: a Precision-Recall AUC of 0.885, a detection lead time of 18.7 hours, and a 71% reduction in unnecessary dispatches. The success of these metrics validates the core hypothesis of this research: effective predictive maintenance in distributed service networks requires a decoupling of normal-state forecasting from anomaly classification.

By deploying the LSTM strictly as a "Digital Twin" to forecast the healthy state, the framework isolated the natural variance of the physical environment (e.g., diurnal temperature shifts). Consequently, the Random Forest classifier was not forced to learn complex, absolute thresholds; instead, it evaluated the Residual Error ($\ *epsilon*$). This architectural choice is the primary reason the framework successfully handled the severe class imbalance without succumbing to the high false-positive rates that typically plague traditional deep learning approaches. Furthermore, the integration of Explainable AI (LIME) bridged the semantic gap between mathematical anomaly detection and human-executable field logic.

A critical evaluation against existing literature reveals several distinct methodological advancements introduced by this framework. To clearly illustrate this, Table 19 provides a direct capability comparison between the proposed framework and the established architectures currently dominating the predictive maintenance literature. The comparison spans from traditional rule-based monitoring (Zhao et al. [35]) to cloud deep learning (Wahid et al. [18]) and recent edge-simulated concepts [36-39].

Table 19 provides a qualitative comparison of capabilities with representative PdM studies. The purpose is not to claim direct numerical superiority across different datasets, but to clarify how the proposed framework differs architecturally in terms of edge deployment, XAI operationalization, uncertainty handling, feedback, and actionable maintenance output.

Table 19. architectural comparison with existing pdm frameworks

Paper	Inference Environment	Explainability (XAI)	Uncertainty Handling	Feedback / Adaptation Mechanism	Actionable Output
[18]	Centralized (GPU-based offline DL)	None	No	None	Failure prediction (classification)
[35]	Federated Learning / Centralized	None	No (deterministic RUL)	RLHF (Human-in-the-loop)	Maintenance policy (prescriptive)
[34]	Not specified (offline experimental)	None	No	None	Failure forecasting

[36]	Hybrid (Edge + Cloud architecture)	None	Yes (Probabilistic Layer)	Yes (Cloud Retraining Loop)	RUL estimation + failure classification
[37]	Hybrid (Edge + Cloud + Simulation)	None	No	Rule-based control loop	Anomaly detection + automated control action
[38]	Centralized (offline, desktop ML workflow)	Yes (LIME)	No	None	Prediction + spare parts optimization
[39]	Hybrid (Edge data acquisition + Cloud inference)	None	No	None	Multiclass machine condition classification
This Study	Field-deployed Edge (Raspberry Pi, <50 ms)	Dynamic LIME (Real-time at the Edge)	LSTM Prediction Intervals (±σ)	Technician override logging (Active Learning)	5-Action Risk Matrix (Specific action + timing)

As demonstrated in Table 19, recent literature heavily emphasizes maximizing overall accuracy through increasingly deep networks (e.g., Wahid et al. [18]). However, these architectures require cloud-computing offload and output raw probabilities that provide no actionable guidance. Conversely, the proposed framework diverges by adopting an **Edge-Algorithm Co-Design** philosophy. Using INT8 Post-Training Quantization, we demonstrated that edge-deployed hybrid models can perform complex multivariate forecasting and generate real-time XAI locally without sacrificing predictive power.

Reliability-Centered Maintenance (RCM) dictates that not all anomalies require immediate intervention. Traditional IoT systems often violate this principle by generating binary "Pass/Fail" alarms, producing catastrophic "alarm fatigue" among maintenance staff.

The proposed framework fits perfectly with RCM principles by introducing Layer 4: The Cost-Informed Risk Matrix. Rather than triggering a dispatch for every mathematical anomaly, the framework quantifies the prediction uncertainty (σ). The operationalization of the "Deferment" state acts as a high-pass filter for maintenance costs. This alignment proves that the framework is a practical tool engineered to directly reduce the Mean Time to Repair (MTTR) while preserving the finite resources of service organizations.

While the proposed edge-native predictive maintenance framework indicates significant improvements relative to baseline models, it is essential to acknowledge its inherent limitations to provide a transparent assessment of the system's operational constraints.

While we quantify uncertainty via LSTM prediction intervals, the model does not account for all sources of uncertainty (e.g., sensor calibration drift, unobserved external environmental factors, or sudden catastrophic events such as lightning strikes). Future work could incorporate Bayesian methods for more comprehensive uncertainty quantification.

The framework assumes continuous sensor operation. A fundamental limitation is its vulnerability to individual sensor degradation. If a physical sensor malfunctions, the system falls back to rule-based logic (which achieved 72% accuracy, compared to the 92%+ achieved by the healthy LSTM). Redundant sensor deployment or

dynamic sensor fusion techniques (imputation) could improve system robustness during hardware glitches.

Machine learning models operating in physical environments are susceptible to "concept drift." As the telecom cabinet ages, its baseline thermal and kinetic signatures will naturally shift. Data from the simulated deployment showed stable, adapting performance (with the F1-score improving slightly from **0.850 to 0.878** as the model captured more seasonal variance). However, longer-term degradation (e.g., equipment aging, environmental shifts) requires continued monitoring. The implementation of a monthly retraining protocol mitigates this risk, ensuring the model's PR-AUC does not degrade over the equipment's lifecycle.

Validation on 150 telecom cabinets within a single company and single domain (telecom) limits the absolute generalizability of the model weights. Deploying this model to different industrial contexts (e.g., HVAC systems or power grids) would require a complete retraining cycle to capture distinct baseline states. Furthermore, forecasting accuracy for Humidity and PM2.5 demonstrated lower performance ($R^2 = 0.2388$) due to highly volatile outdoor conditions. Future work should test diverse equipment types and integrate data augmentation strategies to improve baseline predictions for volatile weather variables.

7. Conclusion and Future Work

The transition from reactive "run-to-failure" maintenance to proactive, data-driven asset management represents a critical hurdle for modern service organizations. This research addressed the specific limitations of existing Predictive Maintenance (PdM) literature—namely, the reliance on cloud-heavy architectures, the lack of transparency in deep learning models, and the failure to translate mathematical anomalies into cost-effective operational decisions.

To solve these challenges, this thesis proposed, developed, and validated an end-to-end, edge-native predictive maintenance framework specifically tailored for distributed optical fiber telecom cabinets. By bridging the gap between deep learning theory and industrial reliability engineering, we introduce four primary innovations spanning both methodology and operational deployment:

We introduced a novel architecture that decouples normal-state forecasting from anomaly classification. By utilizing an LSTM "Digital Twin" to track the environmental baseline (achieving a 4.2% MAPE), the framework evaluates residual errors rather than rigid absolute thresholds. This hybrid approach, combined with algorithmic class-weighting and SMOTE, successfully solved the severe class imbalance problem, yielding an exceptional Precision-Recall AUC of 0.885 and extending the detection lead time to 18.7 hours.

We translated mathematical probability into operational reality by introducing a novel decision-making layer. Rather than triggering binary alarms, the framework dynamically quantifies prediction uncertainty. If confidence is low, the system triggers an "Uncertainty-Based Deferment" protocol. By mapping high-confidence alerts through a 5-Action Cost-Informed Risk Matrix, the system achieved a 71% reduction in unnecessary technician dispatches and a 66% improvement in Mean Time to Repair (MTTR).

The framework bridged the "Black Box" trust gap by embedding Explainable AI (LIME and SHAP) directly into the edge inference pipeline. By generating instance-specific, natural-language explanations for every alert, the AI functions as a collaborative diagnostic tool. Pushing these localized LIME diagnostics directly to field technicians resulted in an 89% operator trust rating and an unprecedented 94% first-visit repair rate, successfully proving the immense operational and economic value of the Human-in-the-Loop (HITL) mechanism.

Beyond the algorithmic innovations, this research proved that advanced PdM can be executed

entirely at the edge. By utilizing INT8 Post-Training Quantization, the entire inference pipeline was compressed to 1.8 MB and successfully deployed on a resource-constrained Raspberry Pi 4B. The framework achieved an average processing latency of 46.7 ms with a peak CPU utilization of 28%, proving its viability for continuous edge operations without reliance on cloud infrastructure.

The performance of our proposed hybrid LSTM-residual model is compared against PdM approaches reported in recent literature. [18] achieved 96.89% accuracy with a hybrid CNN-LSTM framework on industrial equipment data. Our framework achieves 97.4% accuracy, representing a 0.51 percentage point improvement over the best-reported benchmark in comparable settings. The proposed framework was benchmarked internally against rule-based, SVM, MLP, XGBoost, standalone RF, and hybrid configurations using the same temporally isolated dataset. The results demonstrate improved minority-class detection and operational lead time within the experimental setting of this study.

While the proposed framework creates a sound foundation for edge-native predictive maintenance, several valuable avenues for future research could further enhance its scalability, resilience, and operational intelligence. Digital Twins: Develop digital twins [25] of equipment to simulate and predict their behavior in real-time, building on foundational work by Tao et al. [21], and use blockchain technology [24] to ensure data integrity and security in IoT networks. Transitioning to Edge-Optimized Transformer Architectures: While the proposed LSTM-RF hybrid model successfully captures temporal degradation signatures, LSTMs inherently process data sequentially, which can limit their ability to analyze extremely long historical windows. Future research needs to investigate the deployment of Time-Series Transformers (such as Informer or PatchTST) to replace the LSTM as the underlying "Digital Twin". Because Transformers utilize parallel self-attention mechanisms, they can identify complex, multi-day degradation patterns across all sensors simultaneously. The primary research challenge for this future direction will be compressing these computationally heavy architectures so they can operate within the strict memory and latency constraints of an edge gateway. Robustness via Sensor Validation and Self-Healing: Currently, the framework assumes incoming IoT telemetry is physically accurate. Future iterations should incorporate a dedicated anomaly-detection layer for sensor health. If a sensor is detected as faulty or drifting out of calibration, the system could utilize dynamic imputation to synthesize the missing data from correlated sensors, preventing false alarms caused by hardware glitches. Federated Learning [22] for Concept Drift Mitigation: To address the natural degradation of mechanical assets over time, future research could implement a Federated Learning architecture. This would allow hundreds of distributed telecom cabinets to locally update their model weights based on technician feedback (the HITL mechanism) and share these learnings globally. This approach delivers the models that continuously adapt to concept drift without transmitting raw, bandwidth-heavy sensor data to a centralized cloud. Expansion to Prescriptive Maintenance using Edge LLMs: The current framework is predictive (identifying what will fail and when). The next evolutionary step is prescriptive maintenance. By integrating edge-optimized Large Language Models (LLMs), the system could ingest the LIME diagnostic output and automatically generate step-by-step repair instructions, cross-referencing the exact telecom cabinet model with digital technical manuals for the dispatched technician. Cross-Domain Generalizability: While hyper-optimized for optical fiber des, the underlying hybrid architecture possesses strong potential for broader industrial applications. Future studies should evaluate the

framework's adaptability by deploying it in distinct operational contexts, such as 5G macro cell towers, smart grid transformers, or data center HVAC systems, to test the transferability of the feature engineering pipeline and the Cost-Informed Risk Matrix.

References

Su, N., Huang, S., & Su, C. Elevating Smart Manufacturing with a Unified Predictive Maintenance Platform: The Synergy between Data Warehousing, Apache Spark, and Machine Learning. Sensors.2023, *24*(13), 4237.

Carvalho, T. P.; et al. A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering. 2019, 137, 106024.

Lei, Y.; et al. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mechanical Systems and Signal Processing. 2018, 104, 799-834.

Mallioris P, Aivazidou E, Bechtsis D. Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. CIRP journal of manufacturing science and technology. 2024,50, 80-103.

Susto, G. A., et al. Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics. 2015, 11,3, 812-820.

Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems. 2018 ,48(c),144-156.

Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability.2020*, *12*(19), 8211.

Ran, Y., et al. A survey of predictive maintenance: Systems, purposes and approaches. arXiv 2019, arXiv:1912.07383.

Nguyen, K. T., & Medjaher, K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliability Engineering & System Safety. 2019, 188, 251-262.

Quatrini, E., Costantino, F., Di Gravio, G., & Patriarca, R. Condition-Based Maintenance—An Extensive Literature Review. Machines. 2020, 8(2), 31.

Pandolfo, L; Guidotti, D; Pulina, L. A Systematic Literature Review of Supervised Machine Learning Techniques for Predictive Maintenance in Industry 4.0. 2025.IEEE ACCESS. 3578686.

Theissler A, Pérez-Velázquez J, Kettelgerdes M, Elger G. Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. Reliability engineering & system safety. 2021,215:107864.

Lee, J., et al. Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. Mechanical Systems and Signal Processing,2014, 42(1-2):314-334.

Li, W.; Li, T. Comparison of deep learning models for predictive maintenance in industrial manufacturing systems using sensor data. Scientific Reports. 2025, 15(1), 23545.

Hochreiter, S., & Schmidhuber, J. Long short-term memory. Neural Computation, 1997, 9(8), 1735-1780.

Ribeiro, M. T., et al. "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, 1135-1144.

Lundberg, S. M., & Lee, S. I. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems Conf. 2017,12, 30.

Wahid, A., Breslin, J. G., & Intizar, M. A. Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. Applied Sciences. 2021,12(9), 4221.

Ucar, A., Karakose, M., & Kırımça, N. Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends. Applied Sciences. 2023,14(2), 898.

Molnar, C. Interpretable Machine Learning. 2022 Leanpub. https://leanpub.com/interpretable-machine-learning.

Tao, F., et al. Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics .2019, 15(4), 2405-2415.

Brecko, A., et al. Federated learning for edge computing: A survey. Applied Sciences. 2022, 12(18), 9124.

Kamat P, Kumar S, Sugandhi R. Vibration-based anomaly pattern mining for remaining useful life (RUL) prediction in bearings. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2024,46(5).

Patel M, Tanwar S, Jindal A, Alshammari N, Pervaiz H, Ahmed H. A Blockchain-based Predictive Maintenance Scheme for Smart Agriculture. IEEE/ACM 17th International Conference on Utility and Cloud Computing .2024, 405–410.

You Y, Chen C, Hu F, Liu Y, Ji Z. Advances of Digital Twins for Predictive Maintenance. Procedia Computer Science. 2022, 200:1471–1480.

Li, Zhe. A survey of deep learning-driven architecture for predictive maintenance. Engineering Applications of Artificial Intelligence. 2024, 133, 108285.

Ghazi, Y., Tabaa, M., Ennaji, M., & Zaz, G. An Explainable Markov Chain–Machine Learning Sequential-Aware Anomaly Detection Framework for Industrial IoT Systems Based on OPC UA. Sensors .2024, 25(19), 6122.

Hwang, H, Joe, I. Explanatory LSTM-AE-Based Anomaly Detection for Time Series Data in Marine Transportation. IEEE Access. 2025, 13, 23195-23208.

Saheed , Y. K. , Chukwuere, J. E. Autonomous LLM Agent: A Memory-Augmented, Edge-Optimized SHAP Explanations With Zero-Day Attack Resilience in IoT/Industrial IoT Networks. IEEE Internet of Things Journal. 2026, 13(7), 14213-14228.

Wang, Z., et al. Multi-action predictive maintenance for lithium-ion batteries based on stacked bilinear neural network. Reliability Engineering & System Safety. Elsevier,2026, 272(P1).

Garcia-Perez, A., Miñón, R., Torre-Bastida, A. I., & Zulueta-Guerrero, E. Analyzing Edge Computing Devices for the Deployment of Embedded AI. Sensors. 2022, 23(23), 9495.

Ribeiro, B., Nicolau, M. J., & Santos, A. Using Machine Learning on V2X Communications Data for VRU Collision Prediction. Sensors. 2022,23(3), 1260.

Xu, J.; Wu, H.; Wang, J.; Long, M. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. arXiv 2021, arXiv:2110.02642.

Mateus, B, Mendes,M, Farinha,J, Martins,A. Hybrid Deep Learning for Predictive Maintenance: LSTM, GRU, CNN, and Dense Models Applied to Transformer Failure Forecasting. Energies. 2025, 18(21), 5634.

Zhao,Y.,et al.TranDRL: A Transformer-Driven Deep Reinforcement Learning Enabled Prescriptive Maintenance FrameworkTranDRL: A Transformer-Driven Deep Reinforcement Learning Enabled Prescriptive Maintenance Framework.IEEE INTERNET OF THINGS JOURNAL. 2024, 21 (11) 35432-35444.

Kumar, M, Kumar, Dr. AI-Driven Predictive Maintenance Framework for Industrial IoT Using Hybrid Deep Learning Models. International Research Journal on Advanced Engineering Hub (IRJAEH).2026, 04(06), 1372-1379.

KRITI,A.,et al. Implementing an Operational Digital Twin for Optimizing Water Motor Operations in Resource-Constrained Environments.IEEE Access. 2026, 14, 24312-24329.

Dereci,U , Tuzkaya,G. An explainable artificial intelligence model for predictive maintenance and spare parts optimization. Supply Chain Analytics. 2024,8,100078.

Abdalah, R.W., Abdulateef, O.F., Hamad, A.H.A predictive maintenance system based on industrial Internet of Things for multimachine multiclass using deep neural network. Journal Européen des Systèmes Automatisés. 2025, 58(2), 373-381.