2026 Volume: 02 Issue: 11 www.ijrpa.com ISSN 2456-9995 Research Article

International Journal Research Publication Analysis

Page: 01-20

# AGENTIC AI-DRIVEN PREDICTIVE MAINTENANCE FOR ELECTRIC VEHICLES USING IOT SENSOR DATA

**Jeevitha M.[1], Rachana M. K.[1], Kishan G.[1], Raghu G.[1], and Geethapriya G. H.*[2]**

*[1]Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.*

*[2]Assistant Professor, Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.*

**Article Received: 01 October 2026**
**Article Revised: 20 October 2026**
**Published on: 10 November 2026**

***Corresponding Author: Geethapriya G. H.**
Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, India.

DOI: https://doi-doi.org/101555/ijrpa.2838

## ABSTRACT

Electric-vehicle (EV) maintenance systems commonly combine threshold-based alerts with cloud telemetry, but these mechanisms may not account for vehicle-specific operating limits or coordinate a response after a fault is detected. This paper presents VoltPulse AI, a prototype predictive-maintenance platform that links an ESP32-based sensing testbench to cloud telemetry, vehicle-specific calibration profiles, a Per-Unit (p.u.) feature transformation, and a hybrid diagnostic service. The service combines a multi-class Extreme Gradient Boosting (XGBoost) model with deterministic boundary rules, then sends the resulting diagnostic state to asynchronous agents for driver advisories, email notification, and service-bay allocation. A React-based cockpit displays live telemetry and supports controlled fault-injection demonstrations. The project reports a five-class dataset of 12,500 frames, an evaluated held-out classification accuracy of 98.80%, a model inference latency of 3.82 ms, and an end-to-end pipeline latency of 164.50 ms. The paper documents the full architectural design, mathematical normalization, hybrid arbitration logic, empirical baseline comparisons, reproducibility audits, and deployment boundaries.

**INDEX TERMS:** *Electric vehicles, predictive maintenance, Internet of Things, agentic systems, Per-Unit normalization, XGBoost, functional safety, digital twin.*

## I. INTRODUCTION

Electric vehicles depend on coupled electrical, thermal, and mechanical subsystems. Changes in battery temperature, DC-bus voltage, motor current, shaft speed, or vibration can indicate abnormal operating conditions. Maintenance based on fixed limits can be useful for clearly defined physical boundaries, but limits configured for one vehicle or testbench do not necessarily transfer to another without corresponding calibration. A telemetry platform also needs to translate a diagnostic result into a comprehensible and appropriately controlled response.

VoltPulse AI is a prototype intended to connect these functions. An ESP32-based testbench collects electro-thermal and kinematic measurements; a cloud service relays telemetry and associates a vehicle identifier with an operating profile; a backend transforms sensor values into dimensionless features and performs classification; and a web cockpit presents the result. The reported design adds an agent workflow for voice guidance, rate-limited email, service-bay booking, and recovery-state logging.

The project makes four primary contributions:

- It describes an edge-to-dashboard telemetry pipeline using conditioned sensor inputs and an ESP32 microcontroller.
- It defines Per-Unit (p.u.) stress features intended to represent operating severity relative to a vehicle calibration profile.
- It combines a tree-ensemble classifier with deterministic threshold rules in a hybrid diagnostic pipeline.
- It describes an asynchronous remediation workflow and reports prototype classification and latency results across scripted test scenarios.

The reported measurements should be interpreted in the context of the supplied experimental description. The described testbench is a scaled laboratory rig with fault-injection scenarios, not a commercial high-voltage EV fleet. To provide full scientific transparency, this paper documents the reported prototype findings, reconciles technical discrepancies identified during peer audit, and outlines the verification evidence required for automotive production deployment. The system is a research prototype and is not presented as ISO 26262-certified or as a substitute for a production battery-management system (BMS).

## II. RELATED WORK AND RESEARCH GAP

### A. Learning-Based Diagnostics

Ensemble methods are widely used for structured classification problems. Random forests aggregate decision trees through bootstrap aggregation and feature subspace sampling [2], while gradient-boosted trees build additive ensembles and support regularized objectives. XGBoost is a scalable implementation of gradient-boosted decision trees and is used here as the project's tabular telemetry classifier [1]. These references motivate the model family; they do not independently establish the performance reported for VoltPulse AI.

### B. IoT and Electric-Drive Diagnostics

Zheng et al. frame intelligent maintenance as a combination of IIoT sensing, data infrastructure, learned reliability models, and maintenance decision support, demonstrating their method using turbofan degradation data rather than an EV fleet [3]. For rotating machinery, Mohammed et al. describe an IoT/cloud motor-monitoring system using vibration, current, and temperature measurements with MQTT and multiple ML baselines [4]. Kolok et al. present a low-cost ESP32 vibration/acoustic testbench with RMS and FFT processing for small-motor conditions including imbalance and wear [5]. Kavana and Neethi report induction-motor fault classification from three-phase voltage/current signals and a real motor test [6]. These studies motivate the sensing and motor-diagnostic components in VoltPulse AI, but their hardware, labels, and evaluation data are not directly interchangeable with the present project's telemetry classes.

EV-specific diagnostic studies likewise address distinct subsystems and data regimes. Sun et al. propose online battery fault diagnosis and thermal-runaway early warning using voltage and temperature data from vehicles in operation [7]. Zhang et al. study battery-voltage faults while accounting for driving-condition variation in real-world electric-van data [8]. For electric-bus auxiliaries, Guo et al. combine peer-system comparisons, operating-condition models, and domain rules to diagnose bus air-conditioning faults in a fleet [9]. Ercevik and Ozbayoglu investigate AI-based alarm prediction from electric-bus CAN data [10]. These works establish relevant EV applications, while also showing that fault targets, sensor availability, and validation scope vary considerably. VoltPulse AI's stated gap is a prototype integration of profile-scaled telemetry classification, explicit threshold arbitration, and application-level service actions; it does not claim to reproduce or outperform these subsystem-specific methods.

### C. Battery Prognostics and Diagnostic Models

Battery prognostics research includes methods for estimating degradation and useful life from cycling data. Prior work has examined battery lifetime prediction and data-driven cycle-life estimation [11], [12]. The present prototype addresses a different task: classifying operational fault categories from telemetry features. It does not claim to estimate battery state of health (SoH) or remaining useful life (RUL).

For tabular sensor inputs, tree ensembles offer a relevant comparison family. Random forests aggregate decision trees [2], while XGBoost formulates regularized gradient-boosted trees for scalable classification [1]. The supplied project reports comparisons with logistic regression, an RBF SVM, random forest, an MLP, and XGBoost, providing baseline benchmarks across computational scales.

### D. Digital Twins and Safety Context

Digital twins connect information about a physical asset with a computational representation, but the term covers systems with differing levels of model fidelity and synchronization [13]. In this work, 'digital twin' refers to the platform's VIN-associated profile and live telemetry representation; no high-fidelity electrochemical or drivetrain simulation is claimed.

ISO 26262 provides a functional-safety framework for road vehicles [14]. The rule-based guards in VoltPulse AI are described as deterministic safety-oriented checks inspired by the need for explicit boundary handling. Their presence does not demonstrate formal compliance with ISO 26262, assign an externally certified ASIL, or establish a complete safety case. Such claims require a formal hazard analysis, traceable requirements, verification evidence, and an independent assessment beyond the supplied project results.

### E. Agent-Oriented Maintenance Workflows

Multi-agent systems provide a conceptual framework for decomposing a workflow into cooperating software components [15]. In VoltPulse AI, agent roles are represented by asynchronous routines for diagnosis, notification, and service coordination. These roles do not by themselves establish autonomous reasoning or a safety-certified agent; the scope here is event-triggered orchestration.

### F. Research Gap and Project Positioning

The cited literature provides context for tree ensembles, battery prognostics, digital twins, and vehicle functional safety, but does not establish a prior system that is directly comparable across all of these functions. VoltPulse AI therefore targets an integration problem: linking vehicle-profile-scaled telemetry classification with guarded fault-state transitions and

application-level remediation. Table I summarizes the positioning of the project within related technical approaches.

**TABLE I: POSITIONING OF THE PROJECT WITHIN RELATED TECHNICAL APPROACHES.**

| Approach | Typical Focus in Cited Background | VoltPulse AI Project Scope |
|---|---|---|
| Battery prognostics | Degradation and cycle-life estimation [11], [12] | Discrete telemetry fault classification; continuous SoH/RUL estimation remains future work. |
| Tree-ensemble classification | Structured feature learning and class prediction [1], [2] | XGBoost operates on project-defined per-unit features with reported baseline comparisons. |
| Digital-twin systems | Linking an asset representation with operational data [13] | VIN-associated calibration profile and live telemetry view; no electrochemical simulator is claimed. |
| Agent-based workflows | Decomposing work across cooperating software agents [15] | Event-driven notification, email, service-bay allocation, and recovery logging. |

## III. PROBLEM FORMULATION

For each vehicle identifier k, the diagnostic task is to map a telemetry frame and its calibration profile to an operational class:

$$c_k(t) \in \{0, 1, 2, 3, 4\} \quad (1)$$

Representing nominal operation (C0), motor overload (C1), DC-bus voltage fault (C2), thermal hazard (C3), or drivetrain vibration imbalance (C4). The service also produces a confidence value and compares the current state with the previous state to decide whether a new fault, escalation, or recovery event has occurred. The intended design goals are to express measurements relative to vehicle-specific limits, classify the reported fault categories with low computational latency, and avoid repeated outreach during a persistent state. This is telemetry classification and fault response; it is not a demonstrated estimate of remaining useful life or a validated forecast of failure before symptoms appear. Table II details the operational classes, assigned trouble codes, and physical precursors.

**TABLE II: OPERATIONAL CLASSES IN THE REPORTED DIAGNOSTIC TASK.**

| Class | Project DTC | Reported Precursor |
|---|---|---|
| 0 | DTC-SYS-000 | Nominal operating region. |
| 1 | DTC-MECH-101 | Motor overload or stall interaction. |
| 2 | DTC-ELEC-202 | DC-bus voltage sag or surge. |
| 3 | DTC-THERM-303 | Elevated core temperature and internal rise. |
| 4 | DTC-VIB-404 | Dynamic vibration imbalance. |

## IV. PROPOSED SYSTEM ARCHITECTURE

### A. Architecture Overview

The system is organized into four connected stages:

- **Edge Stage:** The ESP32-WROOM-32 collects voltage, current, speed, temperature, acceleration, and environmental readings. Analog oversampling, deadband filtering, and gravity compensation occur prior to serialization.
- **Cloud Stage:** Firebase Realtime Database (RTDB) manages the live telemetry path indexed by VIN. Cloud Firestore stores user bindings, OEM calibration profiles, telemetry history, and fault-transition records.
- **Diagnostic Stage:** An asynchronous FastAPI service associates telemetry with calibration profiles, computes dimensionless features, invokes a pre-trained XGBoost model, and evaluates deterministic safety rules.
- **Agent and Cockpit Stage:** The orchestrator tracks per-vehicle diagnostic state to avoid duplicate alerts and confirm recovery over consecutive frames. Downstream actions include voice advisories, email dispatch, service-bay scheduling, audit logging, and WebSocket dashboard streaming.

### *B. Telemetry and Calibration Vector*

At time t, the supplied feature description starts from the nine-element observation vector:

$$x_{raw}(t) = [V_{dc}, I_{motor}, \omega_{rpm}, T_{core}, a_x, a_y, a_z, T_{amb}, H_{rel}]^T \quad (2)$$

where voltage is measured in volts, current in amperes, rotational velocity in revolutions per minute, temperatures in degrees Celsius, acceleration in meters per second squared, and relative humidity in percent. Each vehicle profile $P_k$ is described by baseline and boundary limits:

$P_k = \{V_{nom,k}, V_{min,k}, V_{max,k}, I_{rated,k}, \omega_{rated,k}, T_{nom,k}, T_{crit,k}, a_{crit,k}, \Delta T_{max,k}\}$ (3)

Table III gives reported example profiles for a 5-V rail, a 3-cell lithium-ion rig, and a nominal 100-V traction profile. Table IV outlines the physical interpretation and calibration dependencies for each feature.

**TABLE III: OEM CALIBRATION PROFILES REPORTED FOR THREE HARDWARE CLASSES.**

| Profile Parameter | 5-V Rail | 3S Li-ion Rig | 100-V Traction |
|---|---|---|---|
| V_nom (V) | 4.35 | 11.80 | 96.00 |
| V_min (V) | 3.10 | 9.80 | 78.00 |
| V_max (V) | 5.25 | 12.60 | 108.00 |
| I_rated (A) | 2.00 | 0.85 | 45.00 |
| ω_rated (RPM) | 225 | 250 | 3000 |
| T_nom (°C) | 28.0 | 28.0 | 30.0 |
| T_crit (°C) | 55.0 | 38.0 | 60.0 |
| ΔT_max (°C) | 25.0 | 15.0 | 30.0 |
| a_crit (m/s²) | 2.20 | 2.20 | 4.50 |

**TABLE IV: INTERPRETATION AND CALIBRATION DEPENDENCIES OF THE REPORTED FEATURE VECTOR.**

| Feature(s) | Source Quantity | Intended Interpretation | Calibration or Validation Note |
|---|---|---|---|
| V_p.u., S_V | DC-bus voltage | Nominal voltage ratio and distance from lower/upper limits | Depends on profile selection and verified circuit attenuation range. |
| S_I, ω_p.u. | Motor current and speed | Loading and speed relative to rated values | Current polarity, stall envelope, and rated limits must match tested motor. |
| Φ_stall | Current and speed interaction | High current when speed is depressed | Thresholds should be evaluated across startup and dynamic transient cases. |

| Feature(s) | Source Quantity | Intended Interpretation | Calibration or Validation Note |
|---|---|---|---|
| S_T,abs, S_T,rise | Core and ambient temperature | Absolute thermal margin and core-to-ambient elevation | Parameterized by ΔT_max,k = T_crit,k - T_nom,k and thermal time constants. |
| S_vib | Three-axis acceleration | Dynamic acceleration relative to critical limit | Norm-minus-gravity baseline requires orientation alignment in dynamic rigs. |
| H_p.u. | Relative humidity | Dimensionless relative humidity representation | Standardized as H_rel / 100.0 over standard environmental ranges. |

### *C. Per-Unit Feature Transformation*

The voltage feature uses a nominal ratio and a direction-dependent stress measure:

$V_{p.u.}(t) = V_{dc}(t) / V_{nom,k}$ (4)

$S_V(t) = (V_{nom,k} - V_{dc}(t))/(V_{nom,k} - V_{min,k})$ if $V_{dc}(t) < V_{nom,k}$; else $(V_{dc}(t) - V_{nom,k})/(V_{max,k} - V_{nom,k})$ (5)

Nominal voltage maps to zero stress (S_V = 0), and the lower or upper boundary maps to unity (S_V = 1). Values beyond the boundary retain proportional magnitude without clipping. Current and rotational velocity ratios are defined as:

$S_I(t) = I_{motor}(t) / I_{rated,k}$, $\omega_{p.u.}(t) = \omega_{rpm}(t) / \omega_{rated,k}$ (6)

The interaction feature for high current during low rotational speed is:

$\Phi_{stall}(t) = S_I(t) \cdot \max(0, 1 - \omega_{p.u.}(t))$ (7)

Thermal features quantify absolute elevation over baseline and core rise over ambient:

$S_{T,abs}(t) = \max(0, (T_{core}(t) - T_{nom,k})/(T_{crit,k} - T_{nom,k}))$ (8)

$S_{T,rise}(t) = \max(0, (T_{core}(t) - T_{amb}(t))/\Delta T_{max,k})$ (9)

where $\Delta T_{max,k} = T_{crit,k} - T_{nom,k}$. The vibration feature is defined by removing gravity from the magnitude of tri-axial acceleration:

$a_{dyn}(t) = \max(0, \sqrt{(a_x^2 + a_y^2 + a_z^2)} - g_0)$, $S_{vib}(t) = a_{dyn}(t) / a_{crit,k}$ (10, 11)

with $g_0 = 9.81$ m/s². Humidity is scaled to $H_{p.u.}(t) = H_{rel}(t) / 100.0$. The final nine-component feature vector is:

$z(t) = [V_{p.u.}, S_V, S_I, \omega_{p.u.}, \Phi_{stall}, S_{T,abs}, S_{T,rise}, S_{vib}, H_{p.u.}]^T$ (12)

### *D. Classifier and Rule-Based Arbitration*

For input z_i, the tree ensemble forms class scores from additive trees and converts them to probabilities with a softmax:

$\hat{y}_{i,c} = \sum_{m=1}^{K} f_{m,c}(z_i), \quad p_{i,c} = \exp(\hat{y}_{i,c}) / \sum_{r=0}^{C-1} \exp(\hat{y}_{i,r})$ (13)

XGBoost optimizes an objective with second-order Taylor expansion:

$L^{(t)} \approx \sum_{i=1}^{N} [g_i f_t(z_i) + 0.5 h_i f_t^2(z_i)] + \gamma J + 0.5 \lambda \sum_{j=1}^{J} w_j^2$ (14)

where g_i and h_i are first and second derivatives of the loss, J is the number of leaves, w_j is leaf weight, and γ, λ penalize complexity. The optimal weight for leaf j is:

$w_j^* = - \sum_{i \in I_j} g_i / (\sum_{i \in I_j} h_i + \lambda)$ (15)

Table V provides the reported model configuration.

**TABLE V: REPORTED XGBOOST CONFIGURATION.**

| Parameter | Reported Value |
|---|---|
| Objective | multi:softprob |
| Estimators | 250 (early stop at 218) |
| Maximum depth | 6 |
| Learning rate | 0.05 |
| Subsample / column sample | 0.85 / 0.85 |
| γ / λ | 0.15 / 1.50 |
| Evaluation metrics | mlogloss, merror |
| Serialized model size | 4.20 MB |

The deterministic rule function is ordered by descending severity: thermal stress maps to Class 3, voltage stress to Class 2, current/stall stress to Class 1, and vibration to Class 4; otherwise it returns nominal (Class 0). Table VI details the rule conditions. Let c_ML be the classifier prediction and c_G the guard rule output. The robust arbitration logic is:

$c^*(t) = c_G \quad \text{if } c_G \neq 0; \quad \text{else } c_{ML}$ (16)

This formulation ensures that critical safety limit breaches immediately enforce the corresponding diagnostic state regardless of statistical classifier variance.

**TABLE VI: REPORTED DETERMINISTIC BOUNDARY RULES AND SEVERITY LABELS.**

| Priority | Condition | Class | Project Severity Label |
|---|---|---|---|
| 1 | $S_T,abs \geq 1.05$ and $S_T,rise \geq 0.65$ | 3, Thermal hazard | ASIL-D label |
| 2 | $S_V \geq 1.00$ | 2, Voltage fault | ASIL-C label |
| 3 | $S_I \geq 1.00$ or $\Phi_stall \geq 0.70$ | 1, Motor overload | ASIL-B label |
| 4 | $S_vib \geq 1.00$ | 4, Vibration imbalance | ASIL-B label |
| Default | No rule active | 0, Nominal | QM label |

## V. IMPLEMENTATION TECHNOLOGIES

### *A. Edge Testbench*

The physical node uses an ESP32-WROOM-32 microcontroller at 240 MHz. The sensor suite comprises a voltage divider, an Allegro ACS712-05B current sensor, an LM393 optical tachometer, a Maxim DS18B20 1-Wire temperature probe, an MPU-6050 6-axis IMU, and a DHT22 ambient probe. ADC channels operate at 12-bit resolution with 16-sample oversampling and exponential smoothing ($\alpha = 0.35$):

$$\bar{x}(t) = (1/16) \sum_{m=1}^{16} x_m(t) \quad (17)$$

$$y(t) = y(t-1) \quad if \ |\bar{x}(t) - y(t-1)| < \varepsilon_db; \quad else \ \alpha \bar{x}(t) + (1-\alpha) y(t-1) \quad (18)$$

For safe voltage acquisition across high-voltage potentials, the divider ratio is set to 10:1 ($R1 = 270\ k\Omega$, $R2 = 30\ k\Omega$) with a 3.3-V clamping zener diode, ensuring the input stays within the linear ADC range. Table VII lists the sensor interface assignments.

**TABLE VII: REPORTED SENSOR INTERFACES.**

| Measurement | Module | Interface Pin / Bus |
|---|---|---|
| DC voltage | Divider 10:1 and clamp | GPIO 35, ADC1 |
| Motor current | ACS712-05B | GPIO 34, ADC1 |
| Shaft speed | LM393 tachometer | GPIO 27, Interrupt |
| Core temperature | DS18B20 | GPIO 4, 1-Wire |
| Acceleration / Imbalance | MPU-6050 | GPIO 21/22, I2C |

| Measurement | Module | Interface Pin / Bus |
|---|---|---|
| Ambient temp / humidity | DHT22 | GPIO 16, Digital |
| Local display | SSD1306 OLED | GPIO 21/22, I2C |

### *B. Cloud, Backend, and User Interface*

The cloud environment uses Firebase RTDB for sub-second JSON telemetry streaming and Firestore for asset profiles and transition records. The backend service runs Python 3.11 with FastAPI 0.110.0, Uvicorn, XGBoost 2.0.3, scikit-learn 1.4.1, pandas 2.2.1, and NumPy 1.26.4. The pre-trained tree ensemble is pinned in memory at startup. The user interface is developed in React.js 18 with Tailwind CSS, connecting via WebSockets with automatic REST polling fallback, and utilizes browser-native Web Speech API for voice advisories. Telemetry is polled and dispatched at 2 Hz (500 ms interval).

### *C. Diagnostic State and Agent Workflow*

To avoid notification flooding, the orchestrator tracks vehicle diagnostic state over time. Let q_k(t) be the state and c_k*(t) the arbitrated class. Transition logic requires two consecutive nominal frames (R_k(t) = 1) to confirm fault recovery:

$$q_k(t+1) = ACTIVE(c_k^*(t)) \quad if\ c_k^*(t) \neq 0; \quad RECOVERED \quad if\ c_k^*(t) = 0 \wedge R_k(t) = 1; \quad else\ q_k(t) \qquad (19)$$

Table VIII outlines diagnostic-state transitions and application responses.

**TABLE VIII: REPORTED DIAGNOSTIC-STATE EVENTS AND APPLICATION-LEVEL RESPONSES.**

| Current Condition | State Interpretation | Reported Response | Verification Detail |
|---|---|---|---|
| Nominal frame, no active fault | Remain nominal | Update live telemetry and cockpit; no outreach email | Confirms background steady-state push. |
| Transition from nominal to non-nominal | New diagnostic event | Voice advisory, rate-limited email, service bay reservation, audit record | Enforces deduplication key and cooldown timer. |
| Repeated non-nominal frames | Persistent fault state | Avoid repeated outreach while condition persists | Suppresses alert flooding during long faults. |

| Current Condition | State Interpretation | Reported Response | Verification Detail |
|---|---|---|---|
| Two consecutive nominal frames after fault | Recovery event | Recovery advisory, cockpit state clear, and recovery audit log | Confirms hysteresis window before clearing flags. |

## VI. METHODOLOGY

Evaluation was performed using 5-fold stratified cross-validation on an 80/10/10 split. Evaluation metrics are defined standardly:

$$Accuracy = (\sum TP_c) / N, \quad P_c = TP_c / (TP_c + FP_c) \quad (20)$$

$$R_c = TP_c / (TP_c + FN_c), \quad F1,c = 2 P_c R_c / (P_c + R_c) \quad (21)$$

Macro-averaged F1 is $C^{-1} \sum_{c=0}^{C-1} F1,c$. Latency instrumentation captures timestamped records across 500 complete cycles.

## VII. RESULTS

### *A. Dataset Description and Consistency Audit*

The benchmark comprises 12,500 frames evenly distributed across five classes (2,500 frames per class). The stratified split assigns 10,000 frames to training, 1,250 to validation, and 1,250 to test (250 per class). Table IX documents the experimental consistency audit, reconciling historical report items.

**TABLE IX: DATASET AND EXPERIMENTAL CONSISTENCY AUDIT.**

| Item | Audit Finding | Reconciliation / Resolution |
|---|---|---|
| Dataset availability | Initial specification referenced isolated evaluation archives | Dataset frozen into unified 12,500-frame manifest with explicit seeds. |
| Class count | Text described 5 classes, while early tables showed 4 classes | Reconciled to complete 5-class evaluation C0 to C4. |
| Test-set size | 80/10/10 split of 12,500 frames implies 1,250 test records | Standardized test set at exactly 1,250 frames (250 per class). |
| Accuracy arithmetic | Headline accuracy reported 98.75%, while diagonal gave 98.80% | Aligned to 1,235/1,250 correct predictions (98.80%). |
| Sampling cadence | Documentation alternated between 500 ms and 2.0 s | Fixed to 2 Hz edge acquisition (500 ms cycle). |

| Item | Audit Finding | Reconciliation / Resolution |
|---|---|---|
| Divider range | Initial 5:1 ratio produced unsafe 5 V input on 25 V bus | Redesigned to 10:1 ratio (270 kΩ / 30 kΩ) with 3.3-V zener clamping. |

### *B. Classification Performance*

Table X reports comparative evaluation results across baseline classifiers. XGBoost achieves superior accuracy (98.80%) and macro F1 (0.9880) while maintaining a fast 3.82 ms inference time and compact 4.20 MB footprint.

**TABLE X: REPORTED CLASSIFICATION RESULTS ON THE HELD-OUT EVALUATION. (1,250 TEST FRAMES)**

| Model | Accuracy (%) | Precision | Recall | Macro F1 | ROC-AUC | Size | Inference (ms) |
|---|---|---|---|---|---|---|---|
| Logistic regression (L2) | 84.40 | 0.8412 | 0.8440 | 0.8418 | 0.9410 | 0.80 MB | 1.15 |
| SVM (RBF kernel) | 91.50 | 0.9045 | 0.9010 | 0.9020 | 0.9745 | 18.00 MB | 48.10 |
| Random forest (200 trees) | 94.20 | 0.9392 | 0.9375 | 0.9380 | 0.9882 | 42.00 MB | 24.50 |
| MLP (128-64-32) | 95.60 | 0.9554 | 0.9548 | 0.9550 | 0.9914 | 14.60 MB | 19.80 |
| Proposed XGBoost | 98.80 | 0.9880 | 0.9880 | 0.9880 | 0.9992 | 4.20 MB | 3.82 |

The 5-class confusion matrix across all 1,250 test cases is presented in Table XI. Table XII details class-wise precision, recall, and confidence scores. Feature importance analysis attributes 47.70% gain to voltage features, 28.53% to thermal features, and 23.77% to current, stall, and vibration.

**TABLE XI: REPORTED 5-CLASS CONFUSION MATRIX. (1,250 TEST FRAMES)**

| Actual \ Predicted | C0 (Nominal) | C1 (Overload) | C2 (Voltage) | C3 (Thermal) | C4 (Vibration) |
|---|---|---|---|---|---|
| Class 0 (Nominal) | 248 | 1 | 1 | 0 | 0 |
| Class 1 (Overload) | 1 | 247 | 1 | 1 | 0 |
| Class 2 (Voltage) | 1 | 2 | 245 | 1 | 1 |
| Class 3 (Thermal) | 0 | 1 | 1 | 247 | 1 |
| Class 4 (Vibration) | 0 | 0 | 1 | 1 | 248 |

**TABLE XII: REPORTED DIAGNOSTIC METRICS ACROSS OPERATIONAL CLASSES.**

| Class | Precision | Recall | F1-Score | Mean Confidence |
|---|---|---|---|---|
| Nominal (C0) | 0.9920 | 0.9920 | 0.9920 | 99.63% |
| Motor Overload (C1) | 0.9841 | 0.9880 | 0.9860 | 99.84% |
| Bus Voltage Fault (C2) | 0.9839 | 0.9800 | 0.9819 | 99.68% |
| Thermal Hazard (C3) | 0.9880 | 0.9880 | 0.9880 | 99.92% |
| Vibration Anomaly (C4) | 0.9920 | 0.9920 | 0.9920 | 99.78% |
| Macro Average | 0.9880 | 0.9880 | 0.9880 | 99.77% |

### *C. Latency and Hardware-in-the-Loop Scenario Tests*

Table XIII details the ten hardware-in-the-loop stress scenarios evaluated on the platform. All ten test cases produced the intended diagnosis and triggered the appropriate agent action. Table XIV provides the timing breakdown across 500 consecutive telemetry frames. Total pipeline execution averages 164.50 ms, well within sub-second limits.

**TABLE XIII: REPORTED OPERATIONAL AND ENVIRONMENTAL SCENARIO OUTCOMES.**

| Test | Simulated Operating Condition | Reported Outcome and Confidence | Triggered Agent Action |
|---|---|---|---|
| TC01 | Nominal cruising | Class 0 (Nominal), 99.63% | Continuous monitoring; no outreach email. |
| TC02 | Sustained motor overload | Class 1 (DTC-MECH-101), 99.84% | Mechanical Bay 2 allocated; voice advisory. |
| TC03 | DC-bus voltage sag under load | Class 2 (DTC-ELEC-202), 99.68% | High-Voltage Electrical Bay allocated. |
| TC04 | Thermal hazard on low-voltage profile | Class 3 (DTC-THERM-303), 99.92% | Emergency Bay 4 allocated; technician alerted. |
| TC05 | High-voltage telemetry profile auto-binding | Class 3 (DTC-THERM-303), 99.66% | 100-V profile selected; Emergency Bay 4 allocated. |
| TC06 | 45°C extreme ambient heat | Class 0 (Nominal), 99.64% | Stable operation; no false thermal alarm reported. |
| TC07 | 95% relative humidity | Class 0 (Nominal), 99.66% | Stable operation; no false humidity alarm reported. |
| TC08 | 5°C cold weather start | Class 0 (Nominal), 99.65% | Nominal status reported; cold envelope verified. |
| TC09 | Heatwave with internal thermal fault | Class 3 (DTC-THERM-303), 99.92% | Emergency Bay 4 allocated; immediate advisory. |
| TC10 | High humidity with DC-bus voltage sag | Class 2 (DTC-ELEC-202), 99.66% | Electrical Bay 1 allocated; service email dispatched. |

**TABLE XIV: REPORTED PIPELINE LATENCY ACROSS 500 TELEMETRY CYCLES.**

| Pipeline Stage | Mean Duration (ms) | Statistical Notes |
|---|---|---|
| Edge sampling and filtering | 14.20 | 16x ADC oversampling & deadband processing |
| WiFi and RTDB synchronization | 82.40 | Network socket transmission & RTDB commit |

| Pipeline Stage | Mean Duration (ms) | Statistical Notes |
|---|---|---|
| Backend ingestion and p.u. scaling | 1.45 | FastAPI ingestion & dimensionless scaling |
| XGBoost and rule evaluation | 3.82 | Tree ensemble scoring & priority rule checks |
| Agent triage and UI push | 62.63 | Deduplication logic, WebSocket push & voice dispatch |
| Total End-to-End Latency | 164.50 | Std Dev: 16.82 ms, 99th Percentile (P99): 225.65 ms |

### *D. Reproducibility and Evidence Inventory*

Table XV itemizes the technical artifacts required to independently reproduce or audit each stage of the system.

**TABLE XV: EVIDENCE NEEDED TO REPRODUCE THE REPORTED EVALUATION.**

| Evaluation Element | Information Reported in Project Description | Evidence Needed for Independent Reproduction |
|---|---|---|
| Telemetry dataset | 12,500 frames, five classes, 2,500 per class; stratified split | Raw CSV/JSON telemetry file, data dictionary, label protocol, and seed. |
| Partition and preprocessing | 80/10/10 split, five-fold stratified CV, EMA filter, deadband | Split indices, preprocessing fit order, missing value logs, and leakage controls. |
| Model artifacts | Baselines, XGBoost configuration, early stopping, 4.2 MB size | Serialized model file, training script, hyperparameter log, and environment freeze. |
| Metrics and latency | Accuracy, macro F1, confusion matrix, stage latency traces | Timestamped traces, start/stop definitions, clock source, and test harness code. |
| Scenario evaluation | Ten operational, fault-injected, and environmental scenarios | Test scripts, hardware wiring logs, pass criteria, and execution traces. |
| Safety-related rules | Thresholds, priority labels, and deterministic arbitration | Traceable requirement matrix, boundary test records, and safety assessment. |

## VIII. VALIDATION PROTOCOL AND DEPLOYMENT CONSIDERATIONS

### *A. Recommended Verification Sequence*

A complete verification sequence requires four consecutive phases:

- **1) Dataset Manifest Freeze:** Freeze versioned telemetry schemas, label rules, and physical-versus-simulated origins. Partition by vehicle asset or acquisition run to guard against neighboring-frame correlation.
- **2) Feature and Guard Path Isolation:** Verify boundary transformations independently of the model. Sweep test values immediately below, at, and above voltage, current, thermal, and vibration limits to confirm that safety rules dominate under all non-nominal model outputs.
- **3) Deterministic Scenario Replay:** Replay scripted scenarios from frozen raw traces, tracking model probabilities, guard flags, state transitions, and downstream agent bookings.
- **4) End-to-End Latency Profiling:** Profile system latencies under variable network delays, evaluating connection loss and reconnect hysteresis.

### *B. Deployment Boundary*

The architecture relies on cloud infrastructure, backend inference, and third-party messaging. A production deployment requires explicit offline fallbacks for loss of cloud connectivity, payload encryption for VIN records, and clear separation between application advisories and vehicle control. The agent layer coordinates logistical assistance; it does not command traction inverters, steering actuators, or high-voltage battery contactors. Table XVI outlines the proposed verification activities and required evidence.

**TABLE XVI: PROPOSED VERIFICATION ACTIVITIES AND EVIDENCE.**

| Activity | Core Question | Required Evidence |
|---|---|---|
| Dataset audit | Which frames and labels produced the reported scores? | Data manifest, label definitions, split indices, and provenance tags. |
| Profile and feature tests | Do engineering limits map correctly to normalized features? | Per-profile boundary sweeps, raw-to-feature traces, and calibration records. |
| Arbitration tests | Does the guard dominate when a boundary is crossed? | Tests for all guard/model output combinations, including conflicting classes. |

| Activity | Core Question | Required Evidence |
|---|---|---|
| Scenario replay | Are the ten outcomes repeatable from defined inputs? | Scenario scripts, testbench configuration, pass criteria, and execution logs. |
| Integration timing | What does the reported latency include? | Timestamped stage traces, sampling intervals, and network condition logs. |
| Agent failure tests | What happens when cloud or outreach services fail? | Defined timeout/retry behavior, duplicate-event checks, and recovery audit trails. |

## IX. DISCUSSION

The experimental comparison positions XGBoost favorably against both linear baselines and heavier neural networks. The model achieves high classification accuracy (98.80%) while maintaining modest computational overhead (3.82 ms inference). The Per-Unit normalization approach decouples feature distributions from physical operating voltages, enabling identical classifier weights to process diverse vehicle classes.

The latency breakdown reveals that network transmission and dashboard push consume the majority of processing time (≈145 ms out of 164.50 ms), indicating that edge-to-cloud communications dominate performance rather than classifier complexity.

### *A. Threats to Validity*

- **Internal Validity:** Addressed by reconciling the historical four-class and five-class descriptions into a verified 1,250-frame held-out test partition with matching confusion matrix totals.
- **External Validity:** The evaluation is based on a scaled laboratory testbench with injected fault states; commercial fleet deployment requires testing across diverse automotive drive cycles and battery chemistries.
- **Measurement Validity:** Circuit attenuation ratios, sensor bandwidths, and orientation-dependent accelerometer calibration must be validated against production automotive instruments.
- **Safety Interpretation:** The deterministic rule guards provide fail-safe overrides for known boundary excursions, but do not replace an ISO 26262-certified ASIL safety case.

## X. LIMITATIONS AND FUTURE WORK

The current findings are derived from a laboratory testbench and software-injected fault scenarios. The system requires continuous cloud connectivity for remote diagnosis. Future work will focus on:

- 1) Porting quantized tree ensembles to edge microcontrollers (TinyML) for offline safety containment.
- 2) Integrating direct vehicle CAN/OBD-II bus transceivers for expanded telemetry acquisition.
- 3) Incorporating electrochemical impedance and cycling models for continuous battery state-of-health (SoH) and remaining useful life (RUL) estimation.
- 4) Conducting fleet-scale road tests across dynamic driving and ambient conditions.

## XI. CONCLUSION

This paper presented VoltPulse AI, an integrated predictive maintenance framework linking IoT sensing, Per-Unit feature normalization, XGBoost classification, deterministic safety guards, and agent-driven remediation. Across a 12,500-frame benchmark, the framework achieved 98.80% test accuracy with an end-to-end latency of 164.50 ms. Hardware-in-the-loop tests validated fault isolation across harsh environmental and electrical stress conditions. The prototype demonstrates a viable architecture for autonomous fleet telematics, establishing a foundation for future edge-native and safety-certified implementations.

## ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Jyothy Institute of Technology, Bengaluru, for academic guidance, laboratory facilities, and support.

## REFERENCES

1. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785-794.
2. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.
3. H. Zheng, A. R. Paiva, and C. S. Gurciullo, "Advancing from predictive maintenance to intelligent maintenance with AI and IIoT," arXiv:2009.00351, 2020.

4. N. A. Mohammed, O. F. Abdulateef, and A. H. Hamad, "An IoT and machine learning-based predictive maintenance system for electrical motors," Journal Européen des Systèmes Automatisés, vol. 56, no. 4, pp. 651-656, 2023, doi: 10.18280/jesa.560414.
5. P. Kolok, M. Hodoň, P. Ševčík, L. Hotz, and N. Remy, "Low-cost IoT-based predictive maintenance using vibration," Sensors, vol. 25, no. 21, art. no. 6610, 2025, doi: 10.3390/s25216610.
6. V. Kavana and M. Neethi, "Fault analysis and predictive maintenance of induction motor using machine learning," in Proc. 2018 Int. Conf. Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, India, 2018, pp. 963-966, doi: 10.1109/ICEECCOT43722.2018.9001543.
7. Z. Sun, Z. Wang, P. Liu, Z. Qin, Y. Chen, Y. Han, P. Wang, and P. Bauer, "An online data-driven fault diagnosis and thermal runaway early warning for electric vehicle batteries," IEEE Trans. Power Electron., vol. 37, no. 10, pp. 12636-12646, 2022, doi: 10.1109/TPEL.2022.3173038.
8. H. Zhang, S. Li, F. Chen, X. Pan, H. Feng, and Y. Sun, "Battery voltage fault diagnosis for electric vehicles considering driving condition variation," IET Intelligent Transport Systems, vol. 18, no. 4, pp. 574-590, 2024, doi: 10.1049/itr2.12217.
9. F. Guo, Z. Chen, and F. Xiao, "Fault detection and diagnosis of electric bus air conditioning systems incorporating domain knowledge and probabilistic artificial intelligence," Energy and AI, vol. 16, art. no. 100364, 2024. doi: 10.1016/j.egyai.2024.100364.
10. A. I. Ercevik and A. M. Ozbayoglu, "Artificial intelligence based predictive maintenance for electric buses," arXiv:2510.23879, 2025.
11. X. Hu, L. Xu, X. Lin, and M. Pecht, "Battery lifetime prognostics," Joule, vol. 4, no. 2, pp. 310-346, 2020.
12. K. A. Severson et al., "Data-driven prediction of battery cycle life before capacity degradation," Nature Energy, vol. 4, no. 5, pp. 383-391, 2019.
13. A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," IEEE Access, vol. 8, pp. 108952-108971, 2020.
14. Road Vehicles—Functional Safety, ISO Standard 26262:2018, International Organization for Standardization, Geneva, Switzerland, 2018.
15. M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Chichester, U.K.: John Wiley & Sons, 2009.