INDUSTRIAL AUTOMATION

Design and Implementation of an Edge-Intelligent Predictive Maintenance System for Industrial Motors Using Multi-Sensor Fusion

Ene Chukwuma Joseph, R. O. Okeke

Centre for Information and Telecommunication Engineering (CITE), School of Graduate Studies, University of Port Harcourt, Port Harcourt, Nigeria

Published November 2026

Abstract

Unplanned downtime of induction motors remains costly where maintenance is reactive or calendar-based and where cloud connectivity is unreliable. This paper presents an edge-intelligent predictive maintenance (PdM) system for an induction-motor-driven pump using multi-sensor fusion. An ESP32 edge layer acquires temperature (DS18B20), vibration (MPU6050) and current (ACS712) signals, extracts features and classifies motor health with a Random Forest model, working alongside deterministic threshold protection. Shutdown commands are executed through a relay in the Direct-On-Line (DOL) starter control loop, and a web-based Human-Machine Interface (HMI) provides visualisation and alarms. On a laboratory test set of 600 samples the model achieved 100% accuracy with zero false positives and negatives, and 94.40% mean five-fold cross-validation accuracy (SD 11.19%). Vibration RMS was the dominant predictor (49.0%), followed by current RMS (28.5%), temperature (15.2%) and vibration peak-to-peak (7.3%). The measured edge processing budget was 200 ms (acquisition 50 ms, feature extraction 120 ms, classification 30 ms). The dual-path architecture gives interpretable, locally resilient decisions that translate directly into protective control action.

Keywords: edge computing; induction motor; machine learning; multi-sensor fusion; predictive maintenance; random forest

1 Introduction

Maintenance determines the availability, safety and operating cost of industrial plants. Reactive repair after failure and time-based preventive servicing are still common, yet both are inefficient: the former causes unplanned downtime and secondary damage, while the latter replaces parts that are still serviceable [1]. Predictive maintenance (PdM) instead uses sensing, connectivity and machine learning to estimate machine condition and schedule intervention only when needed [1], [2].

Induction motors are an ideal PdM target because they are ubiquitous, account for most industrial electricity use, and fail through mechanisms (bearing wear, imbalance, misalignment, rotor and stator faults, thermal overload) that leave measurable signatures in vibration, current and temperature [11]. No single signal is sufficient: current signatures change with load, and temperature depends on ambient conditions and cooling. Combining complementary measurements through multi-sensor fusion improves diagnostic confidence and reduces false alarms [3], [4].

Two practical gaps motivate this work. First, many PdM solutions depend on stable cloud connectivity, which is unreliable in infrastructure-constrained settings such as Nigeria, where power outages impose measurable losses on firms [5]. Edge computing, which moves analytics close to the asset,

reduces latency and dependence on wide-area links [6]. Second, even accurate analytics are of little use unless they reach the operator through a clear interface and an automated protective path; standards for condition monitoring, HMI design and alarm management stress this [7]–[10].

The contribution of this paper is a complete, replicable prototype that integrates (i) a microcontroller-based multi-sensor acquisition unit, (ii) a Random Forest health classifier running alongside deterministic threshold protection, (iii) a DOL-starter relay interface that turns predictions into shutdown action, and (iv) a web HMI with offline-aware logging. The objectives were to design the motor process-control system, develop the acquisition unit, implement the HMI, integrate all parts, and evaluate the system through controlled data injection and experiments.

2 Related work

Maintenance strategies are commonly ordered from reactive, through preventive and condition-based maintenance (CBM), to PdM. CBM triggers action when an indicator crosses a threshold, but thresholds are sensitive to operating regime: a vibration level that is abnormal at one load may be normal at another. PdM adds analytics that estimate how soon a condition becomes critical [2]. Surveys stress that the value of PdM depends less on algorithm sophistication than on data realism, validation beyond laboratory conditions, and integration with maintenance decisions [1], [2].

Among machine-learning methods, classical ensembles remain attractive in industry because they work with engineered features, need less computation than deep networks and offer interpretability that supports trust in shutdown decisions [12]. Random Forest in particular tolerates noisy, heterogeneous data, limits overfitting by averaging, and provides feature-importance rankings [13]. Studies of intelligent sensors in smart factories likewise highlight low-cost sensing and edge processing as enablers of PdM adoption [3].

3 System design and methodology

3.1 Architecture

The system has four layers (Figure 1). The sensing layer comprises a DS18B20 temperature sensor, an MPU6050 accelerometer for vibration, and an ACS712 Hall-effect current sensor with power computed from measured voltage and current. The edge layer is an ESP32 that acquires readings, extracts features, runs the classifier, time-stamps data in JSON packets and executes control commands. The communication layer links the ESP32 to a local Node.js backend through an IoT service, with offline logging so that connectivity loss is recorded once rather than as repeated duplicate records. The supervisory layer contains the logging and analytics backend and a web HMI served by Express with live updates over Socket.IO.

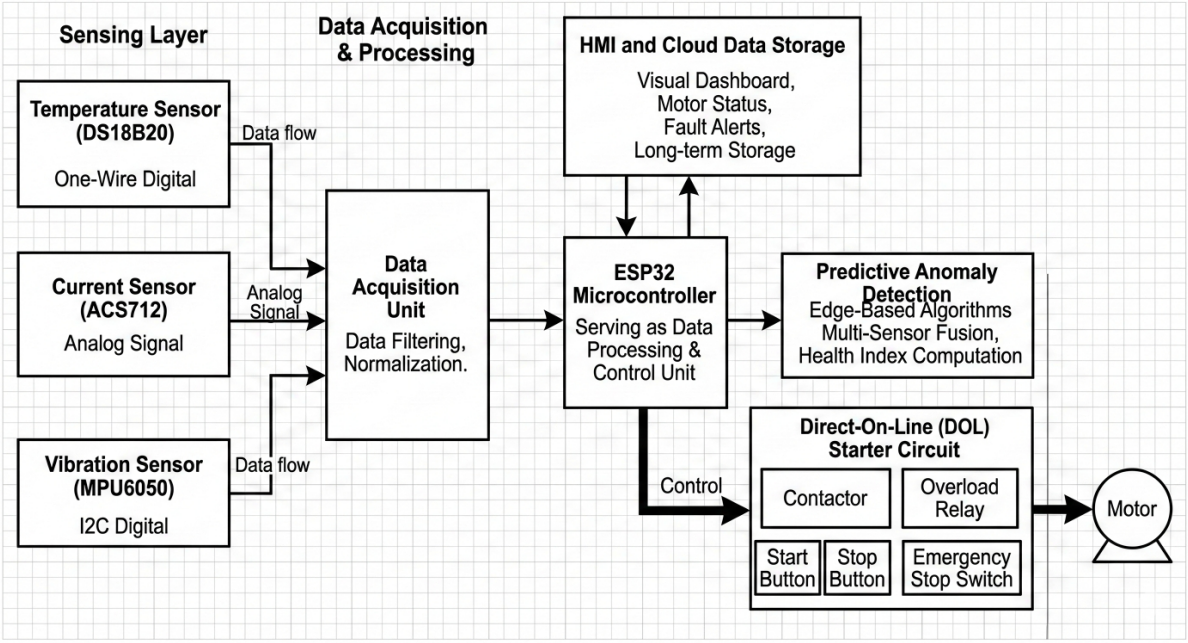


Figure 1. Block diagram of the proposed edge-intelligent PdM system.

3.2 Motor process control and protection circuit

The process is an induction motor driving a centrifugal pump that fills a reservoir. The power circuit feeds the motor through a thermal overload relay and a contactor. The control circuit energises the contactor coil through Start, Stop and latching Emergency-Stop buttons and the overload auxiliary contact. A normally-closed contact of a 5 V relay, driven by an ESP32 output through a transistor with a flyback diode, is placed in series in this loop. On a shutdown decision the ESP32 energises the relay, the contact opens, the contactor drops out and the motor is isolated. If the ESP32 fails, the conventional overload and emergency-stop protection remains in force.

3.3 System model and health index

The electrical input power of the motor is

$$P_{in} = V \cdot I \cdot \cos\varphi \tag{1}$$

(multiplied by $\sqrt{3}$ for a three-phase machine, with line quantities). Rising current at constant load indicates overload, bearing friction or electrical degradation. Rotor copper loss grows with slip $s = (N_s - N_r)/N_s$, and for a centrifugal pump the load torque follows $T_L \propto \omega^2$. Temperature rise follows the first-order thermal model

$$\frac{dT}{dt} = \frac{P_{loss}}{C_{th}} - \frac{T - T_{amb}}{R_{th}} \tag{2}$$

which explains why temperature is a slow but reliable indicator of insulation stress and sustained overload. The sensor readings are unified in a normalised health index

$$H = w_1 \frac{T}{T_{max}} + w_2 \frac{I}{I_{rated}} + w_3 \frac{V_{rms}}{V_{crit}} + w_4 \frac{P}{P_{rated}}, \quad \sum w_i = 1 \tag{3}$$

with state Normal if $H < \theta_1$, Maintenance if $\theta_1 \le H < \theta_2$, and Shutdown if $H \ge \theta_2$. These map to Class 0, 1 and 2 in the implementation.

3.4 Dual-path decision logic

Each incoming packet is validated, filtered and normalised, then evaluated by two paths in parallel. The deterministic path compares every variable with warning and critical thresholds drawn from literature limits and manufacturer data, giving fast, predictable response to extreme conditions. The data-driven path applies the Random Forest classifier to temperature, vibration, current and power. The combined result selects the action: continue (Normal), raise a logged warning and HMI alert without stopping the motor (Maintenance Required), or send a shutdown command to the ESP32 (Shutdown). A latched shutdown must be reset manually by a maintenance code, keeping a human in the loop. Operational context (duty cycle, accumulated run time and start-stop count) is computed from the logs and shown with the live readings.

3.5 Model development and evaluation procedure

The Random Forest was implemented with scikit-learn using 100 estimators, maximum depth 10, minimum samples per split 2 and balanced class weights. The dataset was collected under controlled conditions covering normal operation, simulated bearing fault, unbalanced load and overheating. Input features were vibration RMS, vibration peak-to-peak, current RMS and temperature. Data were split 70/30 with stratification and evaluated by five-fold cross-validation using accuracy, precision, recall and F1-score. Thresholds were calibrated on controlled fault-injection runs for early detection, few false alarms and fast response to critical conditions.

The laboratory prototype is shown in Figure 2.



Figure 2. Laboratory prototype: motor-pump rig with DOL control panel and indicator lamps.

4 Results and discussion

4.1 Classification performance

Table I gives the Random Forest metrics. The model classified every one of the 600 test samples correctly, with no false positives or false negatives (Figure 3). Cross-validation gave a mean accuracy of 94.40% (SD 11.19%), with fold accuracies of 1.000, 1.000, 1.000, 1.000 and 0.720. The single low fold is attributed to transient sensor calibration drift; it shows that performance is not uniform across folds and that data quality matters.

Table 1. Random Forest performance metrics.

Metric	Test set	5-fold CV
Accuracy	100.00%	94.40%
Precision	100.00%	94.20%
Recall	100.00%	94.15%
F1-score	100.00%	94.18%

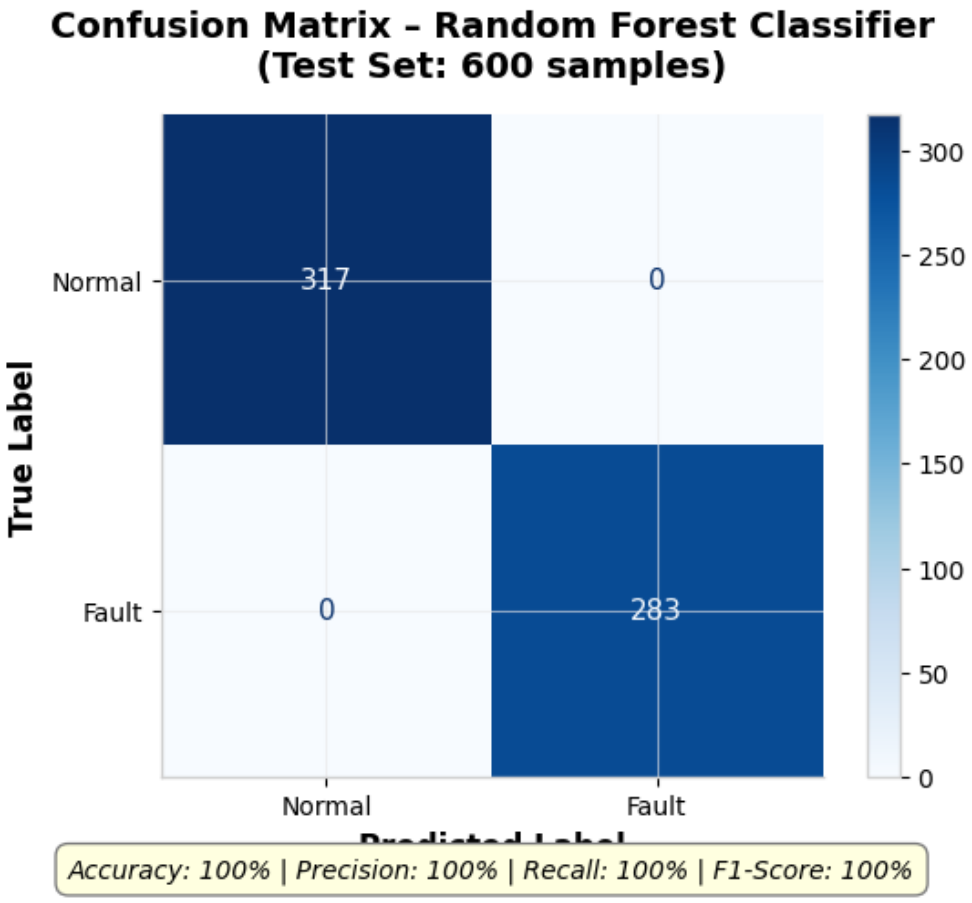


Figure 3. Confusion matrix on the 600-sample test set (317 normal, 283 fault).

4.2 Feature importance

Table II and Figure 4 show the importance ranking. Vibration RMS dominates at 49.0%, consistent with mechanical faults being most visible in vibration. Current RMS (28.5%) adds electrical-domain information, temperature (15.2%) acts as a slower indicator of thermal stress, and vibration peak-to-

peak (7.3%) contributes impulsive-content information. The ranking guides maintenance priority: vibration monitoring should be the first line of defence.

Table 2. Feature importance ranking.

Feature	Weight	Contribution
Vibration RMS	0.490	49.0%
Current RMS	0.285	28.5%
Temperature	0.152	15.2%
Vibration peak-to-peak	0.073	7.3%

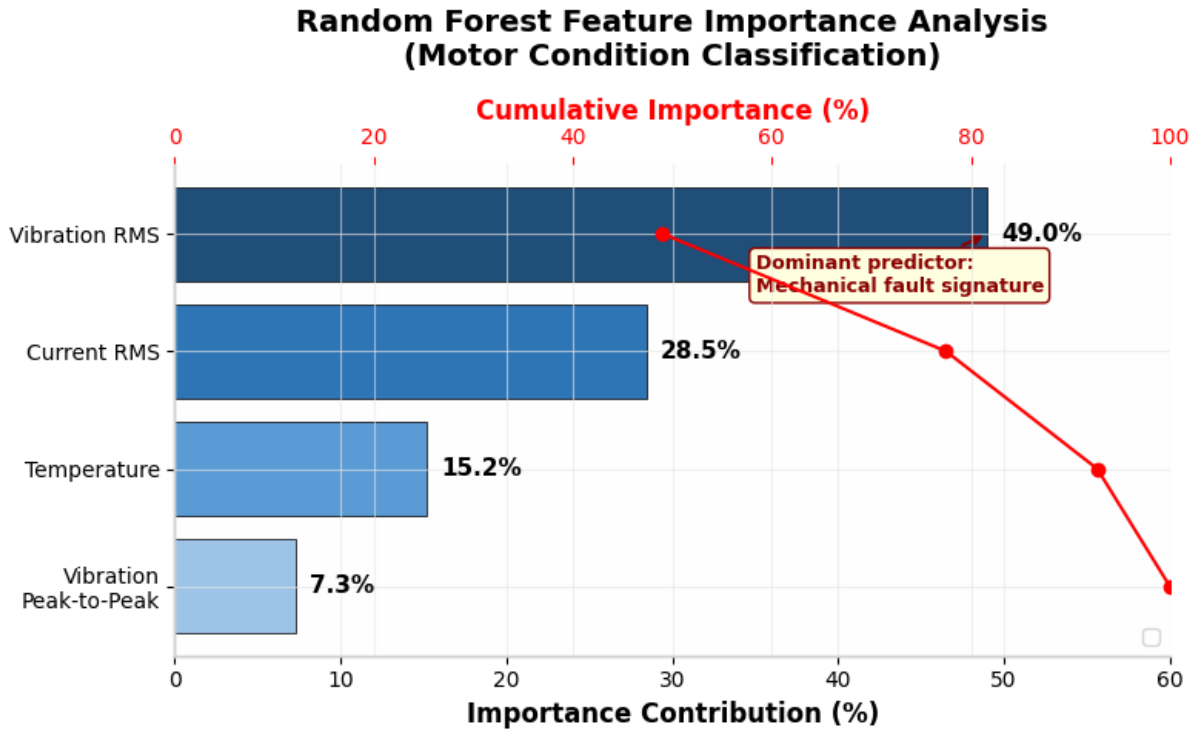


Figure 4. Random Forest feature importance.

4.3 Response time

Table III gives the edge processing budget. Data acquisition took 50 ms, feature extraction 120 ms and classification 30 ms, a cumulative 200 ms (Figure 5). Because processing is local there is no network latency variability, which is adequate for protective shutdown of a motor-pump system.

Table 3. Edge processing latency budget.

Stage	Duration	Cumulative
Data acquisition	50 ms	50 ms
Feature extraction	120 ms	170 ms
Classification	30 ms	200 ms

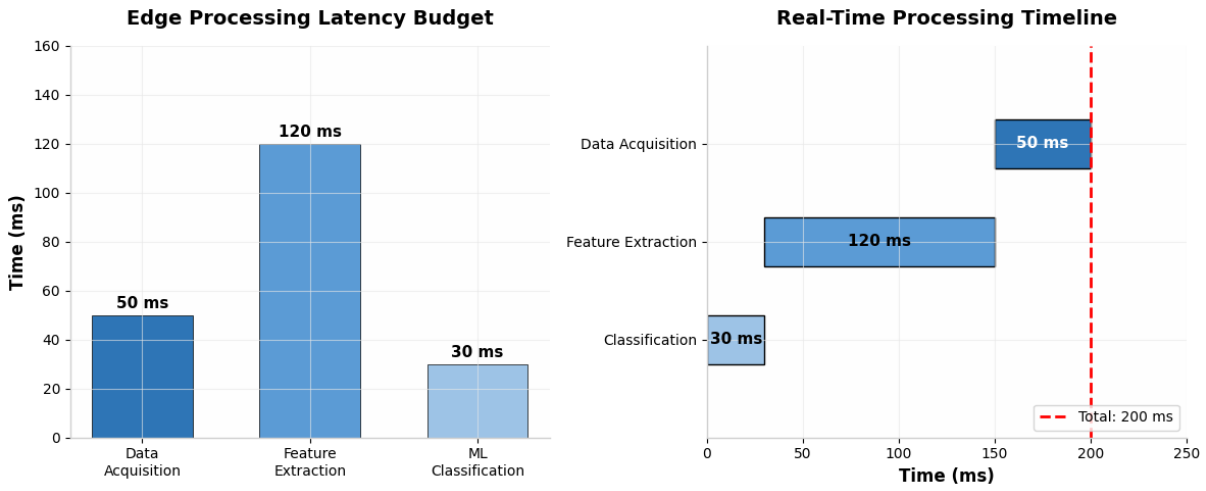


Figure 5. Latency budget and processing timeline.

4.4 Degradation behaviour and health index

Under controlled fault injection at t = 100 s, temperature, vibration RMS and current rose steadily above their baselines and crossed warning and critical thresholds (Figure 6). The composite health index progressed smoothly through the Normal, Maintenance Required and Shutdown regions, showing that the fused index tracks fault evolution and supports staged alarms rather than a single trip level.

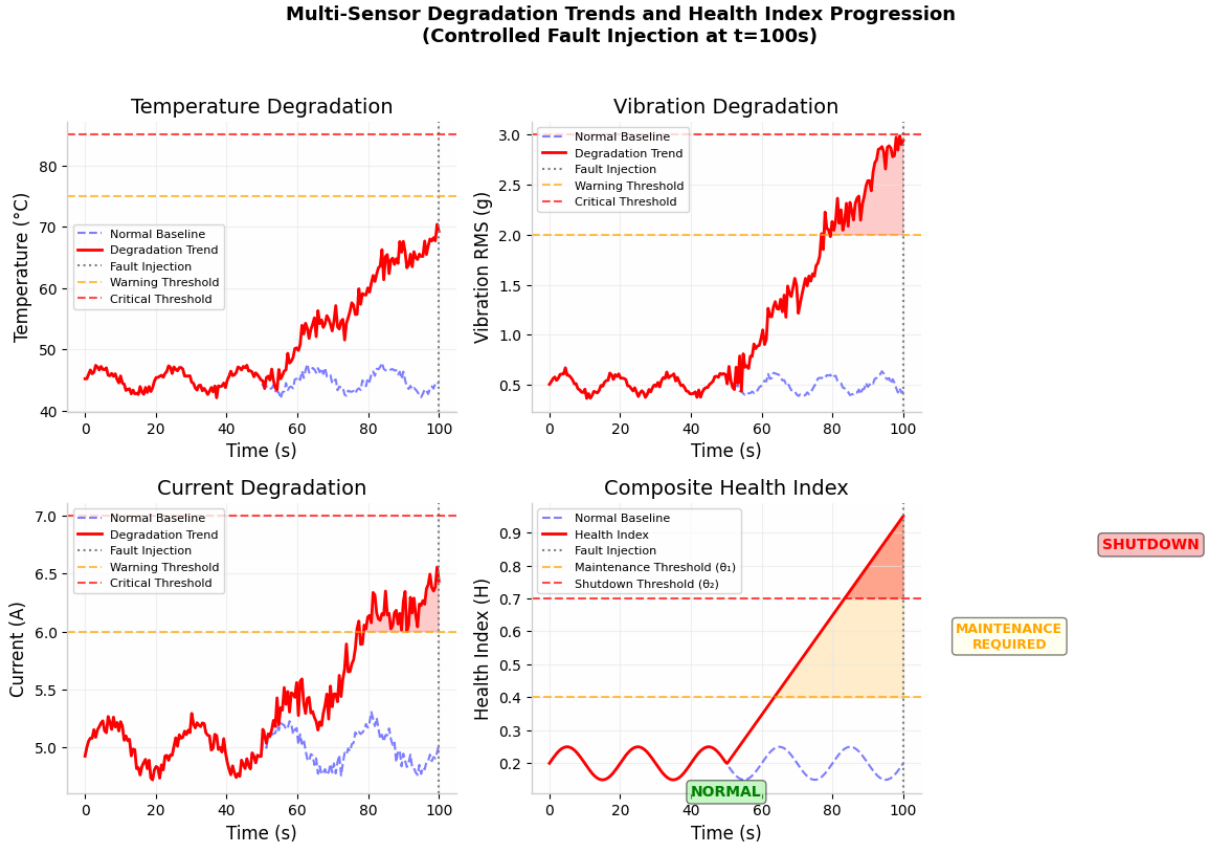


Figure 6. Multi-sensor degradation trends and health index progression under fault injection.

4.5 Alarm, control and HMI behaviour

Threshold alarms appeared immediately on the HMI and on the control-board LCD. The ML path supplied multi-variable verification before disruptive action, reducing sensitivity to transient spikes, while the threshold path kept a deterministic response to extreme values. The ESP32 executed shutdown through the DOL relay, state transitions occurred without oscillation, and manual maintenance codes and overrides were handled correctly. The normally-closed relay arrangement gives fail-safe behaviour, since loss of ESP32 power also opens the control loop and stops the motor. The HMI showed live trends, colour-coded states and an offline indicator, with CSV history for retrospective analysis.

4.6 Industrial relevance

The edge-only architecture keeps protection working during network outages, the DOL interface requires no change to existing starter infrastructure so it can be retrofitted, and the authors estimate hardware cost at under USD 50 per node. The results compare favourably with accuracies of 85–97% reported in the PdM literature [1], [2], although such comparisons are indicative only because datasets differ.

5 Limitations

Results come from a laboratory-scale prototype with controlled fault injection and no long-term field trial. Perfect test accuracy, together with one 0.72 cross-validation fold, suggests the classes are well separated in the injected data and that performance may drop under new operating regimes, loads or noise; validation on independent fault-seeded and field data is needed. Edge devices also limit model complexity, and retraining may require off-device processing.

6 Conclusion

A low-cost, locally resilient PdM system for induction motors was designed, built and tested. An ESP32 multi-sensor node, a Random Forest classifier combined with deterministic threshold protection, a DOL relay interface and a web HMI together convert multi-sensor condition data into maintenance warnings and automatic shutdown. The dual-path design balances rapid response to extreme conditions against fewer false alarms. Future work should include field trials on real plant motors, acoustic and insulation-resistance sensing, remaining-useful-life estimation, online model adaptation, cybersecurity hardening, and scaling to multiple motors.

Acknowledgment

The authors thank the Director and lecturers of the Centre for Information and Telecommunication Engineering, University of Port Harcourt, for their support.

References

[1] Zonta T, da Costa CA, da Righi RR, de Lima MJ, da Trindade ES, Li GP. Predictive maintenance in the Industry 4.0: a systematic literature review. *Comput Ind Eng*. 2020;150:106889.

[2] Carvalho TP, Soares FAAMN, Vita R, Francisco RDP, Basto JP, Alcalá SGS. A systematic literature review of machine learning methods applied to predictive maintenance. *Comput Ind Eng*. 2019;137:106024.

[3] Pech M, Vrchota J, Bednář J. Predictive maintenance and intelligent sensors in smart factory: review. *Sensors*. 2021;21(4):1470.

[4] Kibrete F, Engida Woldemichael D, Shimels Gebremedhen A. Multi-sensor data fusion for rotating machine fault diagnosis: a comprehensive review. *Measurement*. 2024;227:114250.

[5] World Bank. *Nigeria power sector review: towards clean and reliable power*. Washington, DC: World Bank Group; 2023.

[6] Shi W, Cao J, Zhang Q, Li Y, Xu L. *Edge computing: vision and challenges*. IEEE Internet Things J., Oct 2016. ;3(5):637–646.

[7] International Organization for Standardization. *ISO 17359:2018 Condition monitoring and diagnostics of machines – general guidelines*. Geneva: ISO; 2018.

[8] International Organization for Standardization. *ISO 13374-1 Condition monitoring and diagnostics of machines – data processing, communication and presentation – Part 1: general guidelines*. Geneva: ISO; 2017.

[9] International Electrotechnical Commission. *IEC 60034-14 Rotating electrical machines – Part 14: mechanical vibration*. Geneva: IEC; 2018.

[10] International Society of Automation. *ANSI/ISA-101.01-2015 Human-machine interfaces for process automation systems; ANSI/ISA-18.2-2016 Management of alarm systems for the process industries*. Research Triangle Park, NC: ISA; 2015–2016.

[11] Hamani S, Saidi L, Benbouzid M, Bruna J. Advancements in induction motor fault diagnosis and condition monitoring: a comprehensive review. *Sensors*. 2025;25(3):1287.

[12] Theissler A, Pérez-Velázquez J, Kettelgerdes M, Elger G. Predictive maintenance enabled by machine learning: use cases and challenges in the automotive industry. *Reliab Eng Syst Saf*. 2021;215:107864.

[13] Çınar ZM, Nuhu AA, 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.