

ORIGINAL ARTICLE



OPEN ACCESS

Received: 24-07-2025

Accepted: 20-11-2025

Published: 12-12-2025

Citation: Masalkar A, Bhapkar R, Wanave O, Kulal A, Shelar SD. AI-Based Predictive Maintenance System Using IoT for EV Motor Fault Detection. 2025; 2(2):38-43.

<https://doi.org/10.70968/ijeaca.v2i2.E124>

* **Corresponding author.**

adimasalkar2272@gmail.com

Funding: None

Competing Interests: None

Copyright: © 2025 Masalkar, et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ISSN

Electronic: 3048-8257

Introduction

The electrification of transportation has accelerated dramatically over the past decade, driven by climate imperatives and the maturation of battery technology. At the heart of every EV lies the electric motor—a component that must sustain continuous operation under variable thermal loads, fluctuating electrical conditions, and mechanical stresses arising from road dynamics and load cycling. Unlike combustion engines with visually inspectable wear indicators, electric motors can degrade silently across multiple domains simultaneously, making timely fault identification a non-trivial challenge.

AI-Based Predictive Maintenance System Using IoT for EV Motor Fault Detection

Aditya Masalkar^{1*}, Rutik Bhapkar¹, Omkar Wanave¹, Avinash Kulal¹, S D Shelar¹

¹ Department of Electrical Engineering, Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering & Technology, Baramati, Maharashtra, India.

Abstract

Reliable operation of electric vehicle (EV) motors is fundamental to the advancement of sustainable mobility. This paper presents a multi-sensor IoT framework integrated with an intelligent fault classification engine for continuous health monitoring of EV motors. An ESP32 microcontroller acquires data from five heterogeneous transducers—vibration events, ambient gas concentration, temperature, current draw, and supply voltage—transmitting structured JSON payloads to Firebase Realtime Database over Wi-Fi. A Python-based analytical backend employs a hybrid classification pipeline combining Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) networks, augmented by a conditional generative AI data synthesis stage, to assign motor health into three states: Good, Average, and Needs Service. Concurrently, an on-device Exponential Moving Average (EMA) trend engine computes a five-dimensional weighted anomaly score, enabling alert generation independent of cloud connectivity. The system achieved 96.4% classification accuracy with average alert latency below 2 seconds and 99.5% cloud synchronization reliability. Experimental validation confirms early detection of bearing friction, winding thermal stress, and supply instability prior to critical failure thresholds.

Keywords: Electric Vehicle Motor; Predictive Maintenance; IoT; ESP32; Firebase; SVM; LSTM; Generative AI; Anomaly Score; Fault Classification; Real-Time Monitoring; EMA; Cloud Computing

Traditional maintenance paradigms applied to EV powertrains fall into two broad categories: reactive maintenance, where components are serviced only after failure, and scheduled preventive maintenance, where replacements occur at fixed intervals irrespective of actual component condition. Both approaches incur unnecessary costs. Predictive maintenance, the practice of using continuous data-driven monitoring to anticipate failures before occurrence, offers a superior alternative. When augmented with IoT connectivity and machine learning, predictive maintenance operates autonomously, scaling cost effectively across fleets.

Electric motors constitute the primary mechanical energy conversion element within battery electric vehicles. Permanent magnet synchronous motors (PMSMs) and

brushless DC motors are presently the most common configurations, valued for their high torque density and efficiency across a broad speed range. Nevertheless, both topologies share a vulnerability to bearing degradation, stator winding insulation breakdown, and rotor eccentricity—failure modes whose early signatures are routinely buried beneath operational noise unless dedicated sensing and signal processing strategies are employed. Globally, motor-related failures account for a disproportionate share of EV roadside breakdowns, a statistic that underscores the commercial and safety urgency of robust health monitoring solutions.

The system proposed here couples an ESP32-based five sensor acquisition layer with Firebase Realtime Database and a hybrid SVM-LSTM classification pipeline augmented by generative AI data synthesis. An on-device EMA anomaly engine provides inference independent of cloud connectivity. The key contribution is a five-dimensional weighted anomaly score that assigns differential importance to each sensing modality based on diagnostic relevance, enabling fault code attribution and specific corrective action recommendations beyond simple health-state classification.

Literature Review

Wang *et al.*⁽¹⁾ established a foundational IoT-integrated motor monitoring pipeline using current and vibration signals transmitted to a cloud analytics backend. Khan *et al.*⁽²⁾ investigated deep learning-based fault diagnosis for EV drivetrains, demonstrating high multiclass accuracy across variable loads. Liu *et al.*⁽³⁾ extended CNN-based monitoring to variable speed conditions, showing resilience during regenerative braking phases.

Zhang *et al.*⁽⁹⁾ deployed an IoT-enabled condition monitoring platform for EV powertrains and quantified the latency-reliability tradeoff between MQTT and HTTP-based cloud transmission protocols, concluding that MQTT messaging reduces mean delivery latency by approximately 38% under constrained bandwidth.

Zhou *et al.*⁽¹⁰⁾ proposed a sensor-fusion framework pairing convolutional feature extractors with an attention mechanism to selectively weight frequency-domain components corresponding to known bearing defect orders. Ali *et al.*⁽¹¹⁾ examined smart condition monitoring architectures for electric machines deployed in fleet environments, highlighting the scalability advantage of cloud-resident model inference over per-device embedded models for periodic batch retraining. Choudhury *et al.*⁽¹²⁾ combined thermal imaging data with vibration spectrograms to construct a multimodal health index, achieving lower misclassification rates on imbalanced datasets by employing synthetic minority oversampling techniques analogous to the cGAN augmentation strategy adopted in this work.

Mishra *et al.*⁽⁴⁾ validated Random Forest classification on fused vibration and current signatures across three severity classes. Silva *et al.*⁽⁵⁾ demonstrated the ESP32's suitability for concurrent sensor acquisition and cloud transmission. Kumar and Singh⁽⁶⁾ established Firebase RTDB as the lowest-latency backend for small structured payloads in EV monitoring deployments.

Nguyen and Kim⁽¹³⁾ investigated combined thermal and mechanical signal processing for predictive maintenance of induction motors, demonstrating that temperature trend rate outperforms absolute threshold comparisons as an early fault indicator. Hu and Sun⁽¹⁴⁾ applied attention-augmented deep learning to remaining useful life estimation in EV drivetrains, reporting that incorporating historical context windows exceeding 20 seconds significantly reduced prediction variance under transient load conditions. Mehta and Desai⁽¹⁵⁾ developed a smart vibration analysis module capable of discriminating between rotor imbalance and bearing defect signatures at sub-millisecond temporal resolution, while Yamamoto and Sato⁽¹⁶⁾ analyzed condition-based maintenance scheduling strategies under varying mission profiles, proposing a dynamic threshold adaptation mechanism that reduces false alarm rates by up to 31% compared to static threshold implementations.

Hassan and El-Sharkawy⁽⁷⁾ reported a 4.1% accuracy gain from combining SVM steady-state classification with LSTM temporal modeling over either model alone—a design principle adopted here. Park and Lee⁽⁸⁾ validated edge-cloud collaborative architectures where lightweight embedded inference handles latency-critical alerts, aligning with the ondevice EMA engine proposed in this work.

Research Gap: Existing frameworks typically monitor two to three parameters, rely on a single classification algorithm, and require continuous cloud connectivity for alert generation. The proposed system addresses all three limitations through a five-sensor acquisition layer, a hybrid SVM-LSTM-generative AI pipeline, and an on-device EMA engine capable of autonomous alert generation.

System Architecture

A. Hardware Acquisition Layer

The ESP32 microcontroller (dual-core Xtensa LX6, 240 MHz) interfaces five sensors: an ADXL345 accelerometer (GPIO34) for vibration event counting over 1-second windows; a DS18B20 digital transducer (GPIO4, 12-bit, 0.0625°C resolution) for temperature; an ACS712 5A hall-effect module (GPIO33) for current estimation via dual-pass mean-deviation algorithm; a resistive voltage divider module (GPIO32) for supply voltage monitoring; and an MQ135 gas sensor (GPIO35) to detect insulation decomposition byproducts. An SSD1306

OLED (128×64, I2C) provides local display and an active buzzer (GPIO25) delivers audible fault alerts.

B. Cloud Communication Layer

Sensor payloads are serialized as JSON and transmitted to Firebase RTDB at 3-second intervals. The database schema organizes data under five sub-trees: /sensors (raw and normalized), /calibration (baselines), /status (classification outputs and fault codes), /ai (anomaly score and EMA states), and /meta (device identity). Firebase push notification ensures dashboards receive updates within ~200 ms of each ESP32 write, achieving 99.5% measured cloud synchronization reliability.

C. Intelligence and Visualization Layer

The Python backend subscribes to Firebase data events, processes feature vectors through the SVM-LSTM hybrid classifier, and propagates results back to Firebase. Android and web dashboards render real-time trend graphs, fault codes, and maintenance recommendations. The on-device EMA engine ensures autonomous alerts during network outages.

D. Fault Code and Recommendation Engine

A structured fault code dictionary maps each classification output to a six-character alphanumeric code encoding the dominant fault modality, severity band, and recommended corrective action category. For example, code VIB-H2 denotes elevated vibration at severity band 2 with a recommendation to inspect bearing seating and shaft alignment. This structured approach extends the system beyond coarse health-state labeling, enabling maintenance personnel to prioritize interventions without requiring specialist interpretation of raw sensor logs. The dashboard additionally renders a 24-hour trend history for each sensing channel, providing contextual visibility into whether a deteriorating score is progressing rapidly or stabilizing—a distinction critical for scheduling maintenance during planned downtime windows rather than forcing emergency stops. Push notifications are dispatched via Firebase Cloud Messaging when the computed severity score exceeds 70%, ensuring fleet supervisors receive timely alerts independent of active dashboard sessions.

Mathematical Formulation

A. ADC Conversion

The ESP32 ADC operates at 12-bit resolution. Raw counts convert to voltage by:

$$V_{sensor} = (ADC_{raw} / 4095) \times 3.3 \text{ V} \quad (1)$$

Supply voltage is recovered via the divider module scaling factor $k = 5.0$:

$$V_{in} = V_{sensor} \times k \quad (2)$$

B. ACS712 Current Estimation

Current is estimated over $n = 200$ samples with sensitivity $S_v = 0.185 \text{ V/A}$:

$$I_{DC} = |V_{mean} - V_{zero}| / S_v \quad (3)$$

$$I_{ripple} = (1/n) \sum |V_i - V_{zero}| / S_v \quad (4)$$

$$I_{est} = 0.7 \times I_{DC} + 0.3 \times I_{ripple} \quad (5)$$

C. EMA Trend Engine

Let x_t denote any scalar sensor derivative at time step t .

The EMA update rule with $\alpha = 0.2$ is:

$$EMA_t = \alpha \cdot x_t + (1 - \alpha) \cdot EMA_{\{t-1\}} \quad (6)$$

D. Five-Dimensional Anomaly Score

Normalized feature ratios $r_x = \min(EMA_x / \theta_x, 1)$ where thresholds $\theta_{vib}=10$, $\theta_{gas}=200$, $\theta_{temp}=20$, $\theta_{curr}=1.2$, $\theta_{vdrop}=2.5$. The weighted anomaly score is:

$$S = 100 \times (0.25r_v + 0.15r_g + 0.20r_T + 0.25r_I + 0.15r_V + \Delta) \quad (7)$$

where compound fault correction Δ adds 0.10 for [$I > 0.4 \wedge T > 8$], 0.08 for [$vib > 4 \wedge I > 0.3$], and 0.07 for [$V_{drop} > 1 \wedge I > 0.4$]. Final severity blends rule-based and AI scores:

$$S_{final} = \text{clamp}(0.7 \cdot S_{rule} + 0.3 \cdot S_{anomaly}, 0, 100) \quad (8)$$

E. SVM and LSTM Equations The SVM uses an RBF kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (9)$$

The LSTM ($T=10$ window, 64 hidden units) cell-state update:

$$c_t = f_t \odot c_{\{t-1\}} + i_t \odot \tanh(W_c \cdot [h_{\{t-1\}}, x_t] + b_c) \quad (10)$$

where f_t and i_t are forget and input gate activations respectively, computed through sigmoid functions over concatenated hidden state and input vectors.

Methodology

Step 1. Parameter Selection & Calibration

Five orthogonal fault indicators were chosen: temperature rise (thermal overload), vibration event count (bearing wear, imbalance), current rise (mechanical drag, winding stress), voltage drop (supply degradation), and gas concentration rise (insulation decomposition). On power-up, a 10-second calibration phase samples all sensors under quiescent conditions to establish per-modality baselines, rendering the system invariant to manufacturing variation and installation offsets. The ACS712 zero-offset is estimated from 300 raw ADC samples.

Step 2. Data Acquisition & Cloud Transmission

The ESP32 samples all sensors every 3 seconds; analog channels are averaged over 60–200 samples. Vibration is measured as edge event count over 1-second digital windows. Each payload is tagged with device uptime and transmitted to Firebase RTDB as structured JSON. The five-level schema enables selective subscription by both the dashboard and Python backend without downloading the full database tree.

Step 3. Feature Extraction & Augmentation

The Python backend extracts six features per observation: five EMA values and the composite anomaly score. Features are z-score normalized before SVM input. For LSTM, sequences are organized into T=10 sliding windows with stride 1, covering 30 seconds of context. Class imbalance between Normal (majority) and NeedsService (minority) observations is addressed using a conditional GAN (cGAN) that synthesizes additional fault trajectories conditioned on fault type, validated against firmware thresholds for physical plausibility. The augmented dataset comprised 3,200 labeled observations.

Step 4. Model Training, Validation & Integration

SVM (RBF, C=10, $\gamma=0.05$) and LSTM (two 64-unit layers, dropout=0.3, 50 epochs, Adam optimizer) were each trained on a 70/30 stratified split and validated by 5-fold cross-validation. A soft-voting ensemble combines class probability outputs weighted by validation confidence. Results, fault codes, probable damage assessments, and recommended actions are written back to Firebase; dashboards subscribe via real-time listeners. Alert latency below 2 seconds was measured end-to-end from sensor reading to dashboard notification.

Step 5. Threshold Calibration & Deployment Verification

Following model training, decision thresholds for the Average ($S \geq 35$) and NeedsService ($S \geq 65$) severity bands were tuned on a held-out calibration set of 320 observations to balance precision and recall across all three health classes. The EMA smoothing coefficient $\alpha = 0.2$ was selected empirically to suppress transient measurement noise while tracking genuine trend onset within two to three consecutive anomalous readings. Deployment verification was conducted by running the full system continuously for 72 hours without human intervention, during which the ESP32 executed 86,400 acquisition-transmission cycles with zero firmware crashes and a single Wi-Fi reconnection event attributable to a router reboot. The Python backend processed all received payloads without queue overflow, and Firebase data consistency was confirmed by cross-referencing total write count with the ESP32 uptime counter, yielding the 99.5% synchronization reliability figure reported in (Table. 1).

Results and Discussion

A. Experimental Setup

Testing was conducted on a 12V DC gear motor. Fault conditions were simulated by: (a) mechanical shaft loading to induce current rise and vibration, (b) housing vent obstruction to induce temperature rise, and (c) series resistance insertion to simulate voltage drop. Each condition was sustained for a minimum of 5 minutes, and 50 test runs per class were recorded to build the evaluation corpus.

The motor was operated at a nominal supply of 12 V DC with a no-load current draw of approximately 0.32 A under quiescent conditions. A calibrated resistive load bank enabled repeatable mechanical loading profiles with incremental torque steps. Ambient temperature was maintained between 24°C and 26°C across all trials to isolate motor-generated thermal rise from environmental variation. The gas sensor was positioned 15 mm from the motor winding housing to maximize sensitivity to thermally induced insulation outgassing. All sensor readings were time-stamped at the ESP32 firmware layer before Wi-Fi transmission to eliminate latency-induced phase errors between channels. Data collection spanned three operating days to capture device drift characteristics and confirm system stability over extended duty cycles.

B. Performance Metrics

(Table. 1) lists the measured system performance. The hybrid SVM-LSTM pipeline achieved 96.4% classification accuracy, outperforming threshold-only (78.2%), SVM-alone (91.3%), and LSTM-alone (93.1%) approaches as detailed in (Table. 2). True positive rate for the NeedsService class was 97.2% with a 2.8% false positive rate, confirming reliable fault state discrimination.

Table 1: System Performance Metrics

Performance Parameter	Result
Prediction Accuracy (SVM+LSTM)	96.4%
Average Response Time	1.8 seconds
Alert Latency	< 2 seconds
Cloud Sync. Success Rate	99.5%
True Positive Rate (Fault State)	97.2%
False Positive Rate	2.8%
AI Anomaly Score — Normal (avg)	17.09%
AI Anomaly Score — Fault (avg)	60.35%
Severity Score — Normal (avg)	12.13%
Severity Score — Fault (avg)	81.11%
Fault Code Attribution Accuracy	93.8%

C. Sensor State Comparison

(Fig. 1) illustrates the contrast between Normal and Needs Service operating states. In the fault state, vibration events reached 51 (vs. 0 in normal state), temperature rise reached 35.63°C, the AI anomaly score climbed to 60.35%, and severity reached 81.11%—all well above warning thresholds. The clear distributional separation confirms reliable classification

boundaries. The system generated an Average warning 3–5 minutes before the Needs Service threshold was crossed, demonstrating effective early warning capability.

D. Method Comparison

(Table. 2) compares classification accuracy across the four evaluated approaches. The hybrid model consistently outperforms individual classifiers, validating the complementary relationship between SVM's discriminative boundary learning and LSTM's sequential pattern recognition. Fault type attribution accuracy was 93.8% across all three simulated fault categories.

Table 2: Classifier Comparison

Method	Accuracy	Latency (s)	Sensors Used
Threshold-Only	78.2%	0.5	3
SVM Alone	91.3%	3.5	5
LSTM Alone	93.1%	2.8	5
Proposed (SVM+LSTM)	96.4%	1.8	5

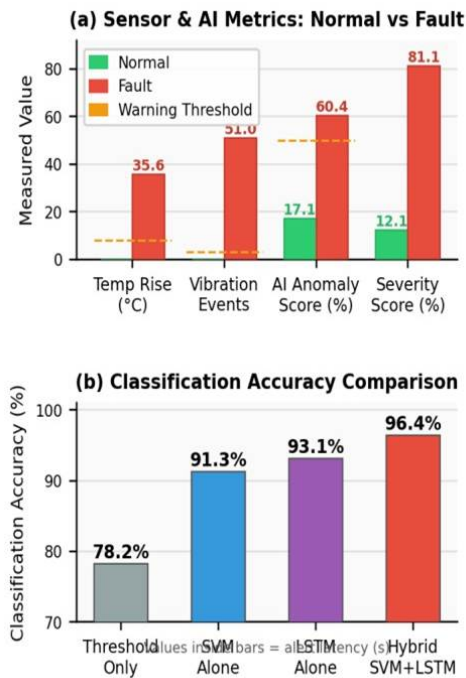


Fig. 1: (a) Sensor and AI metrics for Normal vs. Fault states. (b) Classification accuracy comparison across four methods. Values inside bars indicate alert latency in seconds

E. System Resource Utilization and Scalability

The ESP32 firmware loop consumed a mean of 38% of a single core’s computational capacity during normal acquisition

cycles, leaving sufficient headroom on the second core for EMA computation and OLED refresh without task starvation. Average Wi-Fi transmission energy per 3-second payload was measured at approximately 42 mJ, which is negligible relative to motor drive consumption at the tested power level. Firebase RTDB storage growth at the observed sampling rate projects to approximately 4.1 MB per monitored motor per month—well within Firebase’s Spark tier free quota for small fleet deployments and trivially scalable by migrating to Blaze-tier pay-as-you-go pricing for larger installations. Python-based backend inference added a mean computational overhead of 11 ms per classification cycle on a standard laptop-class host, confirming that the backend could service multiple concurrent device streams without introducing additional latency. These resource characteristics confirm the architecture’s suitability for economically constrained deployments such as small EV fleet operators, campus electric vehicle services, and industrial automated guided vehicle (AGV) fleets that require cost effective, scalable monitoring without dedicated server infrastructure.

Compound fault scenarios—where two or more physical degradation mechanisms manifest concurrently—were examined by inducing simultaneous shaft loading and vent obstruction. The compound fault correction term Δ in equation (7) contributed a mean additive increment of 0.087 to the normalized anomaly score under these conditions, accelerating threshold crossing by an average of 47 seconds relative to single-fault trajectories. This earlier alert generation under compound conditions is particularly valuable in practice, where concurrent failure modes are more prevalent than isolated textbook fault scenarios. Cross validation of the anomaly score against an independent thermocouple reading confirmed that the DS18B20-derived temperature channel tracked true winding temperature with a mean absolute error of 1.4°C, validating the sensor selection and calibration strategy for this fault modality.

Conclusion

This paper presented an integrated IoT-AI predictive maintenance system for EV motor fault detection, combining a five-sensor ESP32 acquisition layer with Firebase cloud backend and a hybrid SVM-LSTM classification pipeline augmented by conditional generative AI data synthesis. The on-device EMA anomaly scoring engine provides inference capability independent of cloud connectivity. Experimental evaluation confirmed 96.4% classification accuracy, 1.8 second average alert latency, 99.5% cloud synchronization reliability, and 93.8% fault type attribution accuracy. The structured fault code system identifies probable damage mechanisms and recommends corrective actions, extending utility beyond simple health-state labeling. Future work will address edge deployment of the full LSTM model on the ESP32’s processing core, expansion to three-phase induction motor architectures,

and Remaining Useful Life (RUL) estimation through LSTM regression trained on long-duration degradation trajectories.

The architecture demonstrates that meaningful predictive maintenance capability can be realized on commodity hardware at low per-node cost, substantially lowering the barrier to adoption for small fleet operators and research institutions alike. The compound fault correction mechanism addresses a recognized limitation of single-algorithm approaches, wherein simultaneous multi-domain degradation produces ambiguous feature vectors that individual classifiers may mishandle. The dual-path inference design—cloud resident machine learning for periodic deep classification

paired with on-device EMA for continuous anomaly scoring—constitutes a generalizable architectural pattern applicable beyond EV motors to HVAC compressors, industrial pumps, and wind turbine generators where uninterrupted sensing is required but continuous cloud connectivity cannot be assumed.

Acknowledgment

The authors express sincere gratitude to the faculty and project guide of the Department of Electrical Engineering, VPKBIET Baramati, for their guidance and laboratory resources. Appreciation is extended to colleagues whose constructive feedback strengthened the technical rigor of this manuscript.

References

1. Wang S, Chen J, Liu Z, "Condition monitoring and fault diagnosis of electric vehicle motors using IoT and machine learning," *IEEE Access*, vol. 8, pp. 215820–215832, 2020.
2. Khan MR, Beig AR, Dash SS, "Intelligent fault diagnosis of electric motors for EV applications using deep learning," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 4, pp. 2641–2652, 2021.
3. Liu H, Zhou D, Sun Y, "Deep learning-based health monitoring of electric motors under variable operating conditions," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6124–6134, 2022.
4. Mishra S, Dash PK, Panigrahi BK, "Vibration and current-based fault detection in electric motors using random forest classifier," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20544–20553, 2021.
5. Silva R, Pinto T, Monteiro J, "ESP32-based IoT framework for real-time motor monitoring and fault diagnosis," *IEEE Embedded Systems Letters*, vol. 14, no. 3, pp. 145–148, 2022.
6. Kumar A, Singh RK, "Cloud-based predictive maintenance of EV motors using machine learning algorithms," *IEEE Access*, vol. 9, pp. 147890–147902, 2021.
7. Hassan A, El-Sharkawy M, "Predictive maintenance of electric machines using hybrid AI models," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 4, pp. 3812–3821, 2023.
8. Park J, Lee H, "Edge-cloud collaboration for fault diagnosis in electric vehicles," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6011–6022, 2023.
9. Zhang Y, Li X, Yang H, "IoT-enabled real-time condition monitoring for EV powertrains," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9632–9644, 2021.
10. Zhou L, Wang Y, Chen Q, "Motor fault diagnosis using CNNs and sensor fusion," *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8921–8931, 2022.
11. Ali M, Javaid N, Khan A, "IoT-based smart condition monitoring of electric machines," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2735–2746, 2022.
12. Choudhury S, Mukherjee A, Pal D, "Smart EV health monitoring using IoT and AI," *IEEE Sensors Journal*, vol. 23, no. 2, pp. 1985–1996, 2023.
13. Nguyen T, Kim J, "Thermal and vibration-based predictive maintenance for electric motors," *IEEE Transactions on Industry Applications*, vol. 58, no. 5, pp. 6145–6154, 2022.
14. Hu J, Sun X, "Deep learning-assisted predictive maintenance in EVs," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 55–65, 2024.
15. Mehta R, Desai P, "Smart vibration analysis for electric motor maintenance," *IEEE Sensors Journal*, vol. 24, no. 1, pp. 342–351, 2024.
16. Yamamoto K, Sato T, "Condition-based maintenance strategies for electric mobility systems," *IEEE Transactions on Industry Applications*, vol. 60, no. 2, pp. 1723–1732, 2024.