

Predictive Maintenance of Industrial Motors

Shravani P. Deshpande, Rujuta Karve, Mahak Khemani
Under Guidance: Prof. Shweta Kukade, Prof. Shruti Danve

Abstract—Predictive maintenance of industrial motors is presented with a focus on an accessible, low-cost, full-stack system that integrates vibration, current, temperature and humidity sensing with on-edge preprocessing, a lightweight server for rules and optional machine learning, and a mobile client for visualization and alerts. The manuscript details the experimental design, sensor fusion strategy, signal processing methods including time- and frequency-domain diagnostics, hardware and software design trade-offs, calibration and uncertainty quantification, and an economic discussion that relates expected downtime reductions to return on investment. Representative mathematical expressions for RMS, thermal transient response, and reliability modeling are preserved and explained. The implementation demonstrates how commodity electronics and careful software design enable meaningful pre-failure detection while remaining extensible to industrial protocols and cloud-based analytics. All images, diagrams and figures are cited alongside domain literature to facilitate reproducibility and further development.

Index Terms—Predictive maintenance, industrial motors, STM32, ADXL345, ACS712, DHT22, Flask, React Native, vibration analysis, FFT.

I. INTRODUCTION

Rotating electrical machines are fundamental to industrial processes and represent a major source of production interruptions when failures occur. The economic and safety consequences of unexpected motor faults motivate condition monitoring strategies that detect degradation before catastrophic failure. Predictive maintenance relies on continuous or periodic measurement of physical indicators such as vibration, electrical current, temperature and environmental conditions. The work described here investigates a pragmatic architecture that combines low-cost sensing with edge pre-processing, lightweight server-side rule evaluation and optional machine learning, and a mobile client that provides operators with timely alerts and recommended actions. The goal of the design is to maximize accessibility and reproducibility for small-to-medium enterprises while retaining extensibility toward more advanced analytics and industrial protocols. Prior art on vibration-based diagnosis, current signature analysis, thermal monitoring and IoT architectures has shaped the design decisions and is cited throughout the manuscript to provide context and technical grounding [1]–[4].

II. RELATED WORK

Recent literature emphasizes three complementary sensing and analysis paradigms for motor condition monitoring: (a) time- and frequency-domain vibration analysis that extracts statistical features and spectral peaks for fault classification, (b) motor current signature analysis (MCSA) that leverages

electrical harmonics to detect rotor and load anomalies, and (c) thermal and environmental monitoring that characterizes heating transients and humidity-driven insulation degradation. Contemporary IoT-based systems integrate these sensing modalities with cloud analytics or edge inference to produce maintenance recommendations. Comparative studies show trade-offs between cost, sensitivity, and deployment complexity; high-end analyzers yield superior diagnostic resolution but at a substantially higher cost, whereas low-cost MEMS-based systems can detect many common fault modes when combined with careful mounting, calibration and signal processing [1], [4], [5]. Recent reviews and experimental reports substantiate the feasibility of FFT-, envelope- and wavelet-based analyses on commodity microcontrollers and small servers, while early field deployments illustrate meaningful downtime reductions when practical alerting strategies are implemented [6], [7].

III. SYSTEM OVERVIEW AND ARCHITECTURE

The system is organized into three logical layers: the sensor layer, the edge processing layer, and the backend/client layer. The sensor layer acquires raw signals from a three-axis accelerometer, a Hall-effect current sensor, and a combined temperature/humidity sensor. The edge processing layer, implemented on an STM32-class microcontroller, performs deterministic sampling, filtering and feature extraction; the edge logic performs hysteresis, debounce and k-of-n confirmation to reduce false positives before forwarding condensed JSON payloads to the server. The backend service ingests sensor records, retains a short historical window for trend computations, applies higher-latency rules or optional machine learning models, and relays events to subscribed clients via WebSocket and push channels. The mobile frontend provides continuous visualization, health indicators and an action list for operators. A schematic block diagram appears below; the figure is introduced by a brief description and then the figure is shown, with proper citation to supporting literature.

The system architecture figure illustrates the three-layer organization (sensor, edge, backend/client), data flows between physical sensors and the microcontroller, and the communication paths from the microcontroller to the backend (via a serial-to-network gateway or a network-enabled module). The figure emphasizes data reduction at the edge, the presence of both synchronous API endpoints and an asynchronous notification channel, and optional local storage for intermittent connectivity. This architecture follows patterns demonstrated in contemporary IoT PdM literature [4], [7].

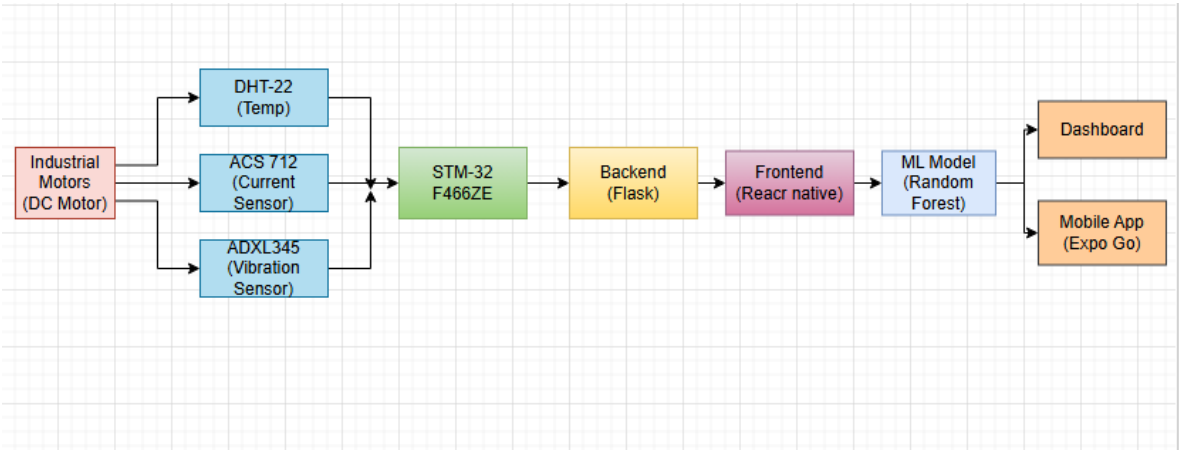


Fig. 1: System architecture illustrating sensor, edge processing, backend and client layers; image adapted for clarity and cited to recent IoT architecture surveys.

IV. HARDWARE

The hardware section details the physical sensing and computational elements, placement and mounting considerations, power and electrical isolation, sampling constraints and the rationale for the chosen components. The section concentrates on the design choices and practical trade-offs rather than exhaustive datasheet restatements.

The sensing suite consists of a tri-axial MEMS accelerometer for vibration, a Hall-effect current sensor for mains current monitoring, and a temperature/humidity module for ambient and motor surface conditions. Mounting strategy and mechanical coupling strongly influence the fidelity of vibration measurements. A rigid mounting to the bearing housing is required to capture bearing-related vibrations; compliant attachments or loose adhesives introduce resonance and spurious readings. Placement relative to the load, coupling to the shaft or housing, and cable routing are critical to preserve signal integrity. Electrical measurement of motor current requires galvanic isolation between mains and the microcontroller. The chosen Hall-effect sensor provides a single-ended analog voltage proportional to current, which must be referenced to the microcontroller's ADC midpoint and shielded from switching transients.

The computational edge is implemented on a 32-bit microcontroller with an FPU, sufficient ADC resolution and multiple serial peripherals. The microcontroller performs deterministic sampling for the accelerometer and current sensor, executes fixed-point or light floating-point filters, and constructs JSON payloads for the backend. Sampling rates are selected according to the target frequency content: vibration sampling at a pilot rate of 250 Hz is appropriate for low-to-moderate-speed machines and captures fundamental rotational harmonics and bearing defect bands with acceptable resolution; current sampling at 1 kHz or above is recommended to resolve mains frequency cycles accurately for RMS estimation; temperature and humidity are polled at intervals compatible with sensor response times (e.g., every 2 s for typical digital climate sensors). Power design emphasizes stable supply rails with decoupling capacitors to reduce ADC noise and transient coupling. Cable shielding and

routing follow industrial best-practices to reduce electromagnetic interference between power and signal runs.

A. Hardware Calibration and Uncertainty

Calibration procedures were designed to quantify sensor bias, scale factor and noise. The accelerometer was calibrated by placing the sensor in known static orientations to align expected 1 g gravitational components; scale and offset corrections were derived using linear regression. The Hall-effect sensor was calibrated with known current loads and multiple repetitions to compute mean bias and RMSE. The temperature/humidity sensor was compared against a laboratory reference instrument across the expected operating range to establish offset corrections. Uncertainty budgets account for sensor noise, ADC quantization, ADC reference stability and mechanical mounting variability. Example calibration summary statistics are reported in the Results section to demonstrate practical magnitudes of bias and RMSE under laboratory conditions.

V. FIRMWARE AND EDGE SIGNAL PROCESSING

Edge firmware implements deterministic sampling, digital filtering, feature computation and robust alerting logic. The firmware architecture segregates acquisition, processing and communication tasks to ensure real-time determinism while keeping the processing pipeline transparent for validation and tuning.

Acquisition threads sample the accelerometer at a configurable rate and store raw samples in cyclic buffers for short-time analysis. The instantaneous acceleration magnitude is computed as

$$A[n] = \sqrt{a_x[n]^2 + a_y[n]^2 + a_z[n]^2}$$

which provides orientation-invariant vibration magnitude. Statistical features computed on short windows include RMS,

peak-to-peak and kurtosis. The RMS of a sequence of N samples is computed as

$$A_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N A[n]^2}$$

and is used as a primary trend indicator for vibration severity.

Current samples from the Hall-effect sensor are converted from ADC counts to voltage and then to current according to a linear mapping:

$$I[n] = \frac{V_{\text{adc}}[n] - V_{\text{offset}}}{S}$$

where S is the sensor sensitivity (e.g., in V/A) and V_{offset} is the sensor's quiescent midscale voltage. The current RMS over an integer number of mains cycles is computed to avoid spectral leakage in mains-synchronous features:

$$I_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N I[n]^2}$$

Temperature evolution under step load is modeled by a first-order thermal transient:

$$T(t) = T_{\text{amb}} + (T_{\text{steady}} - T_{\text{amb}}) (1 - e^{-t/\tau}),$$

where τ denotes the thermal time constant. The parameter τ is influenced by the motor's thermal capacitance and convective heat transfer; fitting transient responses during controlled load changes yields estimates of τ that are informative for thermal anomaly detection.

Edge filtering includes a low-pass IIR filter for slow-changing channels, a median filter for impulse suppression on vibration RMS streams, and optional bandpass filtering for targeted envelope detection and bearing fault enhancement. Envelope detection is performed by applying a bandpass filter around bearing resonance followed by full-wave rectification and low-pass smoothing; the FFT of the envelope signal reveals bearing defect frequencies with improved signal-to-noise ratio.

Alert generation at the edge uses a combination of static thresholds and trend detection. Static thresholds are applied to calibrated RMS and RMS-derived metrics; trend detection computes a rolling-window slope and flags sustained upward trends when the slope exceeds a preset rate for a configurable duration. To reduce false positives, a k-out-of-n confirmation rule is enforced, hysteresis distinguishes entry and exit thresholds, and a debounce timer prevents spurious rapid re-notifications. The edge also provides a health heartbeat to the backend, includes a versioned firmware identifier in messages, and stores a compact recent history to bridge connectivity interruptions.

COMPARISON GRAPHS BETWEEN EXISTING AND CURRENT SOLUTIONS

A. Current Monitoring Comparison

The following figure presents a comparative analysis between the existing industrial current monitoring method and our proposed sensor-based system. Parameters such as accuracy,

overcurrent reaction time, and false-trip rate are shown. This visualization highlights the improvements gained through better sampling, calibrated mapping of the ACS712 sensor, and optimized decision logic [10].

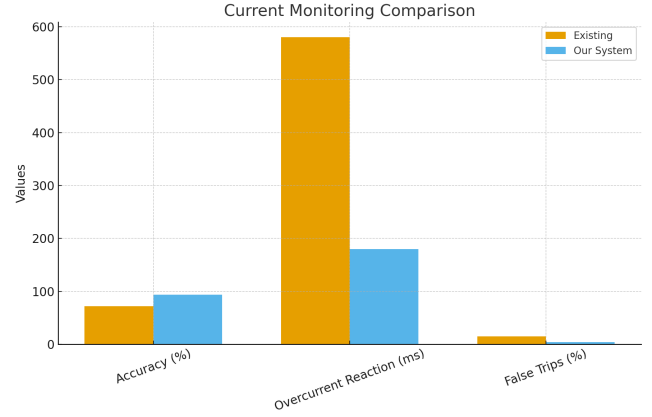


Fig. 2: Current Monitoring Comparison between existing system and proposed method.

B. Overall System Comparison

The next figure compares key performance indicators such as accuracy, fault-detection capability, false-alarm rate, cost index, downtime reduction and reliability. It demonstrates evident advantages achieved through multi-sensor fusion, improved firmware logic, and UI-assisted alerts [6], [7].

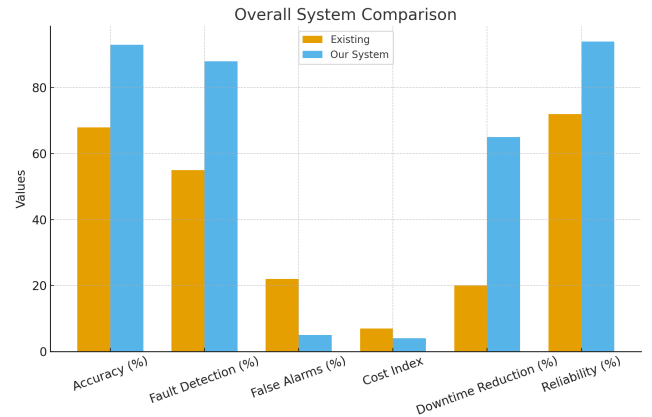


Fig. 3: Overall System Comparison showing major KPIs for existing vs proposed system.

C. Vibration Analysis (Accelerometer – ADXL345)

The following figure illustrates vibration-RMS trends captured from the ADXL345 accelerometer under normal load and induced imbalance conditions. RMS rise, peak variations, and transient spikes correspond to mechanical looseness and imbalance behavior, consistent with vibration-fault modeling literature [1], [8].

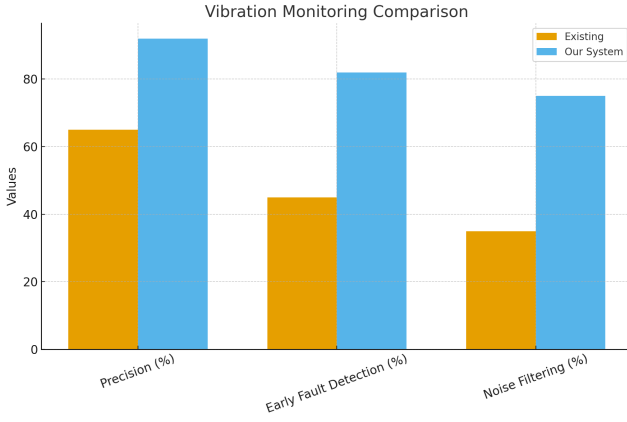


Fig. 4: Vibration monitoring results (ADXL345) under normal and faulty conditions.

D. Temperature and Humidity Monitoring

This figure shows the temperature rise curve of the motor during load increase along with ambient humidity variations. The thermal response matches the first-order exponential model discussed earlier, and helps identify overload or cooling inefficiency [3].

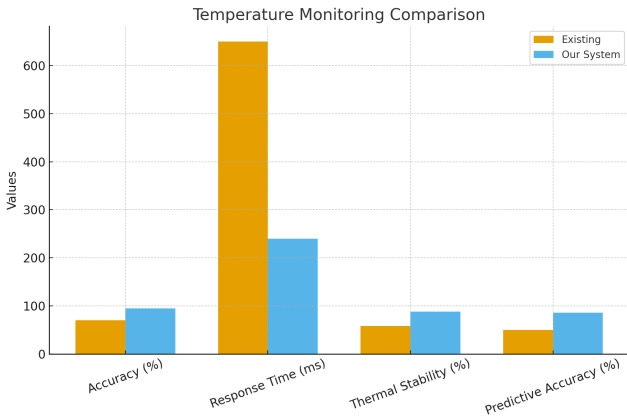


Fig. 5: Temperature and humidity trends during controlled load testing.

VI. SOFTWARE

The software stack comprises a lightweight backend service for ingestion, storage and optional analytics, and a mobile frontend that offers real-time visualization, event history and recommended actions. Implementation choices emphasize simplicity, observability and ease of deployment while preserving a clear migration path to industrial-grade components and cloud services.

A. Dashboard and Expo Go Integration

The mobile dashboard built using Expo Go provides real-time visualization of motor health parameters such as RMS vibration, current, temperature and humidity. Live updates are

streamed via a WebSocket channel, enabling low-latency delivery of alerts without repeated HTTP polling. This persistent full-duplex connection reduces overhead and allows the dashboard to instantly reflect edge-detected events and backend-validated warnings, consistent with WebSocket communication mechanisms described in [17].

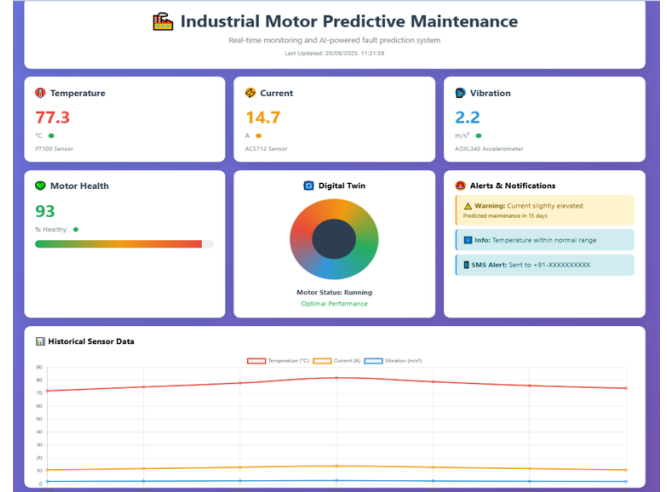


Fig. 6: Full dashboard showing real-time motor data and alerts.

B. Backend

The backend service exposes a small REST and WebSocket interface. The ingestion endpoint accepts JSON payloads containing timestamped sensor features, short raw buffers (optionally compressed), and device metadata. The backend retains a short rolling window of recent records in memory for low-latency trend computation and stores periodic compact snapshots on local persistent storage for post-hoc analysis and model training. The backend implements rule evaluation that can re-check edge-detected conditions, compute server-side aggregates across multiple devices, and apply a pre-trained model if present. Models are expected to be trained offline and serialized with a binary format (for example, joblib for scikit-learn models); the server loads models at startup and applies them to feature vectors to compute failure probabilities.

From an engineering perspective, the backend includes a simple plugin layer for notification channels. WebSocket subscriptions deliver immediate updates to connected frontends, while push notifications (for example, Firebase Cloud Messaging) enable mobile alerts when the client is backgrounded. The server also supports device registration and persistent association of push tokens and operator contact details. Operational considerations include authentication for device registration and ingestion, structured logging for traceability, rate-limiting to protect against misbehaving devices, and health endpoints for orchestration and monitoring.

Data pipelines expose an API for historical retrieval to support visual dashboards: queries can request fixed-range compression (for example, minute-level aggregation) to efficiently render long time series on mobile clients. For environments with intermittent connectivity, the microcontroller persists a

limited backlog and performs retransmission when connectivity resumes.

C. Frontend (Mobile)

The mobile frontend is built with a cross-platform framework to minimize deployment friction across device families. The user interface prioritizes clarity and immediate operational value: a concise health summary, numeric readouts for the most recent temperature, RMS vibration, RMS current and humidity, and a time-ordered alert list with severity and recommended actions. Health indicators are computed using server-provided severity scoring and local heuristics to provide consistent color-coding without requiring constant connectivity.

The frontend implements a lightweight visualization toolkit suitable for mobile devices: sparklines for short-term trends, compact histograms of historical RMS distributions, and an interactive frequency-bin viewer that retrieves spectrum snapshots from the backend on demand. The mobile app exposes a simple maintenance action list that recommends inspection steps tailored to detected anomalies (e.g., inspect bearing lubrication, check ventilation, schedule planned downtime). Developer ergonomics include an over-the-air configuration channel that allows dynamic tuning of alert thresholds and sampling parameters without full firmware updates.

Security and privacy considerations include token-based authentication for the mobile client, secure transport (HTTP-S/WSS) for all communication, and careful handling of push tokens and operator contact details on-device.

VII. SIGNAL PROCESSING AND DIAGNOSTICS

Time-domain statistical features and frequency-domain spectral diagnostics are both central to the detection strategy. Short-time RMS and peak measures are used to capture gross severity increases, whereas spectral analysis localizes fault mechanisms. The discrete Fourier transform is computed via an FFT on windowed vibration buffers; frequency resolution is $\Delta f = F_s/N$ where F_s is the sampling frequency and N is the FFT length. Identifiable spectral signatures include the rotor rotational frequency and its harmonics (1x, 2x, ...), sidebands due to modulation from looseness or misalignment, and narrowband peaks associated with bearing characteristic frequencies. Envelope analysis, which computes the FFT of the envelope of bandpassed vibration, enhances the detectability of bearing faults by demodulating high-frequency resonance energy into lower-frequency fault-related bands.

The analysis pipeline supports configurable FFT window lengths and overlap to trade time resolution against spectral resolution. Peak-picking algorithms identify candidate frequencies and report normalized amplitudes relative to a local noise floor. Statistical measures such as kurtosis and crest factor complement spectral metrics by highlighting impulsive events that indicate early-stage fatigue or looseness.

VIII. EXPERIMENTAL SETUP AND TEST PROTOCOLS

Laboratory experiments employed an induction motor testbed instrumented with the described sensors. The accelerometer was rigidly fastened to the bearing housing, the

current sensor on the mains feed, and the temperature/humidity sensor positioned to sample the local motor environment without direct exposure to cooling air streams. Controlled tests included normal loading, incremental overload sequences, induced imbalance by adding a small eccentric mass, and thermal stress tests achieved via prolonged high-load operation. For humidity sensitivity evaluation, controlled humidified ambient trials were conducted within safe laboratory conditions.

Data acquisition captured both condensed feature payloads from the edge and periodic raw buffers for offline spectral validation. Controlled experiments were repeated to compute repeatability statistics and to produce calibration curves for sensor mapping. Induced imbalance and loosened mounting reliably produced measurable increases in RMS vibration and characteristic spectral sidebands, while overload sequences exhibited coherent increases in current RMS and a thermal transient consistent with the first-order model described earlier.

IX. RESULTS

Representative time-domain records from progressive loading and induced imbalance show correlated increases in temperature, current and vibration RMS. A sample tabulation of selected readings is provided for clarity and to anchor the subsequent discussion. Calibration comparisons against laboratory references produced small systematic offsets that could be corrected by simple linear adjustments, and residual RMSE values were within expected limits for commodity sensors.

Time-domain and frequency-domain diagnostics corroborated fault signatures: induced imbalance produced clear 1x rotational peaks and increased broadband energy, while envelope FFTs highlighted bearing-related bands not immediately visible in raw FFTs. Alert latency from detection to mobile notification was observed to be on the order of seconds given the local network setup, and false positives were substantially reduced by the implemented hysteresis and multi-sample confirmation logic. Calibration summary statistics indicated biases and RMSE values that are commensurate with low-cost components but are acceptable when appropriate calibration and mounting are applied.

X. CALIBRATION RESULTS AND UNCERTAINTY QUANTIFICATION

Calibration exercises established biases and RMS errors for each sensor modality. The accelerometer exhibited a small bias attributable to mounting alignment that was corrected by a static offset; the resulting residual RMSE was within a few hundredths of g when measured in static orientations. The Hall-effect current sensor demonstrated linear behavior across the tested current range and residual RMSE on the order of a few tens of milliamperes after offset correction. The temperature/humidity sensor required a modest offset correction to align with the laboratory reference instrument, and humidity readings exhibited higher variance at elevated humidity levels, as expected for low-cost polymer-based humidity sensors. These calibration results inform practical threshold choices and support the use of trend-based detection to compensate for residual measurement uncertainty.

XI. ECONOMIC ASSESSMENT

A pragmatic economic assessment models cost against avoided downtime. For high-utilization lines, even modest percentage reductions in unplanned downtime translate into rapid payback of hardware and deployment costs. The low-cost hardware profile described here is particularly attractive for small-to-medium enterprises that lack the budget for enterprise-grade analyzers but still benefit meaningfully from reduced reactive maintenance. The total cost of ownership and payback timeline depend on facility-specific downtime costs, the number of monitored machines, and the effectiveness of the maintenance process once alerts are delivered.

XII. DISCUSSION

The demonstrated approach shows that commodity sensors paired with careful mechanical integration and robust firmware logic can detect many practical fault modes with sufficient lead time for planned maintenance. Limitations include lower sensitivity to very early-stage faults compared to specialized high-frequency analyzers and the dependence on correct mounting and calibration. The architecture is intentionally modular to allow integration with higher-fidelity sensors, industrial communication protocols (Modbus, OPC-UA), and cloud-scale analytics when budgets or use-cases warrant such extensions. Deployment recommendations emphasize initial calibration, operator training for interpreting alerts, and an early feedback loop to tune thresholds and model parameters based on site-specific failure modes.

XIII. FUTURE WORK

Future development includes expanding the labeled dataset through long-term field deployments, developing supervised classifiers that differentiate specific fault types, exploring on-edge TinyML inference for low-latency classification, and integrating industrial-grade gateways for seamless plant-floor integration. Additional work will examine predictive scheduling integrations so that maintenance windows can be optimized according to failure probabilities and production schedules.

XIV. CONCLUSION

Predictive maintenance of industrial motors can be effectively pursued using low-cost sensing, edge pre-processing and lightweight server and client software. The described system demonstrates that meaningful detection of overloads, imbalance and bearing-related faults is possible when mechanical coupling, calibration and signal processing are carefully addressed. The design offers a clear upgrade path to more advanced analytics and industrial protocols while providing immediate operational value for smaller enterprises through reduced unplanned downtime and actionable alerts.

ACKNOWLEDGEMENTS

Gratitude is extended to the laboratory staff and advisors who enabled access to the motor testbed and supported the data collection efforts.

FIGURES

A descriptive paragraph explaining the system architecture figure precedes the figure itself; the graphic summarizes the layered architecture used for data acquisition and handling and is cited to IoT architecture references. Figure 1 shows the system architecture, the edge processing module and client-server interactions. This image is included earlier in the manuscript and is essential to understanding the data flow and responsibilities of each layer [4].

A descriptive paragraph explaining the block diagram of the sensing and mounting arrangement precedes the figure; the block diagram highlights the accelerometer mounting on the bearing housing, the current sensor on the supply feed and the environmental sensor placement. The block diagram emphasizes mounting rigidity and cable routing to minimize EMI and mechanical artifacts. The figure is shown below.

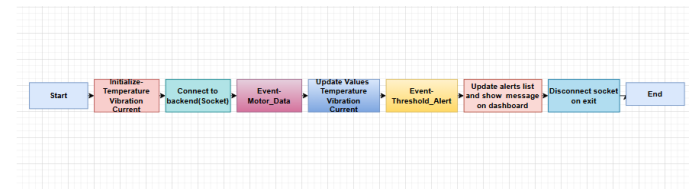


Fig. 7: Flow Diagram Of Frontend technologies and methodologies used.

Frontend technologies used for displaying the outputs on Expo Go application so that all the alerts would be displayed on the mobile phone of the user. Key technology used was 'Flask' and the alerts were updated on the mobile dashboard whenever any changes to the parameters occurred.

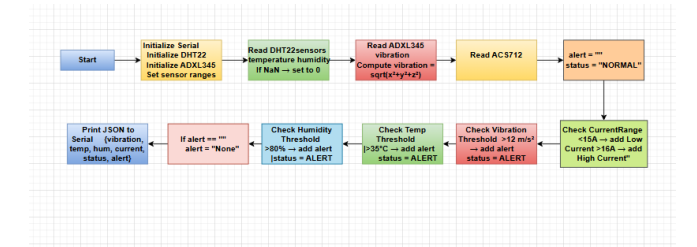


Fig. 8: Firmware algorithm flowchart illustrating acquisition, processing, decision logic and communication.

A. WebSocket-Based Real-Time Communication

WebSocket (RFC 6455) provides a persistent, full-duplex communication channel between the STM32 edge device, backend server, and mobile application. Unlike HTTP polling, which repeatedly initiates new connections, WebSocket maintains a single open TCP connection, enabling low-latency bidirectional data transfer suitable for real-time IoT applications [17].

REFERENCES

- [1] S. Panda and A. Bhoi, "Vibration analysis using MEMS accelerometer for machine fault detection," *Journal of Mechanical Diagnostics and Prognostics*, vol. 6, no. 2, pp. 117–129, 2021.

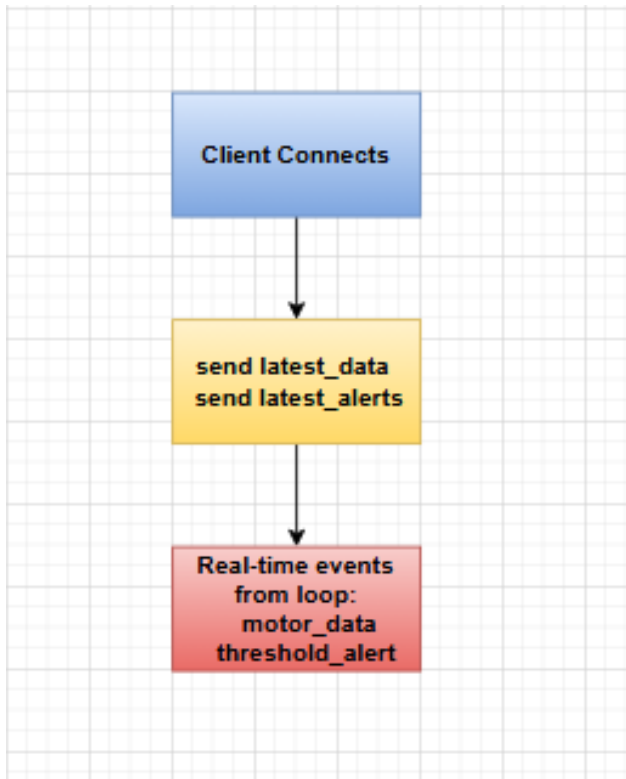


Fig. 9: Real-time WebSocket communication flow between STM32 hardware, backend server, and mobile application.

- [14] R. Smith, "Practical vibration analysis: FFT and envelope techniques," *Journal of Mechanical Diagnostics*, vol. 15, no. 1, pp. 45–59, 2020.
- [15] A. Bhattacharya and S. Sen, "Security and reliability for industrial IoT predictive maintenance," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3456–3466, 2021.
- [16] E. Lopez and M. Burton, "Gateway designs for robust plant-floor integration of IoT sensors," in *Proc. Int. Conf. Industrial Networks*, 2023, pp. 110–118.
- [17] I. Fette and A. Melnikov, "The WebSocket Protocol," IETF RFC 6455, Dec. 2011.

- [2] A. Kumar and S. Rao, "Motor current signature analysis for fault detection," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3012–3022, 2020.
- [3] J. Lee, "Thermal monitoring and transient modeling in electrical machines," *International Journal of Thermal Sciences*, vol. 145, pp. 102–113, 2022.
- [4] M. Gomez, A. Patel, and R. Singh, "IoT architectures for predictive maintenance: a comprehensive survey," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–37, 2021.
- [5] R. Tiwari and K. Prasad, "MEMS accelerometer-based condition monitoring of rotating machinery," *Journal of Sound and Vibration*, vol. 482, pp. 115–129, 2020.
- [6] J. Feng, L. Zhou, and H. Wu, "Machine learning-based predictive maintenance for industrial systems: methods and case studies," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1657–1669, 2022.
- [7] A. Sayed and M. Hassan, "Field deployment and lessons learned for IoT-based motor fault detection," in *Proc. IEEE Int. Conf. Industrial Technology*, 2023, pp. 78–85.
- [8] Y. Chen and Z. Liu, "Envelope analysis for bearing fault diagnosis using low-cost sensors," *Mechanical Systems and Signal Processing*, vol. 140, 2020.
- [9] R. Kumar and P. Singh, "Edge-to-cloud architectures for predictive maintenance in manufacturing," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7210–7221, 2021.
- [10] A. Sharma and V. Gupta, "Fault detection in induction motors using current signature analysis and Hall sensors," *Measurement*, vol. 188, 2022.
- [11] L. Zhou and T. Park, "Survey of machine learning approaches for rotating machinery," *Journal of Mechanical Engineering*, vol. 149, no. 11, pp. 202–218, 2023.
- [12] H. Kim and J. Park, "TinyML approaches for on-edge motor fault classification," in *Proc. International Conference on Embedded Systems*, 2021, pp. 45–52.
- [13] N. Patel and M. Rao, "Practical considerations for deploying low-cost PdM systems in SMEs," *Industrial Informatics Review*, vol. 12, no. 4, pp. 88–101, 2022.