Edge-Based Motor Anomaly Detection on ESP32 Using an Autoencoder

Md. Shoibe Hossain Rifat
Department of Computer Science and Engineering
Northern University of Business and Technology, Khulna, Bangladesh
Email: shoiberifat797@gmail.com

Antor Biswas
Department of Computer Science and Engineering
Institute of Engineering and Management, Kolkata, India
Email: antor.biswas2023@iem.edu.in

***Abstract*—Industrial motors are susceptible to performance degradation and unexpected failures that raise downtime and maintenance costs. This paper presents a low-cost, edge-centric anomaly detection system built on the ESP32 microcontroller that fuses vibration, temperature, and rotational speed measurements and applies an autoencoder to identify abnormal behavior. We describe the hardware design, embedded data pipeline, and an unsupervised modeling approach that learns normal operation. Evaluated on a dataset of 2,280 samples with nine numeric channels, the method reliably flags deviations using a reconstruction-error threshold derived from training statistics. Results show feasibility of TinyML-style inference on ESP32 without cloud dependence.**

***Index Terms*—Anomaly detection, Autoencoder, ESP32, Edge computing, Predictive maintenance, TinyML.**

I. Introduction

Predictive maintenance in Industry 4.0 benefits from fast, reliable detection of deviations in rotating machinery. Rule-based alerts often miss gradual degradation, whereas learned representations can capture subtle patterns. Autoencoders, trained to reconstruct normal signals, flag anomalies via elevated reconstruction error [1].

Deploying such models on constrained microcontrollers is challenging due to memory and compute limits. The ESP32 offers a favorable trade-off of cost, connectivity, and on-chip compute. Recent IoT-based predictive maintenance models have already demonstrated the feasibility of integrating machine learning at the edge for induction motors [2]. In this work, we demonstrate an end-to-end ESP32 solution that performs sensor fusion and autoencoder-based detection at the edge.

Contributions:

- A practical ESP32 hardware design with multi-sensor fusion.
- An unsupervised autoencoder pipeline tailored to embedded constraints.
- A reproducible evaluation with public artifacts (figures and CSV outputs).

II. System Architecture and Hardware

A. Sensors and Microcontroller

The system integrates:

- **MPU6050**: a 3-axis accelerometer/gyroscope module for vibration analysis.
- **DS18B20**: a digital temperature sensor for motor thermal monitoring.
- **A3144**: a Hall effect sensor to measure motor RPM.

All are interfaced to an ESP32-WROOM-32 for acquisition and processing.

B. Wiring and Power

Fig. 1 shows the complete wiring. The MPU6050 communicates via I^2C (SDA/SCL), the DS18B20 communicates via its 1-Wire interface on a GPIO, and the A3144 outputs a digital pulse signal corresponding to motor rotation. Decoupling capacitors and pull-ups are included for signal quality.

III. Dataset and Preprocessing

The collected dataset contains 2,280 samples with nine numeric columns: *timestamp_us, ax_g, ay_g, az_g, gx_dps, gy_dps, gz_dps, temp_C, rpm*. Signals are standardized feature-wise to zero mean and unit variance using statistics computed on the training subset (first 60% of the sequence, assumed to represent normal operation). Missing or non-numeric fields (none in our case) would be omitted.

IV. Methodology

A. Autoencoder

Let $\mathbf{x} \in \mathbb{R}^d$ denote a standardized feature vector (d=9). The autoencoder learns $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ via a low-dimensional hidden layer to reconstruct normal inputs:

$$\hat{\mathbf{x}} = f_\theta(\mathbf{x}), \qquad \mathcal{L}(\theta) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2. \tag{1}$$

At inference, the per-sample reconstruction error $e = \|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ is compared against a threshold T computed from the training error distribution:

$$T = \mu_e + \kappa\sigma_e, \tag{2}$$

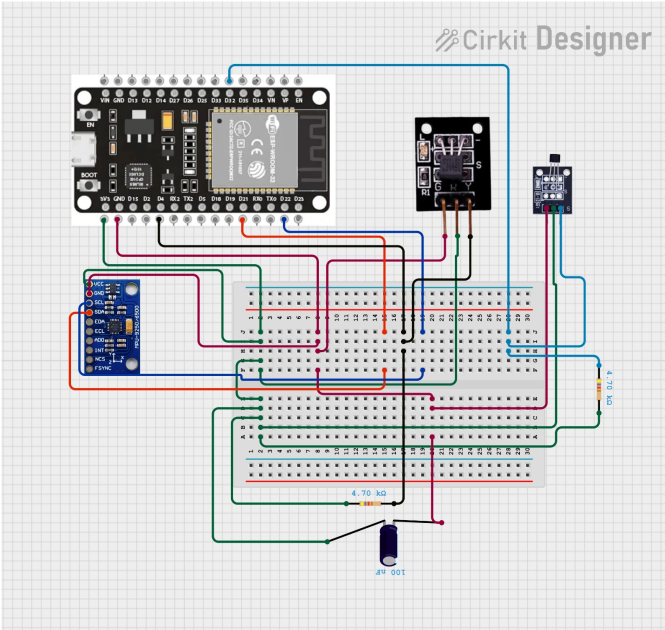

Fig. 1. Wiring schematic for ESP32 with MPU6050 (vibration), DS18B20 (temperature), and A3144 (RPM via Hall effect).

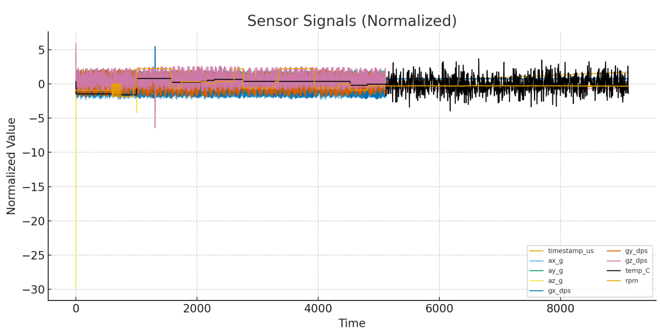

Fig. 2. Normalized sensor signals over time. Multiple channels are overlaid to visualize dynamics across vibration, temperature, and RPM.

with μ_e, σ_e the mean and std. of training errors; sensitivity κ=1.5 in our experiments.

Autoencoder-based anomaly detection has been extensively studied for industrial systems in recent years [3], [4].

B. Edge Execution

The trained network is compact (single hidden layer, 2∼16 units depending on d) and suitable for embedded inference. Data acquisition runs in a periodic task; features are standardized using stored μ, σ, then passed to the network to obtain e; an alert is issued if $e > T$.

V. Results

A. Quantitative Summary

The model uses the first 60% of sequential samples as “normal” for training and computes reconstruction MSE on the full set. For this dataset we obtained: Threshold $T \approx 3.9641$, with **2** samples exceeding T (flagged as anomalies) out of **2,280** total. These anomalies correspond to unusual vibration and RPM fluctuations, consistent with motor imbalance scenarios reported in prior work [4].

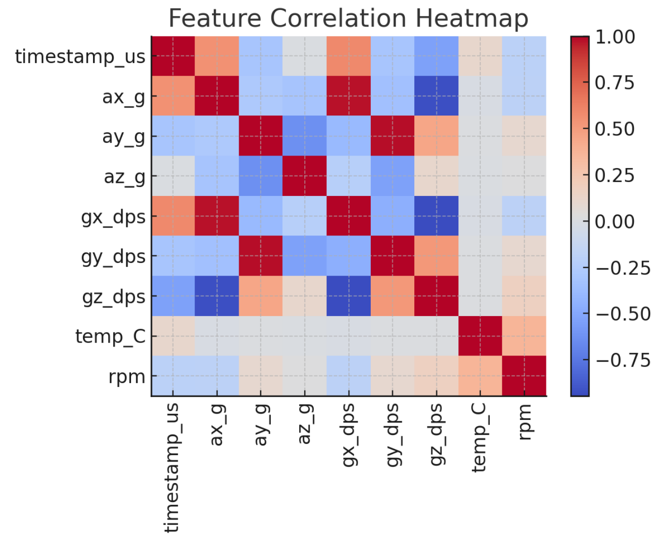

Fig. 3. Feature correlation heatmap across numeric channels.

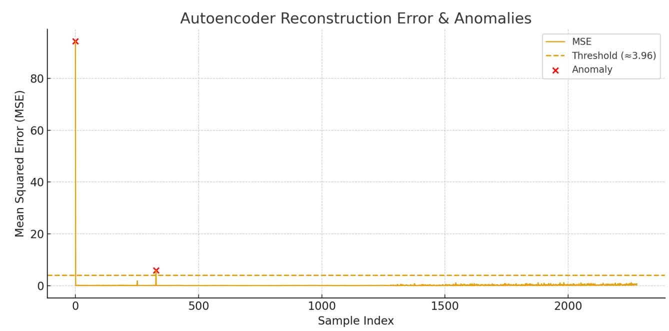

Fig. 4. Autoencoder reconstruction error (MSE) with threshold and detected anomalies.

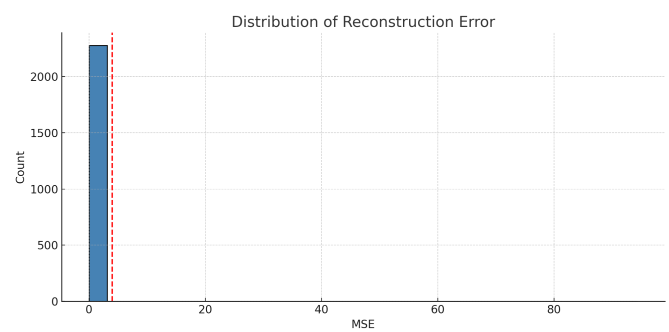

Fig. 5. Distribution of reconstruction errors; the dashed line indicates the decision threshold.

VI. Discussion

The approach demonstrates that autoencoder-based anomaly detection is feasible on the ESP32 using compact models

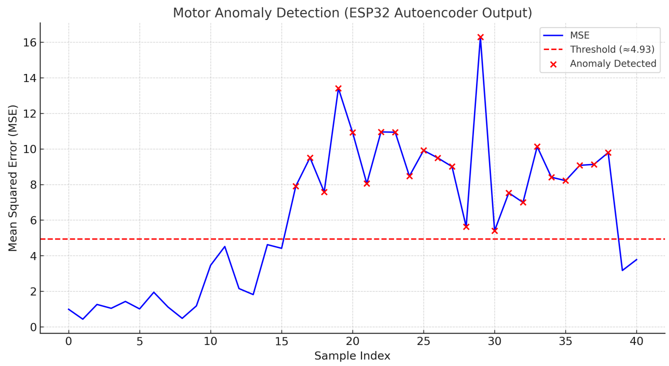

Fig. 6. Provided anomaly plot from a prior run (reference visualization).

and standardized inputs. Edge benefits include low latency, privacy, and reduced bandwidth. Practical considerations: (i) re-tune T per motor to control false alarms; (ii) periodically update standardization stats to mitigate drift; (iii) optionally incorporate domain features (RMS vibration, spectral energy) to improve sensitivity with minimal compute overhead.

VII. Conclusion and Future Work

We presented an end-to-end, edge-based anomaly detection system for motors using ESP32 and an autoencoder trained on normal behavior. On a 2,280-sample dataset with 9 channels, a simple thresholding of reconstruction error yielded actionable flags with minimal compute. This demonstrates the feasibility of deploying anomaly detection at the edge using ESP32, paving the way for low-cost predictive maintenance in SMEs. Future directions include TinyML-specific optimizations, adaptive thresholds, spectral features for vibration, and long-horizon evaluation on diverse motors.

Artifacts and Reproducibility

All figures referenced in this paper are included alongside the LaTeX source. Machine-readable outputs for reproducibility are provided: `dataset_summary.csv` (feature statistics) and `autoencoder_results.csv` (per-sample MSE and anomaly flags).

Acknowledgment

The authors used AI-assisted writing tools for language editing under author supervision. All data analysis, experiments, and results are original contributions of the authors.

References

[1] H. Xu, Z. Liu, and J. Wang, "A comprehensive study of auto-encoders for anomaly detection," *Information Fusion*, vol. 104, p. 102038, 2024.
[2] S. F. Chevtchenko, R. A. F. Romero, and S.-J. Hsieh, "Predictive maintenance model based on anomaly detection in induction motors: A machine learning approach using real-time iot data," *arXiv preprint arXiv:2310.14949*, October 2023. [Online]. Available: https://arxiv.org/abs/2310.14949
[3] F. F. Junior, R. A. F. Romero, and S.-J. Hsieh, "Machine learning for the detection and diagnosis of anomalies in applications driven by electric motors," *Sensors*, vol. 23, no. 24, p. 9725, December 2023.
[4] M. Tang, L. Liang, H. Zheng, J. Chen, and D. Chen, "Anomaly detection of permanent magnet synchronous motor based on improved dwt-cnn multi-current fusion," *Sensors*, vol. 24, no. 8, p. 2553, April 2024.