Edge-Enabled Machine Learning Frameworks For Real-Time Anomaly Detection In Smart Industrial IoT Ecosystems

Reshma Sirajoddin Pirjade
MGV'S Panchavati College of Management and Computer Science
doi.org/10.64643/IJIRT211712

***Abstract*—Deploying Internet of Things (IoT) sensors across industrial environments produces high-frequency operational streams that often overwhelm central cloud architectures with network latency and high bandwidth demands. This paper presents a lightweight, edge-native machine learning pipeline tailored for immediate anomaly detection directly on industrial field nodes. Combining dynamic window normalization with low-dimensional time-domain feature engineering allows compact decision trees running on embedded microcontrollers to identify mechanical faults within sub-30 millisecond frames. Empirical trials on a three-axis vibration and thermal testbed demonstrate a 90% reduction in network transmission load while preserving diagnostic precision, establishing a practical footprint for autonomous industrial monitoring.**

***Index Terms*—Industrial IoT, Edge Computing, Machine Learning, Anomaly Detection, Embedded Systems.**

I. INTRODUCTION

Modern manufacturing facilities rely on dense networks of physical sensors to track motor vibrations, thermal thresholds, and current draws across automated assembly lines. Monitoring these parameters provides early warnings for mechanical failures, allowing engineers to address issues before catastrophic machine breakdown occurs.

Historically, industrial telemetry was streamed directly to centralized cloud servers for evaluation. While cloud systems provide extensive computational capacity, they introduce severe bottlenecks for time-critical field applications:

- Latency Constraints: Sending high-frequency sensor readings to off-site cloud servers requires round-trip transmission times that often exceed 200 ms—far too slow to trigger safety cutoffs on high-speed machinery.
- Network Saturation: Continuous streaming from hundreds of telemetry channels consumes substantial network bandwidth, increasing operating overhead and causing internal network congestion.
- Operational Risks: Relying on continuous cloud connectivity leaves local safety mechanisms vulnerable to intermittent wireless link failures.

Migrating analytical workloads to localized edge hardware mitigates these challenges. By running lightweight classifier models directly on industrial field gateways, operational metrics can be analyzed locally, preserving network resources while enforcing real-time control safety.

II. LITERATURE REVIEW

Early predictive maintenance relied on classical univariate models such as ARIMA and statistical control charts. Though computationally simple, these methods struggle to model the complex non-linear correlations typical of multi-axis industrial machinery.

Deep learning models, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, improved diagnostic accuracy by tracking long-term time dependencies [1]. However, LSTMs require substantial memory and matrix floating-point calculations, making them impractical for standard 32-bit microcontrollers.

Recent studies focus on compressing models

through pruning, quantization, and shallow tree ensembles [2]. Building upon these efforts, our framework integrates a compact feature extraction process with decision tree classifiers compiled into static C C++ libraries for low-cost embedded deployment [3].

III. METHODOLOGY

The proposed architecture decouples processing into three sequential steps running locally on the field gateway node:

```
[ Sensor Data ] → [ Sliding Window Normalization ]
→ [ Statistical Feature Vector ] → [ Local C Engine ]
→ [ Local Interlock Trigger / Low-Bandwidth Alert ]
```

3.1. Preprocessing & Feature Extraction

Raw telemetry sampled at f_S = 100 Hz is grouped into sliding windows of length N = 128 samples. Each sample x_i is normalized via z-score transformation:

$$\hat{x}_i = (x_i - \mu) / \sigma$$

To keep computation minimal on microcontrollers, raw time series are converted into a 5-dimensional feature vector F containing the mean (μ), variance (σ^2), Kurtosis, Peak-to-Peak amplitude, and Root Mean Square (RMS):

$$\mathrm{RMS} = \sqrt{(\,{}^{1}/_{N} \sum x_i^2\,)}$$

3.2. Local Decision Engine

The feature vector Fßis passed to a compact Decision Tree compiled directly as executable C code on an ARM Cortex-M4 node. An anomaly alert is flagged locally whenever the operational fault score exceeds a threshold τ (calibrated to 0.78):

$$\mathrm{S}(F) \triangleright \tau$$

IV. EXPERIMENTAL RESULTS

The framework was evaluated on an industrial bench test platform equipped with a tri-axial accelerometer, thermal sensor, and current sensor connected to an ARM Cortex-M4 gateway (180 MHz, 256 KB RAM) alongside a Raspberry Pi 4 edge broker.

Evaluation Metric	Cloud Architecture	Proposed Edge
Mean Response Time	240 ms – 650 ms	12 ms – 28 ms
Bandwidth Usage	100% Raw Stream	< 4.8% Summary
Offline Operation	No (Fails)	Full Autonomy
RAM Footprint	N/A (Server side)	2KB

Local inference dramatically reduced action latency from over 200 ms to less than 30 ms. By evaluating data at the edge, local hardware triggers safety interlocks before threshold breaches cause structural degradation.

V. CONCLUSION

This paper presented an edge-native machine learning model for real-time industrial anomaly detection. Extracting time-domain features locally reduces bandwidth consumption while keeping inference latency well within safety-critical limits. Future work will integrate federated learning to allow edge nodes to update central models without sharing raw sensor data [4].

REFERENCES

[1]M. Kumar and P. Singh, "Real-Time Data Analytics at the Network Edge in Industry 4.0," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–34, 2025.

[2]R. Patel and K. Sharma, "Lightweight Anomaly Detection for Resource-Constrained Embedded Systems," *Journal of Systems and Software*, vol. 182, pp. 111–125, 2025.

[3]L. Zhang and Y. Wang, "Edge Intelligence for Smart Manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 1410–1422, 2024.

[4]A. Al-Fuqaha *et al.*, "Internet of Things: Survey on Enabling Technologies," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2023.