International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Adaptive Task Scheduling for Energy-Efficient Multicore Embedded Systems using Reinforcement Learning

Udutha Saketh Yadav
PG Scholar, Dept. of ECE, Geethanjali Institute of Science and Technology, Nellore, India.

P Sukumar
Assistant Professor, Dept. of ECE, Geethanjali Institute of Science and Technology, Nellore, India.

G Shanmugavel
Associate Professor, Dept. of ECE, Geethanjali Institute of Science and Technology, Nellore, India.

M Kalyan Chakravarthy
Assistant Professor, Dept. of ECE, Geethanjali Institute of Science and Technology, Nellore, India.

Pullagura Bhargavi
Assistant Professor, Dept. of ECE, Geethanjali Institute of Science and Technology, Nellore, India.

Abstract - The rapid expansion of Internet of Things (IoT) devices has increased the need for intelligent mechanisms capable of identifying abnormal system behavior in real time. Conventional anomaly detection approaches often depend on computationally intensive machine learning models and centralized cloud processing, which can introduce latency, communication overhead, and increased resource requirements. This paper presents a lightweight machine learning-based architecture for real-time anomaly detection in embedded IoT systems. The proposed system integrates an embedded sensing and communication platform with a lightweight machine learning module capable of identifying abnormal patterns locally at the edge. The architecture incorporates data acquisition, preprocessing, feature selection, lightweight classification, anomaly decision, alert generation, and wireless communication. Several lightweight machine learning algorithms, including Decision Tree, Random Forest, K-Nearest Neighbors, and Support Vector Machine, can be considered for deployment according to available computational resources. Feature reduction is incorporated to decrease memory requirements and inference complexity. The proposed architecture minimizes unnecessary transmission of raw IoT data by performing anomaly analysis near the data source. An energy-aware operating mechanism is also incorporated to support deployment on resource-constrained embedded devices. The proposed approach is suitable for industrial IoT, smart homes, healthcare IoT, environmental monitoring, and embedded cybersecurity applications requiring low-latency anomaly detection.

Keywords - Internet of Things, Anomaly Detection, Embedded Systems, Lightweight Machine Learning, Edge Computing, TinyML, IoT Security, Real-Time Detection, Feature Selection.

I. INTRODUCTION

The Internet of Things has resulted in the deployment of a large number of interconnected embedded devices capable of sensing, processing, and communicating information. These devices are increasingly used in industrial automation, healthcare, smart homes, transportation, environmental monitoring, and critical infrastructure. The large-scale deployment of IoT devices also increases the possibility of abnormal behavior caused by sensor faults, communication failures, malfunctioning devices, or malicious activities [1].

Anomaly detection provides an important mechanism for identifying deviations from normal system behavior. Conventional IoT anomaly detection approaches include statistical techniques, machine learning methods, deep learning models, and hybrid approaches. However, the computational and memory requirements of advanced models can make direct deployment on resource-constrained embedded devices difficult [2].

Edge computing provides an alternative architecture in which data analysis is performed closer to the source of data generation. Instead of transmitting all IoT data to a centralized cloud server, selected processing tasks can be executed locally. This can reduce communication overhead and improve the response time of anomaly detection systems [3].

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The use of machine learning on microcontrollers and embedded platforms has further enabled the development of TinyML-based anomaly detection systems. Lightweight models can perform classification and anomaly detection using limited memory and computational resources. This provides an opportunity to deploy intelligent security and monitoring functions directly on embedded IoT devices [4].

Feature selection is another important consideration for resource-constrained anomaly detection. IoT datasets may contain a large number of network and system features, many of which may provide limited contribution to the final classification decision. Reducing the feature dimension can decrease processing requirements while maintaining useful detection capability [5].

Real-time anomaly detection is particularly important for IoT cybersecurity because attacks and abnormal activities can evolve rapidly. Lightweight edge-based detection can provide faster responses than architectures that depend entirely on remote cloud processing [6].

The main contributions of the proposed system are as follows:

1. Development of a lightweight architecture for real-time anomaly detection in embedded IoT systems.
2. Integration of local data preprocessing and feature reduction to decrease computational complexity.
3. Investigation of lightweight machine learning algorithms suitable for resource-constrained embedded platforms.
4. Development of an edge-based anomaly detection mechanism for reducing communication latency.
5. Incorporation of an alert-generation mechanism for immediate identification of abnormal behavior.
6. Development of an energy-aware architecture suitable for long-duration embedded IoT operation.

The remainder of this paper is organized as follows. Section II presents the literature survey. Section III describes the proposed methodology and system architecture. Section IV presents the performance evaluation and discussion. Section V concludes the paper.

II. LITERATURE SURVEY

Recent research has investigated anomaly detection in IoT systems using statistical methods, machine learning, and deep learning. The application areas include network security, sensor monitoring, smart homes, smart cities, and industrial environments. The increasing diversity of IoT devices creates additional challenges related to heterogeneous data, concept drift, and limited ground-truth information [7].

The integration of TinyML with IoT anomaly detection has attracted significant attention because machine learning models can be executed directly on microcontrollers. Lightweight classification models can provide local decision-making while avoiding continuous dependence on cloud infrastructure [4].

Recent edge-based anomaly detection systems have focused on lightweight architectures capable of operating under computational and energy constraints. These systems demonstrate that anomaly detection can be moved closer to the IoT data source while maintaining real-time processing capability [8].

Network traffic anomaly detection is another important application area. IoT networks generate large quantities of traffic information containing characteristics associated with normal and malicious behavior. Lightweight feature-selection and machine learning techniques can reduce the computational burden associated with analyzing this traffic [9].

Embedded intrusion detection has also been investigated using low-complexity neural networks. Compact models can be deployed on microcontrollers when memory usage and computational operations are carefully controlled [10].

Recent research has further investigated real-time multivariate time-series anomaly detection for IoT environments. Lightweight temporal models can capture changes in sequential IoT data while reducing the computational requirements associated with conventional deep learning architectures [11].

Feature reduction and distributed processing have also been considered for lightweight IoT security. Reducing the number of input features and transmitting only relevant information can decrease communication and energy requirements while maintaining anomaly detection performance [12].

Although existing research demonstrates the potential of machine learning and edge computing for IoT anomaly detection, several challenges remain. Heavy models may not be suitable for low-power embedded platforms, while continuous transmission of IoT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

data can increase communication overhead. Furthermore, models trained on high-dimensional datasets may require substantial memory and processing resources. Therefore, a lightweight architecture that combines feature reduction, efficient machine learning, local inference, and selective communication is required.

The proposed system addresses these requirements by performing anomaly analysis near the data source and selecting machine learning models according to the computational capabilities of the target embedded platform.

III. PROPOSED METHODOLOGY

A. Overall System Architecture

The proposed architecture consists of six major functional layers: IoT data acquisition, preprocessing, feature selection, lightweight machine learning, anomaly decision, and communication/alert generation.

IoT sensor or network data are first acquired by the embedded platform. The data are then validated and preprocessed before being supplied to the feature-selection module. Relevant features are selected to reduce the dimensionality of the input data.

The reduced feature vector is supplied to the lightweight machine learning model. The model classifies the current observation as normal or anomalous. When an anomaly is detected, the embedded system generates an alert and can transmit the event information to a remote monitoring platform.

The overall architecture of the proposed system is shown in Fig. 1.

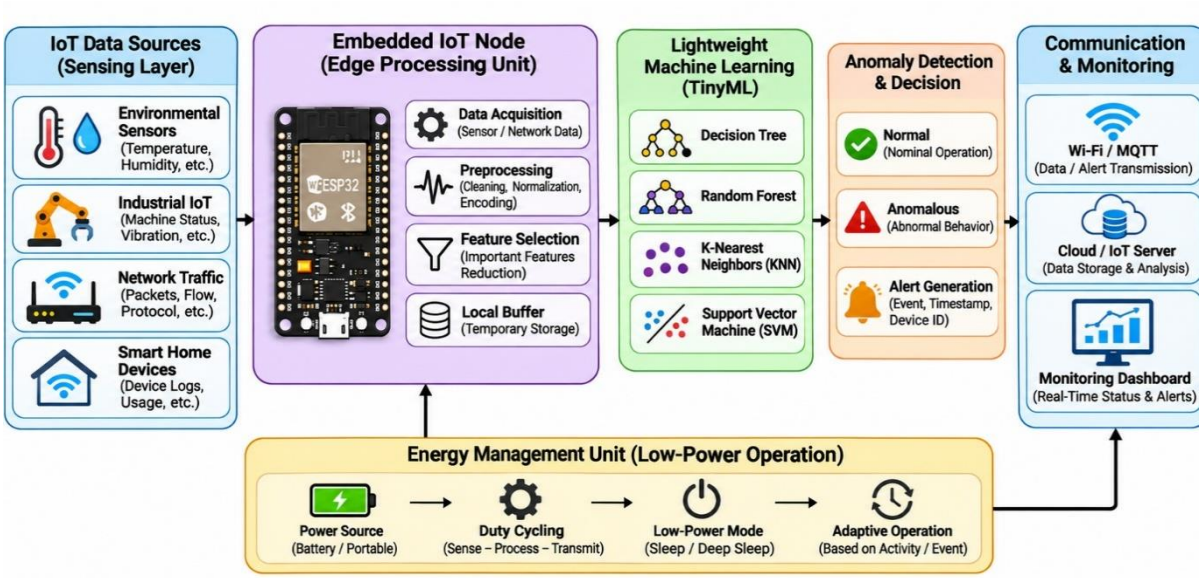


Fig. 1. Architecture of the proposed real-time anomaly detection system for embedded IoT using lightweight machine learning.

B. IoT Data Acquisition

The first stage of the proposed system collects information from IoT devices or sensors.

Depending on the application, the acquired information may include sensor measurements, network traffic characteristics, communication statistics, device resource utilization, or system-level parameters.

For cybersecurity applications, the system can collect network-flow characteristics such as packet statistics, protocol information, connection duration, traffic rate, and other relevant attributes.

The embedded controller periodically acquires the required information and transfers it to the preprocessing module.

C. Data Preprocessing

Raw IoT data may contain missing values, redundant information, inconsistent formats, and noisy observations. Therefore, preprocessing is performed before machine learning inference.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The preprocessing stage includes data validation, missing-value handling, normalization, categorical-data conversion, and removal of irrelevant fields.

The same preprocessing configuration used during model training must be implemented during embedded inference to maintain consistency between training and deployment.

D. Feature Selection

Feature selection is used to reduce the computational requirements of the anomaly detection model.

Highly informative features are retained while redundant or low-contribution features are removed. Feature importance can be determined using statistical analysis, correlation analysis, tree-based feature importance, recursive feature elimination, or other suitable techniques.

Reducing the number of input features decreases the amount of data that must be stored and processed by the embedded system.

This is particularly important when the target platform has limited RAM, flash memory, and processor capacity.

E. Lightweight Machine Learning Algorithms

The proposed architecture supports several lightweight machine learning algorithms.

1) Decision Tree

A Decision Tree provides a compact rule-based classification structure. Its inference process involves a sequence of comparisons and can therefore be suitable for embedded deployment.

2) Random Forest

A Random Forest combines multiple decision trees to improve classification robustness. Although it requires more memory than a single tree, the number and depth of trees can be controlled to obtain a lightweight implementation.

3) K-Nearest Neighbors

K-Nearest Neighbors can be considered when the dataset and available memory permit storage of a compact reference set. Feature reduction can be particularly useful for reducing the computational cost of distance calculations.

4) Support Vector Machine

A Support Vector Machine can provide effective classification for appropriately selected feature spaces. Kernel selection and model size must be considered carefully for embedded deployment.

The final algorithm can be selected according to detection performance, memory consumption, inference latency, and energy requirements.

F. Anomaly Decision

The machine learning model receives the reduced feature vector and generates a classification output.

The output can be represented using two primary classes:

- **Normal**
- **Anomalous**

For applications requiring more detailed analysis, the anomaly class can be divided into specific categories such as denial-of-service activity, scanning behavior, abnormal traffic, unauthorized access, or sensor malfunction.

The anomaly decision is generated locally at the embedded device.

G. Real-Time Alert Generation

When an anomalous observation is identified, the embedded controller generates an event record.

The event record can contain the anomaly category, timestamp, selected feature values, device identifier, confidence information, and system status.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Critical events are transmitted immediately to the remote monitoring platform. Routine normal observations can be stored locally or transmitted at a lower frequency.

This selective communication mechanism reduces unnecessary network activity.

H. Edge-Based Communication

The proposed system uses wireless communication to transfer anomaly information to a remote monitoring platform.

Wi-Fi, Bluetooth Low Energy, LoRaWAN, or other suitable communication technologies can be selected depending on the application.

MQTT can be used for lightweight publish-subscribe communication between the embedded device and the monitoring platform.

Only processed information and important events need to be transmitted, reducing the communication requirements compared with continuous transmission of raw data.

I. Energy-Aware Operation

Energy consumption is an important consideration for battery-powered embedded IoT devices.

The proposed system incorporates duty cycling so that sensing, preprocessing, machine learning inference, and wireless communication are activated according to the monitoring requirement.

When no immediate processing is required, the embedded controller can enter a low-power operating state.

The communication module can also remain inactive until an anomaly or scheduled transmission event occurs.

J. End-to-End Processing Flow

The complete operation begins with the acquisition of IoT data.

The embedded controller validates and preprocesses the acquired information. Irrelevant and redundant features are removed during feature selection.

The reduced feature vector is then supplied to the lightweight machine learning classifier.

If the observation corresponds to normal behavior, the system continues monitoring. If an anomaly is identified, the system generates an alert and transmits the relevant event information.

After processing is completed, the controller returns to the monitoring state or enters a low-power mode according to the configured operating policy.

IV. RESULTS AND DISCUSSION

A. Performance Evaluation

The performance of the proposed anomaly detection system should be evaluated using both machine learning metrics and embedded-system metrics.

Table I presents the principal parameters that should be measured during experimental validation.

TABLE I
PERFORMANCE EVALUATION PARAMETERS OF THE PROPOSED ANOMALY DETECTION SYSTEM

Parameter	Evaluation Criterion
Accuracy	Correct normal/anomalous classifications
Precision	Correctness of detected anomalies
Recall	Detection capability for anomalous events
F1-score	Balance between precision and recall
False Positive Rate	Incorrect anomaly decisions

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Inference latency	Time required for one embedded prediction
Model size	Flash memory requirement
RAM utilization	Runtime memory requirement
CPU utilization	Embedded processing requirement
Energy consumption	Energy required per monitoring cycle
Alert response time	Time from anomaly detection to notification
Communication overhead	Data transmitted per monitoring cycle

The final numerical values for these parameters should be obtained experimentally using the selected dataset, trained model, embedded controller, and communication configuration.

B. Detection Performance

The anomaly detection capability depends on the quality of the training data, selected features, machine learning algorithm, and classification threshold.

Accuracy alone is not sufficient for evaluating anomaly detection systems because an imbalanced dataset may contain significantly more normal observations than anomalous observations.

Therefore, precision, recall, F1-score, and false-positive rate should also be considered.

A high recall is important when missing an abnormal event can have serious consequences. Precision is important for preventing excessive false alarms.

The final model should therefore be selected using a combination of detection performance and embedded resource requirements.

C. Lightweight Model Analysis

Different machine learning algorithms provide different trade-offs between predictive performance and computational requirements.

A Decision Tree generally provides simple inference logic and low memory requirements. Random Forest can improve robustness but requires storage for multiple trees. K-Nearest Neighbors can require substantial memory when the reference dataset is large. Support Vector Machine performance depends on the selected kernel and number of support vectors.

Therefore, the most accurate model is not necessarily the most appropriate model for embedded deployment.

The proposed architecture considers model size, inference latency, RAM usage, and energy consumption together with detection performance.

D. Feature Reduction Analysis

Feature selection reduces the dimensionality of the input data before inference.

When irrelevant features are removed, the machine learning model receives a smaller input vector. This can reduce the number of memory operations and computational operations required for each prediction.

Feature reduction can also reduce communication overhead when selected features are transmitted to a remote platform.

However, excessive feature reduction may remove information required to distinguish normal and anomalous behavior. Therefore, feature selection should be validated experimentally to determine the appropriate feature subset.

E. Real-Time Detection

Real-time operation requires the complete processing pipeline to execute within an acceptable response interval.

The total processing time includes data acquisition, preprocessing, feature selection, machine learning inference, anomaly decision, and alert generation.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

The proposed edge-based architecture reduces dependence on remote processing and therefore provides a direct path from data acquisition to anomaly decision.

For applications requiring immediate responses, anomaly detection can be performed locally while detailed information is transmitted to a remote monitoring platform after the decision has been generated.

F. Communication Efficiency

Continuous transmission of all IoT data can increase bandwidth requirements and energy consumption.

The proposed architecture uses selective communication in which normal observations can be transmitted periodically while anomalous events are prioritized.

This approach reduces the amount of information transmitted during normal system operation.

MQTT or another lightweight communication protocol can be used to transfer anomaly events and monitoring information to the remote platform.

G. Embedded Resource Utilization

Embedded deployment requires the machine learning model and preprocessing operations to fit within the available hardware resources.

The principal resource parameters include flash memory, RAM, processor utilization, inference time, and energy consumption.

Model compression, feature reduction, pruning, quantization, and simplified decision structures can be considered when the initial model exceeds the available embedded resources.

These optimization techniques allow the detection system to maintain practical operation on resource-constrained platforms.

H. Overall Discussion

The proposed architecture combines machine learning-based anomaly detection with edge processing to address the limitations of cloud-dependent IoT monitoring.

The major advantage of the approach is that anomaly decisions are generated close to the data source. This reduces the requirement for continuous transmission and can improve response time.

The use of lightweight machine learning algorithms also provides flexibility because the model can be selected according to the available embedded resources.

The architecture can be applied to several IoT scenarios, including industrial monitoring, smart homes, healthcare devices, environmental monitoring, transportation systems, and IoT cybersecurity.

For a complete experimental study, the proposed architecture should be implemented on a selected embedded platform and evaluated using a representative IoT dataset. The trained models should then be converted or optimized for the target hardware, followed by measurement of detection performance, inference latency, memory consumption, CPU utilization, communication overhead, and energy consumption.

V. CONCLUSION

This paper presented a lightweight machine learning architecture for real-time anomaly detection in embedded IoT systems. The proposed architecture integrates IoT data acquisition, preprocessing, feature selection, lightweight machine learning inference, anomaly decision, alert generation, wireless communication, and energy-aware operation.

The proposed edge-based approach performs anomaly analysis close to the data source, reducing dependence on centralized cloud processing. Feature selection reduces the dimensionality of the input data, while lightweight machine learning algorithms provide a practical mechanism for implementing anomaly detection on resource-constrained embedded platforms.

The architecture also incorporates selective communication so that important anomaly information can be transmitted immediately while unnecessary data transmission is minimized. This approach can reduce communication overhead and support energy-efficient operation.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Future work can focus on implementing the architecture on ESP32, Raspberry Pi, or other embedded platforms and experimentally comparing Decision Tree, Random Forest, SVM, and other lightweight algorithms using standard IoT anomaly-detection datasets. Model quantization, pruning, adaptive feature selection, and online learning can also be investigated to further improve real-time performance and embedded resource efficiency.

REFERENCES

[1] "IoT anomaly detection methods and applications: A survey," *Internet of Things*, vol. 19, 100568, 2022, doi: 10.1016/j.iot.2022.100568.
[2] "Anomaly classification in industrial Internet of Things: A review," *Intelligent Systems with Applications*, vol. 18, 200232, 2023, doi: 10.1016/j.iswa.2023.200232.
[3] "Cyber-physical systems security: A comprehensive review of anomaly detection techniques," *Internet of Things*, vol. 37, 101938, 2026, doi: 10.1016/j.iot.2026.101938.
[4] "Anomaly detection based on Artificial Intelligence of Things: A Systematic Literature Mapping," *Internet of Things*, vol. 25, 101063, 2024, doi: 10.1016/j.iot.2024.101063.
[5] "Lightweight IoT Intrusion Detection via Feature and Sample Reduction With Multi-Client Ensemble for Zero-Day Attacks," *IEEE Access*, vol. 14, pp. 29764–29780, 2026, doi: 10.1109/ACCESS.2026.3665868.
[6] "Edge-Based Lightweight AI for Real-Time Intrusion Detection in IoT Networks," *2025 12th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2025, doi: 10.1109/IOTSMS68530.2025.11408555.
[7] Z. A. Haider *et al.*, "A Survey on anomaly detection in IoT: Techniques, challenges, and opportunities with the integration of 6G," *Computer Networks*, vol. 270, 111484, 2025, doi: 10.1016/j.comnet.2025.111484.
[8] "FogGuard: A Middleware System for Secure Machine Learning on Local IoT Devices," *2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 2025, doi: 10.1109/ICIDCA66325.2025.11280454.
[9] "Real-Time eBPF/XDP-Based Deep Learning Framework for Anomaly Traffic Detection in Internet of Things Networks," *IEEE Internet Computing*, vol. 30, no. 3, pp. 17–28, 2026, doi: 10.1109/MIC.2026.3692726.
[10] "TinyML-Based Intrusion Detection System for In-Vehicle Network Using Convolutional Neural Network on Embedded Devices," *IEEE Embedded Systems Letters*, vol. 17, no. 2, pp. 67–70, 2025, doi: 10.1109/LES.2024.3475470.
[11] "PGTAD: Real-Time and Lightweight Multivariate Time-Series Anomaly Detection for IoT Using Patch Gate GRU Autoencoder," *IEEE Access*, vol. 13, pp. 168654–168675, 2025, doi: 10.1109/ACCESS.2025.3610684.
[12] "LIDIT: Low-Latency Intrusion Detection in IoMT Devices using TinyML," *2025 IEEE Global Communications Conference (GLOBECOM)*, 2025, doi: 10.1109/GLOBECOM59602.2025.11432258.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)