Energy-Efficient XGBoost-Based Approach for IoT Threat Detection

1st Ahmad Ghaffari
Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
mirzahmad7204@gmail.com

2nd Fahad Khalid
Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
itzfahad72@gmail.com

3rd Aoun Muhummad
Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhummad@iub.edu.pk

4th Umar Fayyaz
Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
umar.fayyaz@iub.edu.pk

5th Sehrish Raza
Faculty of Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sehrishraza6@gmail.com

***Abstract*—The IoT is widely used globally to automate operations, enhance system functionality, and enable remote monitoring and diagnostics. IoT devices incorporate sensors that continuously collect real-time information and relay it to a central system or controller. This central system typically employs machine learning models to process incoming data and regulate system behavior. Despite these benefits, IoT systems present several challenges. Security is a major concern, as most IoT devices are connected to the Internet or local networks and may be vulnerable to attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), and Man-in-the-Middle (MitM). Energy efficiency is another important concern, since many existing intrusion detection models are computationally intensive and consume substantial energy, limiting their use in resource-constrained IoT devices. In view of these challenges, this work presents a lightweight machine learning model for IoT threat detection. The proposed solution classifies network traffic using selected features to detect malicious activity while maintaining low energy consumption. The TON-IoT dataset is used to train and evaluate the model, achieving a balance between detection accuracy and computational performance.**

***Index Terms*—IoT Threat Detection, Energy-Efficient Machine Learning, XGBoost, Edge Computing**

I. Introduction

The Internet of Things (IoT) is a rapidly developing technology that enables pervasive connectivity in smart homes, healthcare systems, industrial automation, and critical infrastructure [1]. IoT devices continuously sense and respond to their surrounding environments, generating large volumes of data that can be used to automate processes, support analytics, and improve operational efficiency. Although these capabilities offer significant advantages, they also introduce important security challenges. Many IoT devices possess limited built-in security features, operate with low-power hardware, and have constrained memory and processing capacity, making them attractive targets for network-based attacks [2], [15].

Traditional intrusion detection systems (IDSs) and enterprise-oriented machine learning models are often unsuitable for IoT environments. They typically require substantial memory, high computational resources, and exhibit slow response times, restricting real-time threat detection in resource-limited networks. Furthermore, many previous studies rely on synthetic datasets or oversampling to balance traffic distributions, which can overestimate performance and fail to reflect the heterogeneity of real IoT traffic [3].

This study focuses on the design of an energy-efficient XGBoost-based model for IoT threat detection. The design emphasizes realistic traffic modeling, low energy consumption, and interpretability. The major contributions of this work are:

- development of an energy-aware XGBoost framework incorporating feature selection, tree pruning, and histogram-based splitting to reduce computational and memory overhead;
- evaluation using real traffic from heterogeneous IoT devices without artificially modifying traffic distribution;
- resource-aware analysis including CPU and memory monitoring to support deployment on IoT edge gateways;
- consideration of trade-offs between energy efficiency and detection performance to support transparent and reproducible evaluation.

Recent studies indicate that consumer and industrial IoT devices are increasingly exposed to network-based attacks such as DDoS, botnet propagation, and protocol misuse, highlighting the need for effective yet lightweight intrusion detection mechanisms [10], [11]. Recent advances in edge computing further emphasize deploying security systems closer to IoT devices to reduce latency and bandwidth usage. However, edge-based intrusion detection operates under strict constraints on energy, memory, and computational capacity. This has created demand for solutions that balance detection accuracy with operational efficiency. Lightweight machine learning techniques deployed on IoT gateways therefore represent a

DOI: https://doi.org/10.67535/tsp.000002.030
Published under the ISAISS 2026 Proceedings

promising approach for scalable and sustainable IoT security [10], [12], [15]. Based on the above challenges, the main contributions of this paper are summarized as follows.

A. Novelty and Contributions

The novelty of this work can be summarized as follows:

- A lightweight XGBoost-based intrusion detection model is designed specifically for resource-constrained IoT environments.
- The model is evaluated on heterogeneous real IoT traffic without synthetic class balancing, ensuring realistic assessment conditions.
- Energy, CPU, and memory consumption are explicitly monitored to study deployment feasibility on IoT edge gateways.
- The study highlights the trade-off between detection accuracy and computational efficiency rather than optimizing accuracy alone.

These contributions complement existing research that mainly focuses on maximum accuracy or relies on computationally intensive deep learning approaches.

II. Related Work

Intrusion detection in IoT networks has been widely studied. Classical statistical and rule-based methods rely on anomaly thresholds or manually configured signatures. Although effective in specific scenarios, such approaches struggle to cope with the diversity of IoT devices and communication protocols. Machine learning methods such as Random Forests, Support Vector Machines (SVM), and k-Nearest Neighbors (k-NN) can be trained on historical traffic data to identify attack patterns with improved accuracy [5], [14].

Deep learning approaches, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, capture complex spatial and temporal traffic relationships [7], [8]. Although often achieving high detection performance, these models typically require substantial computational resources and memory, which limits their practicality on energy-constrained IoT devices [7], [8], [12].

Gradient boosting models such as XGBoost provide a practical balance between efficiency and accuracy [6]. XGBoost incorporates pruning, feature subsampling, and histogram-based splits, making it suitable for resource-constrained environments. This work evaluates an energy-efficient XGBoost model on heterogeneous IoT traffic and conducts detailed resource and energy consumption analysis, in contrast to several earlier studies that relied on synthetic data or single-device scenarios [4]. Recent research has emphasized lightweight, ensemble-based intrusion detection systems designed for IoT and edge environments, demonstrating that carefully constructed models can achieve competitive performance while maintaining low energy and computational demands [14], [17], [18].

A. Comparison with Existing Work

Most previous IoT intrusion detection systems reported in the literature mainly focus on improving detection accuracy, often using balanced or synthetic datasets. These settings do not always represent realistic IoT traffic, which is typically heterogeneous and highly imbalanced. In addition, many works rely on computationally expensive deep learning models that are difficult to deploy on resource-constrained gateways. In contrast, the present work emphasizes energy-efficient intrusion detection using a lightweight XGBoost model. The evaluation is carried out on real heterogeneous IoT traffic without data oversampling, and CPU and memory usage are explicitly measured. This makes the proposed system more aligned with practical deployment requirements in edge environments compared with existing approaches.

III. Methodology and Workflow Diagrams

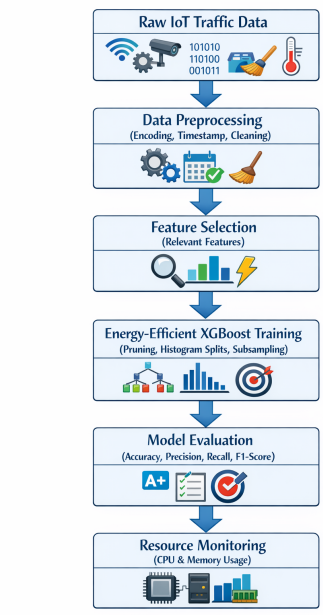

Fig. 1. Workflow for Data Preprocessing and Energy-Aware XGBoost Training

The workflow diagram in Fig. 1 summarizes the proposed energy-efficient IoT intrusion detection pipeline. Raw IoT traffic is collected from heterogeneous devices and preprocessed through encoding, timestamp conversion, and data cleaning. Relevant features are then selected to reduce computational cost. The XGBoost model is optimized using pruning, subsampling, and histogram-based splits, trained on the processed dataset, and evaluated while monitoring CPU and memory usage.

IV. Dataset Preparation and Preprocessing

The dataset is prepared, preprocessed, and split in Sections A, B, and C, respectively.

A. Dataset Construction

The TON_IoT dataset [4] contains real IoT traffic collected under normal and attack conditions from heterogeneous devices such as smart refrigerators, thermostats, GPS trackers, and Modbus industrial sensors. These devices exhibit distinct traffic characteristics in protocol usage, communication frequency, and payload size.

To reflect heterogeneous IoT environments, traffic from all devices is aggregated into a single dataset. The dataset contains over 1.9 million samples with 17 features selected based on relevance to device behavior and network activity.

B. Preprocessing

Preprocessing is performed with emphasis on efficiency and realism:

- categorical features are label-encoded to minimize memory overhead;
- timestamps are converted to Unix epoch format to preserve temporal order;
- incomplete or missing records are removed to prevent noise;
- feature selection retains the most informative attributes to reduce computation while preserving threat-relevant information.

Ensures realistic evaluation by retaining natural traffic distribution and device behavior.

C. Train-Test Split

A chronological split is applied, using the first 80% of data for training and the final 20% for testing. This avoids data leakage and reflects realistic deployment conditions.

V. Model Design and Optimization

A. Baseline Random Forest Model

A Random Forest classifier is used as a baseline. While it achieves competitive detection accuracy, it requires substantial memory and computation, limiting deployment on constrained IoT systems. Ensemble size and tree depth contribute to increased runtime and memory demand.

B. Energy-Efficient XGBoost Architecture

The proposed XGBoost model is designed to improve energy efficiency while maintaining satisfactory detection capability:

- 60 trees with a maximum depth of 4 are used to limit computational cost;
- histogram-based tree construction accelerates split computation;
- row and column subsampling reduces memory usage during training;
- parallelism is restricted to two CPU threads to limit power consumption while maintaining reasonable training speed.

C. Energy-Saving Strategies

- **Feature selection:** retaining only informative features reduces memory and computation;
- **Model pruning:** limiting depth and estimator count lowers energy usage with minimal accuracy loss;
- **Histogram-based splits:** continuous variables are discretized to accelerate training.

VI. Experimental Setup

All experiments in this were conducted on a commodity laptop representative of edge-level resources. The implementation was executed on a system with an Intel Core i7 processor, 16 GB RAM, and Windows 11. Python 3.12 with Scikit-learn, XGBoost, and psutil libraries was used for model implementation and monitoring. CPU usage, memory footprint, and latency were measured during both training and inference using psutil while the IDS processed test traffic, yielding realistic resource-consumption estimates.

VII. Experimental Results

A. Detection Performance Metrics

The Random Forest (RF) model achieved 89.10% accuracy, 70.45% precision, 84.20% recall, and a 76.50% F1-score. The energy-aware XGBoost model achieved 88.40% accuracy, 69.91% precision, 83.99% recall, and a 76.31% F1-score.

TABLE I
Performance comparison of RF and XGB (Energy-Aware)

Model	Acc.	Prec.	Rec.	F1
RF	89.10%	70.45%	84.20%	76.50%
XGB (EA)	88.40%	69.91%	83.99%	76.31%

B. Training Results Visualization

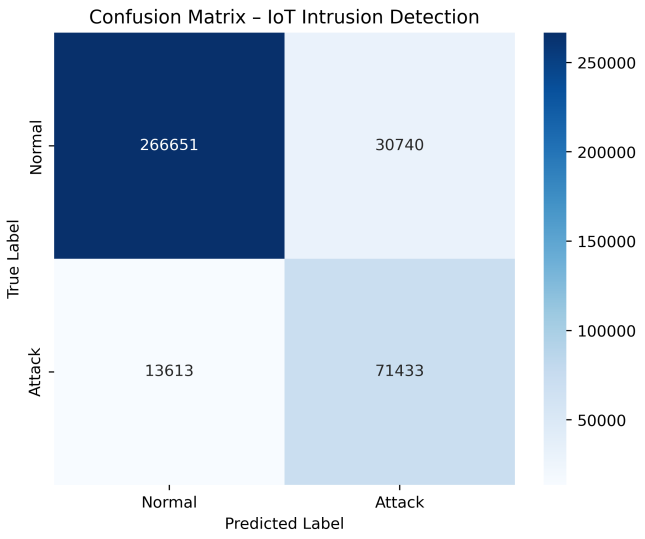

Fig. 2. Confusion Matrix after Training

The confusion matrix in Fig. 2 illustrates model performance on heterogeneous IoT traffic. A high proportion of normal traffic is correctly classified, and most attack traffic is detected, with relatively few false positives and false negatives. Misclassifications primarily arise from traffic heterogeneity and class imbalance. Overall, the matrix reflects a balanced trade-off between detection capability and efficiency.

C. Distribution of TP,FP,FN,TN during Training

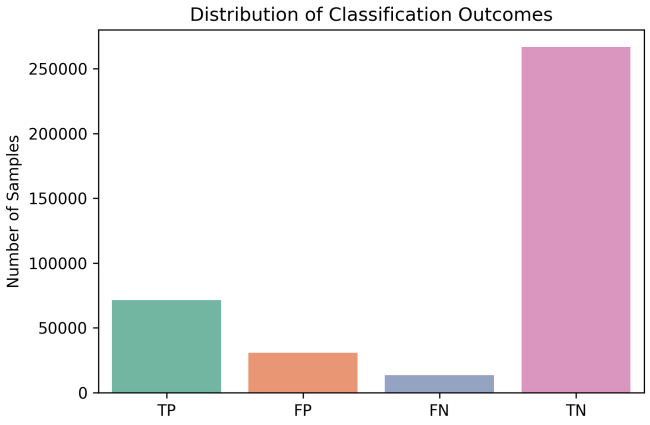

Fig. 3. Distribution of True Positives, False Positives, False Negatives, and True Negatives after Training

The distribution in Fig. 3 shows high true-positive and true-negative counts, indicating effective discrimination between benign and malicious traffic. Low false-positive and false-negative counts reflect balanced sensitivity and specificity.

D. Resource Utilization during Training

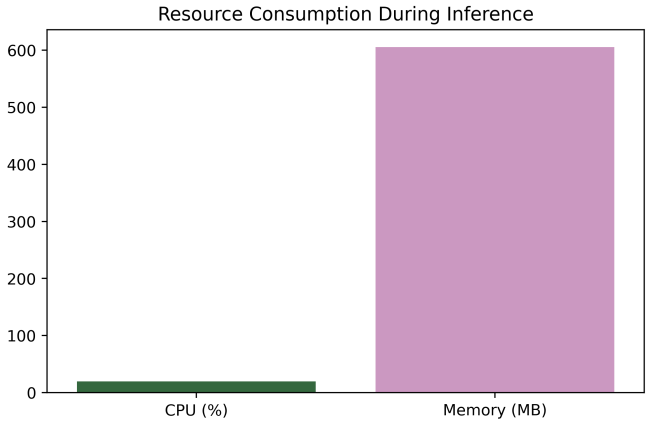

Fig. 4. CPU and Memory Consumption during Training

Fig. 4 shows CPU and memory consumption during training. CPU usage remains low, supporting deployment on resource-constrained edge gateways. Memory usage is moderate due to ensemble storage but remains acceptable for edge environments.

E. Detection Performance Metrics

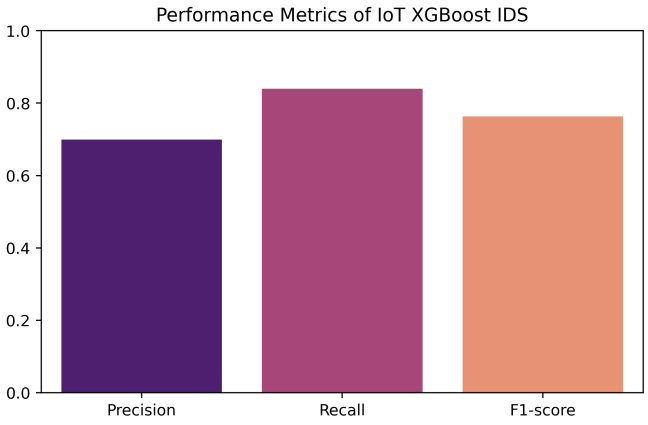

Fig. 5. Detection performance metrics of the proposed XGBoost-based IoT intrusion detection model

Fig. 5 presents the precision, recall, and F1-score achieved by the proposed model. The high recall value indicates strong capability in detecting malicious traffic, while the balanced precision and F1-score demonstrate that the model maintains reliable overall classification performance without significantly increasing false alarms.

TABLE II
COMPARISON WITH EXISTING IOT INTRUSION DETECTION STUDIES

Study	Model Type	Dataset Handling	Resource Awareness
[7]	Deep learning	Balanced / synthetic	Not considered
[14]	Ensemble ML	Partially balanced	Limited discussion
[17]	Edge IDS	Synthetic traffic	Partial evaluation
This work	**Lightweight XGBoost**	**Real heterogeneous traffic, no oversampling**	**CPU and memory evaluated**

F. Discussion and Justification of Results

The obtained results was interpreted in the context of realistic evaluation choices and deployment constraints. The TON-IoT dataset is highly imbalanced and reflects heterogeneous traffic from multiple device types, which makes classification more difficult than studies that rely on small or artificially balanced datasets. No oversampling, synthetic attack generation, or aggressive hyperparameter tuning was applied, which avoids optimistic bias and improves reproducibility. The proposed model is intentionally constrained in depth and size to reduce energy consumption and inference latency on edge gateways. A further strength is that performance remains stable across different device categories rather than being tuned for a single device type. The system therefore emphasizes generalization, resource efficiency, and deployability instead of only maximizing accuracy on simplified benchmarks, which is more consistent with real IoT operating environments.

VIII. Limitations and Future Work

This study has several limitations that suggest directions for future research. First, the current intrusion detection system performs binary classification, distinguishing only between normal and attack traffic. Extending the model to multiclass detection would enable identification of specific attack types. Second, evaluation was carried out using a single publicly available dataset; validating the model on additional real-world datasets would strengthen generalizability. Third, experiments were performed on an edge-class laptop rather than on ultra-low-power IoT hardware. Future work includes deployment on Raspberry Pi–class devices to obtain detailed energy measurements. Finally, incorporating online or incremental learning could allow the IDS to adapt to evolving attack patterns over time.

IX. Conclusion

The results demonstrate that the proposed XGBoost-based intrusion detection model delivers stable detection performance under heterogeneous traffic and constrained computing conditions. Low CPU usage and moderate memory consumption are achieved through feature selection, shallow tree structures, and histogram-based splits. The model successfully distinguishes benign from malicious traffic without artificial data augmentation, helping preserve realistic traffic characteristics.

The study highlights energy and resource constraints as primary design considerations for IoT security systems. Rather than maximizing accuracy at the expense of computational cost, the approach balances performance and efficiency, which is essential for edge-based deployment. The findings indicate that well-tuned ensemble learning can provide effective and practical IoT intrusion detection, offering a foundation for future work on multiclass detection, adaptive learning, and on-device intelligence.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. Available: https://doi.org/10.1016/j.comnet.2010.05.010

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015. Available: https://doi.org/10.1016/j.comnet.2014.11.008

[3] T. A. Nguyen, G. Manzo, and L. Caviglione, "A survey of machine learning techniques for cybersecurity in the Internet of Things," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 4192–4230, 2018. Available: https://doi.org/10.1109/COMST.2019.2896380

[4] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique for IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 481–492, 2021. Available: https://doi.org/10.1109/JIOT.2020.3008472

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. Available: https://doi.org/10.1023/A:1010933404324

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794. Available: https://doi.org/10.1145/2939672.2939785

[7] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in *Proc. IEEE Int. Conf. on Big Data*, 2016, pp. 3038–3045. Available: https://doi.org/10.1109/BigData.2016.7840872

[8] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018. Available: https://doi.org/10.1109/TETCI.2017.2772792

[9] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection systems: Feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, p. 36, 2024. Available: https://doi.org/10.1186/s40537-024-00802-4

[10] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Security and Privacy Workshops*, 2018, pp. 29–35. Available: https://doi.org/10.1109/SPW.2018.00013

[11] Y. Meidan *et al.*, "Profiling IoT devices based on network traffic analysis," in *Proc. NDSS Symposium*, 2018. Available: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-3_Meidan_paper.pdf

[12] F. Restuccia, S. D'Oro, and T. Melodia, "Deep learning at the edge for cyber-physical systems," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 110–117, 2018. Available: https://doi.org/10.1109/MCOM.2018.1701310

[13] S. Marchal *et al.*, "AuDI: Toward autonomous IoT device-type identification using periodic communication," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1402–1412, 2019. Available: https://doi.org/10.1109/JSAC.2019.2904343

[14] A. Alqhatani, M. S. Hossain, and G. Muhammad, "Lightweight intrusion detection for IoT using ensemble learning," *IEEE Access*, vol. 10, pp. 34210–34223, 2022. Available: https://doi.org/10.1109/ACCESS.2022.3161839

[15] M. A. Ferrag *et al.*, "Security for 5G and IoT networks: A survey," *IEEE Access*, vol. 8, pp. 163448–163472, 2020. Available: https://doi.org/10.1109/ACCESS.2020.3015715

[16] B. A. Alzahrani and M. Alenazi, "Energy-efficient intrusion detection system for IoT networks," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4657–4665, 2021. Available: https://doi.org/10.1109/JSEN.2020.3035070

[17] A. Verma and V. Ranga, "ELNIDS: Ensemble learning-based network intrusion detection system for edge computing," *Computer Communications*, vol. 182, pp. 203–215, 2022. Available: https://doi.org/10.1016/j.comcom.2021.11.010

[18] A. Saleh, H. Bannour, and A. Kachouri, "Resource-aware intrusion detection for IoT networks using lightweight machine learning," *Future Generation Computer Systems*, vol. 139, pp. 148–160, 2023. Available: https://doi.org/10.1016/j.future.2022.09.016