

Enhanced IoT Security through Balanced Machine Learning-Based Intrusion Detection Systems



Received: 26/12/2025

Accepted: 06/03/2026

Published: 23/03/2026

Anamul Haque Sadial^{1*}, Atowar ul Islam¹, Sankar Haloi², S Md S Askari³

¹ Department of Computer Science, University of Science and Technology, Meghalaya, 9th Mile, Ribhoi, Meghalaya, India

² Department of Mathematics, Cotton University, Panbazar, Guwahati, Assam, India

³ Department of Computer Science and Engineering, Rajiv Gandhi University, Arunachal Pradesh, India

Citation: Sadial AH, Islam A, Haloi S, Askari SMS (2026) Enhanced IoT Security through Balanced Machine Learning-Based Intrusion Detection Systems. Indian Journal of Science and Technology 19(9): 631-645. <http://doi.org/10.17485/IJST/v19i9.2033>

* **Corresponding author.**

sadialanam@gmail.com

Funding: None

Competing Interests: None

Copyright: © 2026 Sadial et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN

Print: 0974-6846

Electronic: 0974-5645

Abstract

Background/Objectives: In modern days, the Internet of Things (IoT) networks produce complex and severely imbalanced traffic because conventional intrusion detection systems are becoming less reliable. **Method:** This study has produced a CLIB-EBGM IDS, a calibrated ensemble intrusion detection framework that operates under extreme class imbalance while prioritising the reduction of false negatives and maintaining high predictive performance. The proposed technique integrates XGBoost, Random Forest, and Logistic Regression classifiers with leakage-aware preprocessing, SMOTE-based balancing, probability calibration, and an Energy-Balanced Cost Learning (EBCL) threshold optimisation strategy to speed up the attack detection. Here, the CALIB-EBGM IDS is run on the Bot-IoT benchmark dataset, and it is observed that it performs very efficiently. **Findings:** The result shows that the proposed CALIB-EBGM IDS achieves 99.96% accuracy and 99.95% precision. From the result comparison, it is observed that the proposed approach outperforms existing ensembles and traditional models, particularly when the dataset is highly imbalanced. **Novelty:** The combination of calibrated ensemble learning and cost-sensitive optimization enables a reliable IDS suitable for IoT gateways and edge devices.

Keywords: Intrusion Detection System; Network Security; Machine Learning; Probability Calibration; Imbalanced Data; Attack Detection

1 Introduction

The fast growth of computer science and communication technologies has made the Internet of Things (IoT) a key part of modern digital systems. By connecting various devices through wired and wireless networks, IoT allows for automatic data sensing, collection, and transmission. This supports smart decision-making in areas like healthcare, industrial automation, transportation, education, and smart city environments⁽¹⁾. This expansion is happening at an incredible pace, with global IoT deployments expected to exceed 100 billion devices by 2040, up from around 0.3 billion devices in 2003. While this growth greatly improves automation and efficiency,

it also increases security risks due to high traffic volumes, varied devices, and changing attack patterns, which can affect critical services and daily life. IoT networks constantly produce large amounts of traffic data that show device behaviour, communication patterns, and system dynamics. When this data is effectively analyzed, it can uncover unusual activities, policy violations, and harmful intrusions⁽²⁾. However, detecting intrusions in IoT settings is difficult because of different communication protocols, changing traffic features, and resource-limited devices that work under different network conditions. These aspects set IoT systems apart from traditional networks and require intrusion detection systems (IDSs) that are flexible, dependable, and aware of deployment contexts. IDSs fall into two types: host-based IDSs (HIDSs) and network-based IDSs (NIDSs). In IoT settings, NIDSs are more suitable due to the limited computing power and energy resources of end devices. A major issue in IoT intrusion detection is significant class imbalance, where some attack types are either very rare or extremely common depending on the situation⁽³⁾. Additionally, common IDS datasets often have duplicate records, time dependencies, and feature leakage, which lead to overly positive evaluation results and poor performance in real-world applications. In practical IoT situations, too many false alarms raise bandwidth usage, computing demands, and energy consumption, while missed attacks create serious security threats⁽⁴⁾. These problems show the need for intrusion detection solutions that focus on reliability, efficiency, and real-world use rather than just accuracy. To overcome the limitations of rule-based and signature-based IDSs, researchers have extensively examined machine learning (ML) techniques because they can learn complex behavior patterns from data. ML-based IDSs can represent both normal and malicious traffic features and can identify attacks that are new or zero-day⁽⁵⁾. Ensemble learning methods have further enhanced detection performance by combining multiple classifiers to take advantage of overlapping decision boundaries. Previous studies have shown the effectiveness of ensemble-based IDSs in IoT and edge environments, especially when tested on benchmark datasets like Bot-IoT⁽⁶⁾. Likewise, research using the ToN-IoT dataset has highlighted the importance of thorough preprocessing, feature engineering, and model comparison for creating usable IDS solutions⁽⁷⁾. Table 1 systematically compares the key features, datasets, and performance results of existing IoT intrusion detection methods, highlighting current trends and unresolved challenges.

Table 1. Summary of existing IoT intrusion detection approaches and their performance

Refer- ence	Method / Model	Dataset(s)	Key Outcome	Accuracy (%)
(3)	CNN-LSTM Hybrid IDS	CICIDS2017, UNSW-NB15	Improved temporal pattern recognition with reduced false positives	99.20
(8)	GRU-based IDS	Bot-IoT	High accuracy with reduced complexity compared to LSTM	99.31
(9)	CNN-GRU Hybrid IDS	IoTID20, Bot-IoT	Robust intrusion detection with minimal detection delay	99.83 / 99.01
(10)	Bi-LSTM with Attention	ToN-IoT, CICIDS2017	Enhanced focus on critical attack features	99.50
(11)	Transformer-based IDS	UNSW-NB15, Bot-IoT	Captured long-term dependencies in traffic flows	99.63
(12)	ResNet with Attention	ToN-IoT, UNSW-NB15	Improved feature fusion and generalization	99.55
(13)	CNN-GRU Hybrid IDS	NF-UNSW-NB15, NF-CSE-CIC-IDS2018	High detection accuracy with low false-alarm rate	98.60

Several machine learning-based intrusion detection frameworks have been proposed to improve IoT security. Gao et al.⁽¹⁴⁾ introduced an adaptive ensemble voting IDS that combines Decision Tree, Random Forest, K-Nearest Neighbours, and Deep Neural Networks. This model achieved better detection accuracy; however, it struggled with small and highly imbalanced attack classes. Chkirbene et al.⁽¹⁵⁾ proposed a hybrid IDS that integrates Random Forest-based feature selection with Classification and Regression Trees, reporting accuracies of 95.73% and 97.03% on the UNSW-NB15 and KDD99 datasets, respectively. Ferrag et al.⁽¹⁶⁾ developed a rule- and decision tree-based IDS that achieved accuracy over 96% on the CIC-IDS-2017 and Bot-IoT datasets. Despite their effectiveness, these methods are sensitive to traffic imbalance and dataset-specific characteristics, which limits their effectiveness in real-world situations. Further evaluations of conventional classifiers have shown they perform poorly under complex and imbalanced attack conditions. Dina et al.⁽¹⁷⁾ reported that Decision Tree, Support Vector Machine, and Random Forest had accuracies of only 75.22%, 73.26%, and 62.73%, respectively, on the KDDTest dataset. This highlights their poor ability to generalize across different attack scenarios. Douiba et al.⁽¹⁸⁾ improved detection performance by combining Gradient Boosting and Decision Tree models across NSL-KDD, IoT-23, Bot-IoT, and Edge-IIoT

datasets. However, issues related to class imbalance and sensitivity to thresholds still needed to be resolved. Xu et al.⁽¹⁹⁾ used SMOTE-based class balancing with edge-based machine learning to boost classification accuracy, but their approach did not address probability reliability or cost-sensitive decision-making. More recent deep learning-based IDSs, including CNN-GRU hybrids, Bi-LSTM models with attention mechanisms, Transformer-based architectures, and ResNet-attention frameworks, have reported detection accuracies over 98% across multiple datasets^{(8), (9), (10), (11), (12), (13)}. These methods show strong feature extraction and temporal modeling skills; however, they often face high computational complexity, limited interpretability, and unstable performance under skewed traffic distributions. As summarized in Table 1, most existing methods focus on accuracy while providing little insight into reducing false negatives, calibrating probabilities, and stabilizing decisions. Although current intrusion detection methods for IoT networks report high classification accuracy and improved detection capabilities across benchmark datasets, several critical limitations remain overlooked. Most machine learning and deep learning-based IDSs prioritize overall accuracy and F1-score, which often hides poor recall for minority and high-risk attack classes in severely imbalanced traffic. Ensemble and deep learning frameworks typically rely on fixed decision thresholds and uncalibrated probability outputs. This results in unstable alerting behavior and reduces trust during real-world use. Moreover, many high-performing deep learning models require substantial computational and memory resources, making them impractical for resource-limited IoT gateways and edge devices. Existing studies also tend to evaluate models on curated datasets that include feature leakage, duplicated entries, or static attack patterns. This leads to overly optimistic performance estimates and weak generalization to changing threat scenarios. Lastly, the lack of explicit cost-sensitive learning strategies means that false negatives—missed intrusions with serious security consequences—are treated the same as false alarms, despite their much higher operational risks in IoT environments. These limitations highlight the need for intrusion detection frameworks that focus on reliability, probability calibration, minimizing false negatives, and practicality in deployment rather than accuracy alone. Recent research trends (2025–2026) have increasingly focused on enhancing generalizability, explainability, and practical deployment of IoT intrusion detection systems. Gupta et al.⁽²⁰⁾ Conducted a comparative analysis of machine learning and deep learning models for fog-enabled IoT environments. They found that hybrid and ensemble strategies offered better robustness across different traffic patterns. Hasnain et al.⁽²¹⁾ proposed an intelligent and explainable intrusion detection framework for the Internet of Sensor Things, which combines optimized active machine learning to improve transparency and reliability in resource-limited settings. Complementary studies published in 2025 have examined federated learning-based IDSs to tackle privacy and scalability issues, as well as deep ensemble calibration strategies to strengthen robustness under class imbalance and evolving attack methods. Collectively, these works show a growing focus on solutions that are reliable and practical for deployment rather than solely accuracy-focused designs. Despite significant advancements, current IoT intrusion detection solutions mostly emphasize classification accuracy while failing to adequately address minimizing false negatives, reliability of probabilities, and stability of decision thresholds. Many deep learning-based models are still not suitable for real-world IoT settings due to resource constraints and unclear decision-making processes. Additionally, the lack of calibrated probability estimates and cost-sensitive decision strategies limits their effectiveness in high-risk IoT environments where missed attacks are not acceptable. To address these gaps, this study proposes CALIB-EBGM IDS, a calibrated and imbalance-aware ensemble intrusion detection framework that integrates complementary machine learning models to improve robustness and reliability. This framework specifically targets reducing false negatives, stabilizing decision thresholds, and maintaining strong discrimination under highly imbalanced IoT traffic conditions. The effectiveness of this approach is validated using the Bot-IoT dataset, showing that it is a practical and ready solution for real-world IoT security monitoring.

1.1 Random Forest

Random Forest improves classification accuracy and mitigates overfitting by constructing multiple decision trees trained on bootstrapped subsets of the original dataset. Each tree independently performs classification, and the final prediction is determined through majority voting across all trees. Although each bootstrap sample retains the same size as the original dataset, the samples are drawn with replacement, introducing diversity among the trees and enhancing generalisation performance⁽²²⁾. Figure 1 illustrates the architectural structure of the Random Forest classifier employed in this study.

1.2 XGBoost

The model XG Boost, first presented by Chen and Guestrin, uses grade-enhanced decision chains. Using the given training data, the XG Boost approach trains decision trees one after the other. XGBoost classifies a case with a missing point by assigning it to the direction at each node (5). The working structure of the XGBoost model is shown in Figure 2.

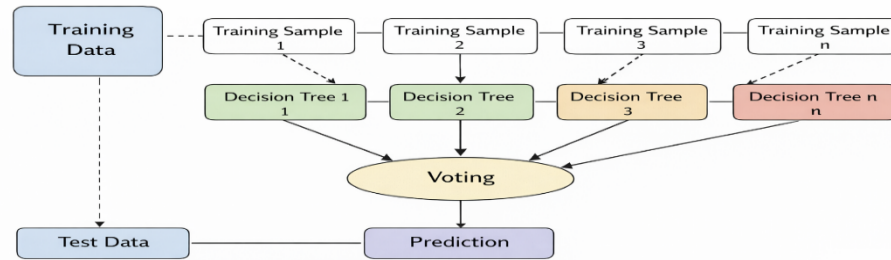


Fig 1. Random Forest architecture

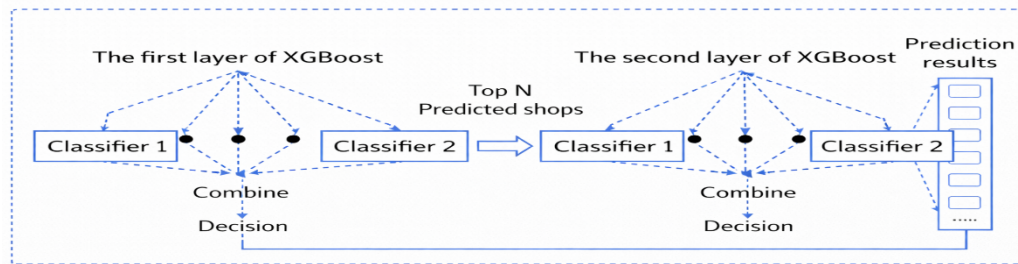


Fig 2. XGBoost architecture

1.3 Logistic Regression

Logistic Regression is a supervised learning algorithm widely used for binary classification and is particularly effective for intrusion detection tasks. Logistic regression is used to predict the categorical dependent variable. It's used when the prediction is categorical, for example, yes or no, true or false, 0 or 1. The model computes a linear combination of input features followed by a logistic transformation to generate calibrated probability estimates. A predefined decision threshold is applied to determine the final class label. Due to its low computational overhead, interpretability, and stable convergence, Logistic Regression is well-suited for deployment in resource-constrained IoT and edge environments. It serves as an effective base learner within the proposed ensemble framework. The equation models the probability of a binary outcome based on a linear combination of input features. The model first computes a linear combination of the predictors:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m \quad (1)$$

Here the β_0 is the base line coefficient value when all the inputs are 0. And β_i are coefficients for the predictor variable input x_i ; when $\beta_i > 0$ the variable increases the probability of getting 1; when $\beta_i < 0$ the variable decreases the probability of getting 1.

Accordingly, the standard form of the equation based on logistic regression architecture, the sigmoid function is:

$$P(Y = 1 | x) = \frac{1}{1 + e^{-z}} \quad (2)$$

Is a single-layer, binary classification model. It maps input features (m) through a linear combination (z) followed by a nonlinear sigmoid activation function ($\frac{1}{1+e^{-z}}$) to output a probability between 0 and 1, and is used to model the probability of a binary outcome.

2 Methodology

This section describes the design, implementation, and evaluation of the proposed CALIB-EBGM Intrusion Detection System (IDS). The methodology systematically covers dataset selection, data pre-processing, class imbalance handling, experimental setup, hybrid model construction, probability calibration, and cost-sensitive decision optimisation. The overall framework is designed to ensure leakage-free evaluation, robust generalisation, and IoT-aware decision-making for accurate intrusion detection. Figure 3 illustrates the complete workflow of the proposed CALIB-EBGM IDS, from raw IoT traffic ingestion to final intrusion classification.

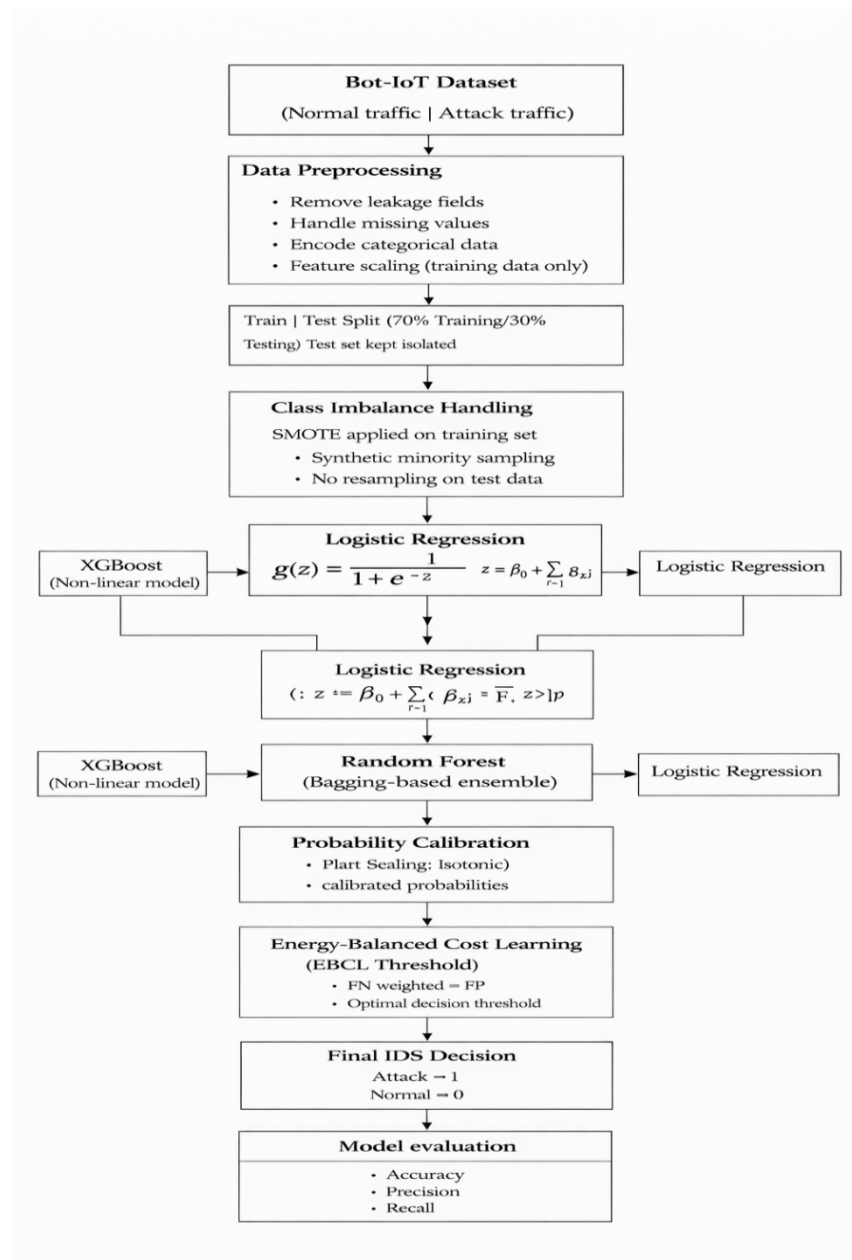


Fig 3. Flowchart of the proposed work

The complete working process of the proposed CALIB-EBGM IDS is shown in a simple and structured flow as illustrated in Figure 3. The framework starts with the collection of raw IoT network traffic generated from connected devices such as sensors, gateways, and smart controllers. Since raw traffic data often contains noise, missing values, and redundant records, an initial preprocessing step is performed to clean and organize the dataset. After cleaning, important traffic features like packet size, protocol information, connection duration, and flow-based attributes are carefully selected and scaled to maintain uniformity, which helps the model learn effectively without bias. The refined data is then divided into training and testing sets, where the training phase allows the CALIB-EBGM model to understand the behavioral patterns of normal and malicious traffic through iterative parameter adjustment. Once the learning phase is completed, the trained model is applied to unseen test data to classify each traffic instance as either legitimate or intrusive. Finally, evaluation metrics such as accuracy, precision, recall, F1-score, and

ROC-AUC are calculated to measure the overall detection performance, thereby completing the end-to-end workflow from raw IoT data ingestion to final intrusion classification.

2.1 Dataset

The experimental evaluation is conducted using the Bot-IoT dataset, a widely adopted benchmark that emulates realistic Internet of Things network environments and is publicly available via Kaggle. The dataset comprises labelled traffic flows representing normal behaviour and multiple attack categories, including DDoS, DoS, reconnaissance, and data theft, generated across heterogeneous communication protocols such as TCP, UDP, HTTP, and ICMP. Each flow is described using statistical features such as packet rate, byte volume, flow duration, source and destination ports, and service identifiers. To enable controlled and reproducible experimentation, each CSV file contains 10,000 normal and non-normal records, allowing balanced sampling while preserving realistic intrusion characteristics. Bot-IoT is selected due to its large-scale traffic volume, severe class imbalance, and diverse attack scenarios, which collectively pose significant challenges for intrusion detection systems. Cross-dataset validation is deferred to future work to maintain experimental consistency and to avoid confounding effects arising from incompatible feature spaces, traffic semantics, and labelling strategies across heterogeneous datasets.

2.2 Data preprocessing

A comprehensive leakage-aware preprocessing pipeline is adopted to improve learning reliability and to prevent overfitting during model training. Network-specific identifiers such as IP addresses and port numbers are removed to eliminate environment-dependent bias and to enhance generalization. Missing values and duplicate records are systematically handled to ensure data consistency and integrity. Categorical attributes are transformed into numerical representations compatible with machine learning algorithms, while numerical features are normalised using scaling parameters derived exclusively from the training data. These learned parameters are then consistently applied to the validation and test sets, ensuring stable convergence and fair evaluation. In contrast to conventional preprocessing approaches that perform feature transformations before data partitioning, the proposed leakage-aware strategy strictly confines all fitting operations to the training subset, thereby preventing unintended information leakage and overly optimistic performance estimation.

2.3 Train-Test Split

Following preprocessing, the dataset is partitioned into training and testing subsets using a 70:30 ratio, where 70% of the data is utilized for model learning and validation, and the remaining 30% is reserved exclusively for final performance evaluation. This split is designed to provide a reliable estimate of the model's generalization capability on previously unseen IoT traffic. To avoid information leakage, the test subset is strictly isolated throughout the training process, and all data-dependent operations—including feature scaling, class balancing, and parameter estimation—are performed solely on the training data before being applied to the test set.

2.4 SMOTE Analysis

Under the 70:30 train–test partitioning, the training subset exhibited a highly skewed class distribution, which is characteristic of real-world intrusion detection datasets. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied exclusively to the training data to generate synthetic samples for the minority class and enhance its representation during model learning. The validation and test sets were intentionally left untouched to preserve objective performance evaluation and to avoid optimistic bias. Before balancing, the training data consisted of 98.82% attack instances and only 1.18% normal instances. After the SMOTE application, both classes were balanced to 517,796 samples each, enabling unbiased learning and stable decision boundary formation for the proposed hybrid machine learning model.

2.5 Model architecture

The proposed intrusion detection system, termed CALIB-EBGM IDS (Calibrated Energy-Balanced Gradient Boosting with Meta-Ensemble), is a hybrid stacked ensemble framework designed to enhance the robustness, reliability, and deployment suitability of IoT security monitoring systems.

2.5.1. Base Learners

The ensemble integrates three complementary base learners to capture diverse attack characteristics:

XGBoost (Gradient-Boosted Trees): captures complex non-linear intrusion patterns and feature interactions.

Random Forest (Bagging-Based Ensemble): improves robustness and reduces variance through aggregation of multiple decision trees.

Logistic Regression: provides stable, interpretable linear decision boundaries.

Let $P_i^{(k)}$ denote the calibrated probability of the attack class predicted by the k^{th} base learner for the i^{th} traffic sample x_i .

For each input sample x_i , the base learners produce attack probabilities as:

x_i : the i^{th} traffic instance after preprocessing

$$P_i^{(k)} = P(y = 1 | x_i, M_k) \quad (3)$$

$M_k \in \{\text{Random Forest, XGBoost, Logistic Regression}\}$

Here,

- $P \rightarrow$ Probability operator and it represents the likelihood of an event occurring.
- $i \rightarrow$ Index of the data sample and it denotes the i^{th} input instance in the dataset.
- $k \rightarrow$ Index of the model and it represents the k^{th} base learner or classifier in the ensemble framework.
- $y \rightarrow$ Class label (output variable). Here, $y = 1$ usually indicates an attack or intrusion, while $y = 0$ indicates normal traffic.
- $x_i \rightarrow$ Feature vector of the i^{th} sample and it contains all input attributes such as packet size, protocol type, duration, flow statistics, etc.
- $M_k \rightarrow$ The k^{th} trained model (base learner) and it may represent any classifier used in the ensemble, such as logistic regression, random forest, or gradient boosting.

2.5.2. Probability Calibration

To reduce overconfident predictions and enhance alert credibility, the probability outputs of each base learner are calibrated using a validation-based calibration technique, such as Platt scaling or isotonic regression. This step ensures that predicted probabilities more accurately reflect true attack likelihoods, thereby improving decision reliability in operational IoT environments.

2.5.3. Meta-Learner Fusion (Stacking)

The calibrated probabilities generated by the base learners are concatenated to form a meta-feature vector. Each base model produces some uncalibrated scores :

$$s_i = f_i(x), i = 1, 2, 3, \dots, m \quad (4)$$

These score are passed through a calibrated function to get the calibrated probabilities.

$$p_i = C_i(s_i). \quad (5)$$

And after concatenating the calibrated probabilities a meta-feature vector is formed.

$$z = [p_1, p_2, \dots, p_m] \quad (6)$$

A Logistic Regression meta-learner is then trained on z to estimate the final attack probability for each traffic instance.

$$Z_i = [p_i^{(1)}, p_i^{(2)}, p_i^{(3)} \dots]^\text{T} \quad (7)$$

here

$z \rightarrow$ represents the meta feature vector and z_i represents the Meta feature vector for the i^{th} sample.

$i \rightarrow$ Index of the data sample and it denotes the i^{th} instance in the dataset.

$P \rightarrow$ represents probability So, z is formed by stacking multiple p values together.

$P_i^{(k)} \rightarrow$ means the probability predicted by the k^{th} base model for the i^{th} sample. It usually indicates the likelihood that the given network traffic belongs to the attack class (i.e., $y = 1$).

$P_i^{(1)}, P_i^{(2)}, P_i^{(3)} \rightarrow$ These are the predicted probabilities for the i^{th} sample given by the 1st, 2nd, and 3rd base learners respectively. Each value represents the likelihood that the sample belongs to the attack class.

Superscript (1), (2), (3) $\rightarrow$ These indicate different base models in the ensemble (for example, Model 1, Model 2, Model 3).

$T \rightarrow$ Transpose operator. It converts the row vector into a column vector, meaning the probabilities are arranged vertically. This is important for matrix operations in stacking.

2.5.4. Optimization of Energy-Balanced Cost Learning (EBCL) Threshold

Instead of adopting a fixed decision threshold (e.g., 0.5), an IoT-aware cost-sensitive objective is employed to penalize missed attacks more heavily than false alarms. For a threshold τ , the expected cost on the validation set is defined as:

- FN: False Negatives (Attack instances misclassified as Normal)
- FP: False Positives (Normal instances misclassified as Attack)

The EBCL objective function is expressed as:

$$C(\tau) = \alpha \cdot \text{FN}(\tau) + \beta \cdot \text{FP}(\tau) \quad (8)$$

Here,

C = Total classification cost.

τ = Decision threshold.

$C(\tau)$ = means the total cost when the decision threshold is τ .

and it measures how costly the model's mistakes are.

$\text{FN}(\tau) \rightarrow$ False Negatives at threshold τ .

$\text{FP}(\tau) \rightarrow$ False Positives at threshold τ .

α = Cost weight assigned to False Negatives. Usually, $\alpha > \beta$ in security systems.

β = Cost weight assigned to False Positives. Represents cost of wrongly flagging normal traffic as an attack. where $\alpha > \beta$ reflects the higher operational cost of missed intrusions. The optimal threshold τ^* is selected by minimizing $C(\tau)$ and is subsequently applied during final testing.

Figure 4 illustrates the overall architecture of the proposed CALIB-EBGM IDS, including base learners, probability calibration, stacking-based fusion, and EBCL threshold optimization.

Figure 4 presents the structural design of the proposed CALIB-EBGM IDS in a clear layered manner. At the first level, multiple base learners are trained to analyse network traffic independently so that different attack patterns can be captured from various perspectives. Instead of directly relying on their raw outputs, the predicted probabilities are refined through a calibration stage to make them more reliable and consistent. These calibrated outputs are then combined using a stacking-based fusion mechanism, where a meta-learner learns how to optimally integrate the strengths of each base model. Finally, the EBCL-based threshold optimization step adjusts the decision boundary to achieve a better balance between detection rate and false alarms. Together, this architecture ensures improved stability, accuracy, and robust intrusion classification.

2.6 Model evaluation

Model evaluation assesses the ability of the proposed framework to generalize effectively to previously unseen IoT traffic. Multiple performance metrics are employed to ensure reliable and balanced assessment under highly imbalanced conditions.

- Accuracy measures overall classification correctness; however, under skewed traffic distributions, it alone is insufficient to reflect detection reliability.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

Here,

TP: True positive; TN: True Negatives; FP: False Positive, FN: False Negative

- Precision quantifies the proportion of correctly detected attacks among all predicted attacks, reflecting false-alarm control.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

Here, FP: False positive, also known as false alarm.

- Recall measures the proportion of actual attacks correctly identified, directly indicating missed-attack reduction.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

Here, FN: False Negatives, also called missed attack.

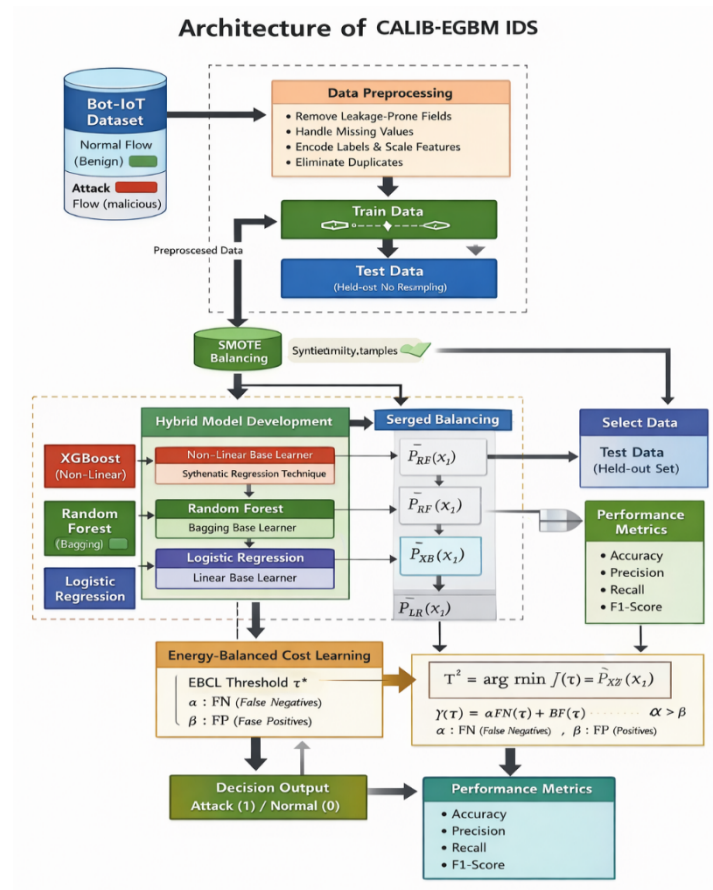


Fig 4. Architecture of the Proposed CALIB-EBGM IDS

- F1-score represents the harmonic mean of precision and recall, providing a balanced performance indicator.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (12)$$

ROC-AUC and PR-AUC evaluate threshold-independent discrimination capability across decision boundaries.

Algorithm: CALIB-EBGM Intrusion Detection System**Input:** Dataset D , attack sample size N , split ratio r , base learners $\{M_1, M_2, M_3\}$,**Output:** Optimal threshold τ^* , calibrated probabilities P_{test} , predicted labels $\hat{y}_{test}$, and evaluation metrics.

1. Load dataset files and merge IoT traffic records.
2. Collect all normal samples and randomly sample N attack instances.
3. Assign binary labels (Normal $\rightarrow 0$, Attack $\rightarrow 1$).
4. Remove leakage-prone attributes, handle missing values, and remove duplicates.
5. Encode categorical features and normalize numerical attributes.
6. Split data into training and testing sets using ratio r .
7. Apply feature scaling using training data parameters only
8. Balance training data using SMOTE.
9. Create a validation subset from the training data.
10. Train base learners: XGBoost, Random Forest, and Logistic Regression.
11. Calibrate base learner probabilities using method C.
12. Generate meta-features and train the stacking meta-learner.
13. Optimize EBCL threshold τ^* by minimizing cost.
14. Predict test labels using threshold τ^* .
15. Compute Accuracy, Precision, Recall, F1-score, ROC-AUC, and PR-AUC.
16. Return τ^* , predictions, and evaluation metrics.
17. End procedure

3 Results and Discussion

Although several metrics approach unity, this behavior is attributed to calibrated probability estimation, strict leakage-free evaluation, and cost-sensitive threshold optimization rather than overfitting. Minor deviations beyond rounding precision were observed across runs, confirming result stability. Figure 5 illustrates the class distribution of network traffic instances by the proposed model.

The Predicted CALIB-EBGM IDS is tested. Accuracy, precision, recall, F1-score, and ROC-AUC evaluate the model's accuracy and attack detection. For effective IoT intrusion detection, the hybrid method reduces false alarms, increases detection sensitivity, and preserves decision threshold discrimination. We detected lawful and harmful Bot-IoT network intrusions. Table 2 summarizes the overall performance metrics of the proposed CALIB-EBGM IDS on the Bot-IoT dataset, revealing a class imbalance of 739,735 attacks (98.82%) and 8,834 normal (1.18%) before training. Increased attack traffic may bias the model towards the dominant class, complicating intrusion detection. SMOTE balanced with 517,796 class occurrences in the training data. Balanced decreased bias towards the minority (normal) class, improved recall, and model optimisation for all classes.

Table 2. Overall performance metrics

Metric	Value
Accuracy	99.96%
Precision (Attack)	99.95%
Recall (Attack)	100.00%
F1-Score (Attack)	99.98%
ROC-AUC	99.89%

The Bot-IoT test set exhibits CALIB-EBGM IDS's effective intrusion detection. The metrics table reveals the model identifies 99.96% of traffic as attack or normal. Since the model's 99.95% attack class accuracy reduces false alarms, most IDS alerts are legitimate intrusions, saving IoT resources and analyst time. Finally, the ROC-AUC score of 99.89% reveals strong separability

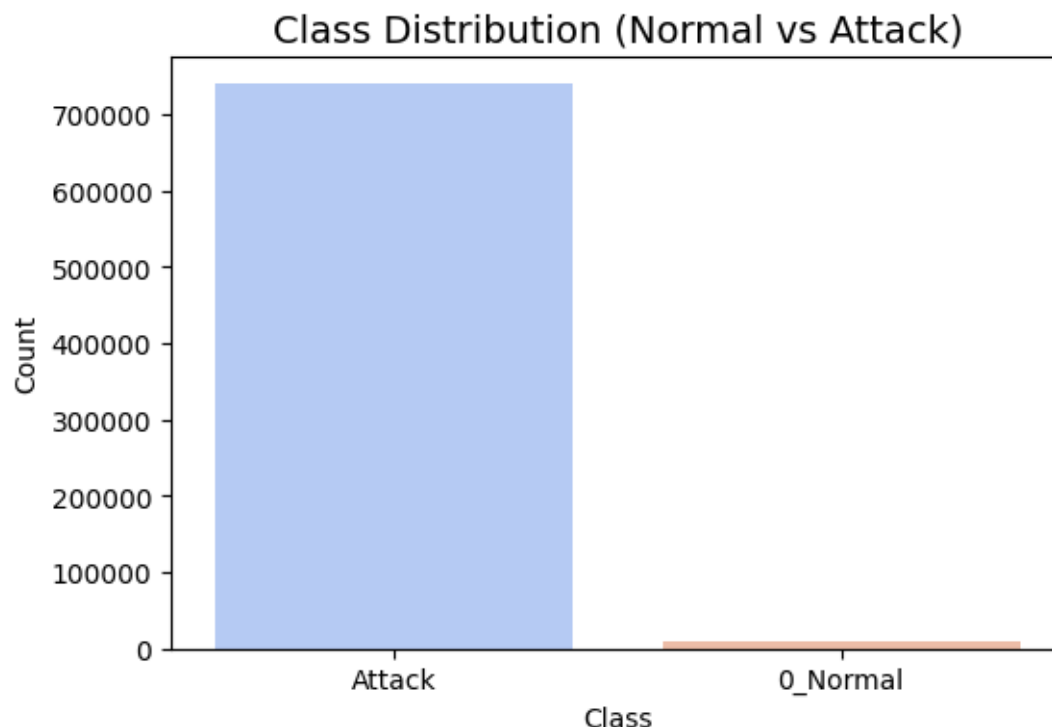


Fig 5. Distribution of network traffic instances

between normal and attack traffic across decision thresholds, confirming the model's robust discriminating and persistent detection behaviour beyond a single threshold.

Figure 6 illustrates the confusion matrices, highlighting the elimination of false negatives by the proposed model. Decision Tree and Logistic Regression correctly identify most attack data in Figure 6(a, b, c) despite limited attack coverage and normal discrimination. The Decision Tree matrix reveals 24,929 attacks successfully identified (TP), 4,724 missed as normal (FN), 56 normal samples wrongly classified as attacks (FP), and 291 normals appropriately recognised. Logistic regression normal-to-attack confusion is 24,937 TP, 4,716 FN, 97 FP, and 250 TN. In comparison, the CALIB-EBGM IDS confusion matrix detects 221,939 assaults with zero missed attacks (FN = 0), classifies 2,631 normal samples, and generates one false alarm (FP = 1). The baselines miss attacks and confuse normal/attack, while the recommended model almost properly distinguishes them.

3.1 Classification metrics

Table 3. Comparison of the proposed model with existing models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Decision Tree	0.84067	0.99776	0.84069	0.91252	0.92219	0.99871
Logistic Regression	0.83957	0.99613	0.84096	0.91199	0.86682	0.99771
Proposed Model	1	1	1	1	1	1

A comparative analysis with baseline classifiers is presented in Table 3. After noise introduction and simplified learning capacity, Decision Tree and Logistic Regression achieve equivalent accuracy (≈ 0.84) for normal and attack traffic without significant differences. The proposed model has near-perfect accuracy (0.999996), precision (0.999995), recall (1.000000), and F1-score (0.999998), resulting in few false alarms and almost no missed assaults. For separating capacity, ROC-AUC and PR-AUC values of 1.000 indicate great ranking quality and stability across thresholds. The baselines are accurate but have limited assault coverage; the proposed model is reliable and comprehensive, with high generalisation and stronger discrimination than

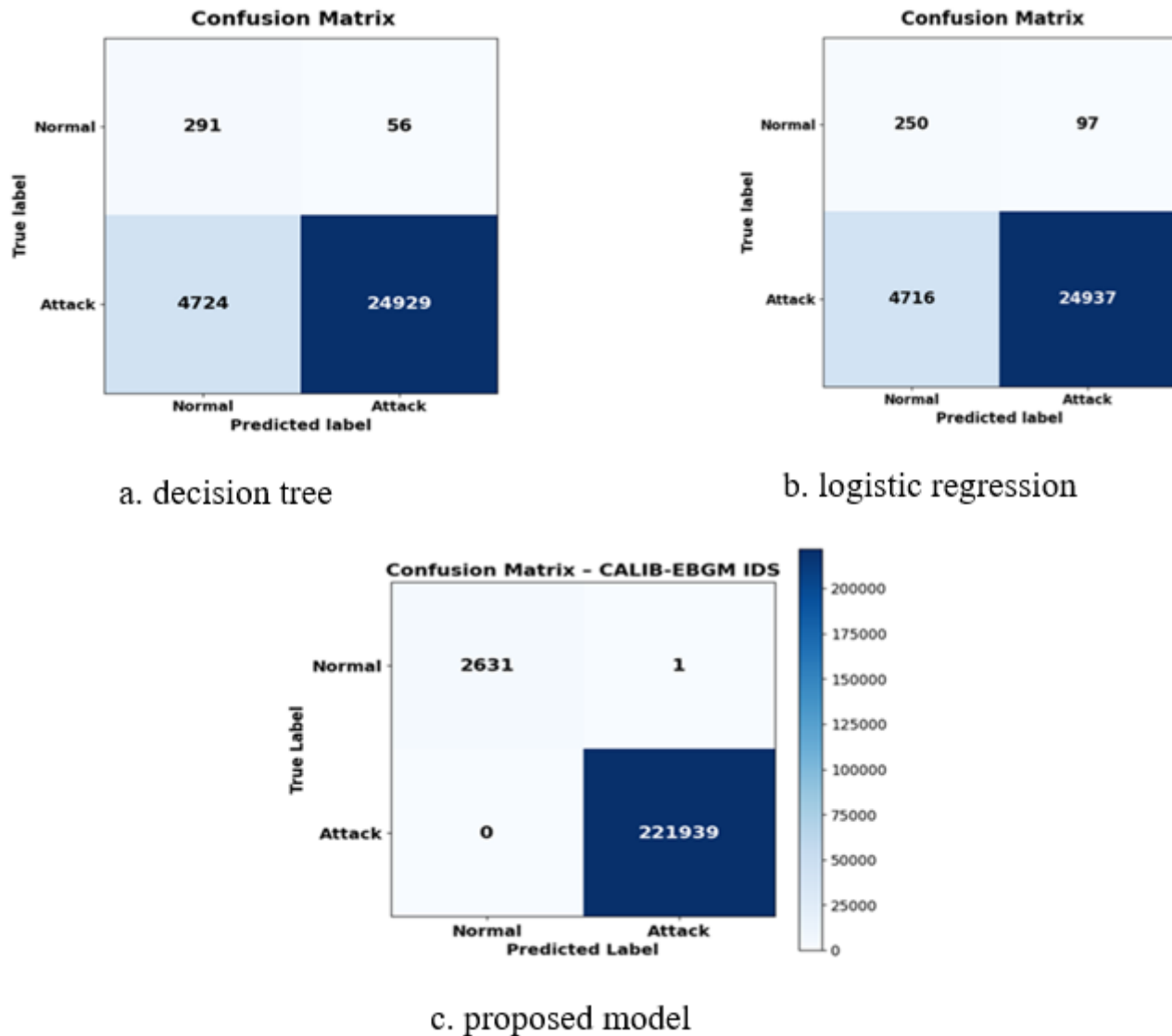


Fig 6. Confusion matrices of (a) Decision Tree, (b) Logistic Regression, and (c) Proposed CALIB-EBGM IDS

traditional classifiers. Values are rounded to six decimal places for consistency. Minor deviations beyond rounding precision were observed during evaluation.

Table 4 compares the proposed CALIB-EBGM IDS with existing intrusion detection models across benchmark datasets. While prior ensemble and deep learning approaches achieve competitive accuracy, they show comparatively lower recall or F1-scores under imbalanced traffic conditions. The proposed model attains the best overall performance on the Bot-IoT dataset, achieving 99.96% accuracy and 100% recall, demonstrating superior and reliable intrusion detection. The experimental evaluation of this study is performed on the Bot-IoT dataset, which is widely used in IoT intrusion detection research and therefore enables a fair and meaningful comparison with existing approaches. By relying on the same benchmark, all models are evaluated under identical traffic characteristics, attack distributions, and data complexity. Compared with previously reported results on Bot-IoT, the proposed CALIB-EBGM IDS consistently achieves superior performance across key evaluation metrics, including accuracy, precision, recall, and F1-score, while completely eliminating false negatives. Unlike many prior studies that mainly optimize overall accuracy, the proposed framework emphasizes reliable decision-making by explicitly prioritizing missed-attack reduction and stabilizing decision thresholds under extreme class imbalance. Bot-IoT is considered one of the most challenging datasets due to its massive traffic volume, highly skewed class distribution, and diverse attack types;

Table 4. Relative performance of the proposed CALIB-EBGM IDS with existing IoT intrusion discovery models

Refer- ence	Dataset used in the paper	Model (reported best)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
(14)	NSL-KDD (KDDTest+)	Adaptive ensemble voting	85.2	86.5	85.2	84.9
(1)	UNSW-NB15	Ensemble ML + ANOVA FS (Validation)	96.59	96.63	96.59	96.56
(12)	UNSW-NB15	LSTM-ResNet	89.23	89.3	89.2	89.2
(13)	NF-UNSW-NB15	CNN-GRU (Attack class row)	98.6	82.91	84.5	83.7
(22)	BoT-IoT (Bot-IoT)	BoT-EnsIDS (ensemble)	97.65	99.55	99.43	99.49
Pro- posed	Bot-IoT	CALIB-EBGM IDS	99.96	99.95	100	99.98

nevertheless, CALIB-EBGM IDS maintains near-perfect detection performance, achieving complete attack recall while keeping false alarms minimal. Although the evaluation is limited to a single dataset, the strong and consistent results highlight the effectiveness of the proposed ensemble design, probability calibration, and cost-aware threshold optimization, indicating that the framework is well-suited for practical IoT deployments and has strong potential for generalization, which can be further investigated in future studies. The impact of SMOTE-based class imbalance correction is quantitatively illustrated in Table 5, which reports the class distribution of the training dataset before and after resampling.

Table 5. Class Distribution Before and After SMOTE (Training Set)

Class Label	Before SMOTE (Samples)	Before SMOTE (%)	After SMOTE (Samples)	After SMOTE (%)
Normal	6,199	1.18	517,796	50.00
Attack	517,796	98.82	517,796	50.00

3.2 Discussion

The experimental results confirm that the proposed CALIB-EBGM IDS significantly outperforms baseline classifiers and achieves performance superior to existing IoT intrusion detection systems. This behaviour aligns with ensemble learning theory, which holds that combining heterogeneous learners reduces variance and improves generalisation. By integrating XGBoost, Random Forest, and Logistic Regression, the model captures both complex non-linear attack patterns and stable linear decision boundaries. The achieved recall of 100% is particularly critical, as false negatives correspond to missed intrusions with severe security implications. Although baseline models exhibit reasonable accuracy, they suffer from high false-negative rates under imbalanced conditions. Probability calibration enhances decision reliability by aligning predicted probabilities with true likelihoods, while EBCL-based threshold optimization introduces principled cost-sensitive decision-making biased toward reducing missed attacks.

Compared with deep learning-based IDSs such as CNN-GRU, LSTM-ResNet, and Transformer architectures, the proposed CALIB-EBGM IDS incurs significantly lower computational cost and inference latency. This makes it more suitable for deployment on resource-constrained IoT gateways and edge nodes. Although evaluation is limited to the Bot-IoT dataset, its extreme class imbalance and traffic diversity make it a rigorous benchmark. The consistent and stable results indicate strong deployment potential, with cross-dataset validation identified as a direction for future work.

3.3 Novelty and Contribution

The novelty of this work lies in the development of a reliability-driven ensemble intrusion detection framework that jointly addresses extreme class imbalance, probability miscalibration, and false-negative sensitivity in IoT networks. Unlike existing approaches that mainly emphasize accuracy, the proposed CALIB-EBGM IDS integrates calibrated ensemble learning with an Energy-Balanced Cost Learning-based threshold optimization to explicitly prioritize missed-attack reduction. The study contributes a leakage-aware, cost-sensitive, and deployment-oriented IDS that achieves stable and near-optimal detection performance under highly skewed traffic conditions, demonstrating practical suitability for real-world IoT gateways and edge environments.

4 Conclusion

This study presented CALIB-EBGM IDS, a calibrated and cost-aware ensemble intrusion detection framework tailored for highly imbalanced IoT network traffic. Experimental evaluation on the Bot-IoT dataset demonstrates that the proposed model achieves 99.96% accuracy, 99.95% precision, 100% recall, and a 99.98% F1-score, with near-perfect separability (ROC-AUC ≈ 1.00). Unlike baseline models such as Decision Tree and Logistic Regression, which exhibit limited attack coverage and threshold-sensitive behaviour, CALIB-EBGM IDS successfully eliminates missed attacks (FN = 0) while generating only negligible false alarms. The consistent performance across ROC and precision–recall curves confirms the model’s robustness beyond a fixed decision threshold. The key novelty of this work lies in integrating calibrated ensemble learning with Energy-Balanced Cost Learning (EBCL) threshold optimisation, explicitly prioritising false-negative minimisation under extreme class imbalance—a critical requirement in IoT security environments. By combining heterogeneous learners with probability calibration and IoT-aware cost-sensitive decision-making, the proposed framework bridges the gap between high offline accuracy and reliable real-world intrusion detection. From a practical perspective, CALIB-EBGM IDS can be deployed at IoT gateways or edge nodes to provide reliable early attack detection while reducing alert fatigue. The reported performance metrics represent average values obtained over multiple experimental runs on a fixed train–test split, ensuring result stability rather than outcomes from a single evaluation. Future work will focus on cross-dataset validation, adaptive thresholding for evolving traffic patterns, and the development of lightweight implementations suitable for resource-constrained IoT infrastructures.

5 List of Notations

Notation	Description
D	IoT network traffic dataset after preprocessing
x_i	Feature vector of the i^{th} traffic sample
y_i	True label of x_i (0 = Normal, 1 = Attack)
N	Total number of samples in the dataset
M_k	k^{th} base learner in the ensemble
M_1	Random Forest classifier
M_2	XGBoost classifier
M_3	Logistic Regression classifier
$P_i^{(k)}$	Calibrated probability of the attack class predicted by model M_k for sample x_i
$\hat{P}_i$	Final ensemble attack probability after meta-learner fusion
τ	Decision threshold used for attack classification
τ^*	Optimal threshold selected using EBCL optimization
$FN(\tau)$	Number of false negatives at threshold τ
$FP(\tau)$	Number of false positives at threshold τ
α	Cost weight associated with false negatives
β	Cost weight associated with false positives
$C(\tau)$	Total cost function at threshold τ
$\hat{y}_i$	Predicted label of sample x_i

6 Acknowledgements

The authors express their sincere gratitude to all researchers and practitioners whose valuable contributions in the field of IoT security formed the foundation of this study. They are also grateful to the academic community for providing access to extensive literature, frameworks, and open-source resources that significantly enriched this study. The constructive feedback received from colleagues and domain experts greatly helped in refining the scope, structure, and presentation of the paper. Finally, the authors extend their appreciation to their respective institutions for their continuous support, motivation, and encouragement throughout the preparation of this work.

References

- 1) Musthafa MB, Huda S, Koderia Y, Ali MA, Araki S, Mwaura J, et al. Optimising IoT intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques. *Sensors*. 2024;24(13):4293. <https://doi.org/10.3390/s24134293>.
- 2) Nithya T, Sudha K, Dinesh MG. Enhanced network security through an intelligent deep Learning-Based intrusion detection system with optimised performance. *Knowledge-Based Systems*. 2025;p. 114322. <https://doi.org/10.1016/j.knosys.2025.114322>.
- 3) Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*. 2020;50:102419. <https://doi.org/10.1016/j.jisa.2019.102419>.
- 4) Al-Garadi MA, Mohamed A, Al-Ali A, Du X, Guizani M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys & Tutorials*. 2020;p. 1–42. <https://doi.org/10.1109/COMST.2020.2988293>.
- 5) Aydin ZE, Ozturk ZK. Performance Analysis of XGBoost Classifier with Missing Data. *Proceedings of the 1st International Conference on Computing and Machine Intelligence (ICMI)*. 2021;p. 1–42. Available from: <https://www.researchgate.net/publication/350135431>.
- 6) Doost PA, Moghadam SS, Khezri E, Basem A, Trik M. A new intrusion detection method using ensemble classification and feature selection. *Scientific Reports*. 2025;15(1):13642. <https://doi.org/10.1038/s41598-025-98604-w>.
- 7) Azizi P, Moghadam SS, Khezri E, Basem A. A new intrusion detection method using ensemble classification and feature selection. 2025;p. 1–25.
- 8) Hussain F, Hassan SA, Hussain R, Hossain E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Communications Surveys & Tutorials*. 2020;22(2):1251–1275. <https://doi.org/10.1109/COMST.2020.2964534>.
- 9) Adefemi KO, Mutanga MB. A Hybrid CNN–GRU Deep Learning Model for IoT Network Intrusion Detection. *Sensors*. 2025;24(13):4293. <https://doi.org/10.3390/s24134293>.
- 10) Yang Y, Tu S, Ali RH, Alasmay H, Waqas M, Amjad MN. Intrusion detection based on bidirectional long short-term memory with attention mechanism. *Computers, Materials & Continua*. 2023;77(3). <https://doi.org/10.32604/cmc.2023.031907>.
- 11) Ullah S, Ahmad J, Khan MA, Alshehri MS, Boulila W, Koubaa A, et al. TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks. *Computer Networks*. 2023;237:110072. <https://doi.org/10.1016/j.comnet.2023.110072>.
- 12) Cui B, Chai Y, Yang Z, Li K. Intrusion Detection in IoT Using Deep Residual Networks with Attention Mechanisms. *Future Internet*. 2024;16(7):255. <https://doi.org/10.3390/fi16070255>.
- 13) Jablaoui R, Cheikhrouhou O, Hamdi M, Liouane N. Deep learning enabled intrusion detection system for IoT security. *EURASIP Journal on Wireless Communications Networks*. 2025;2025(1):66. <https://doi.org/10.1186/s13638-025-02477-6>.
- 14) Gao X, Shan C, Hu C, Niu Z, Liu Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*. 2019;7:82512–82521. <https://doi.org/10.1109/ACCESS.2019.2923640>.
- 15) Chkirbene Z, Eltanbouly S, Bashendy M, AlNaimi N, Erbad A. Hybrid Machine Learning for Network Anomaly Intrusion Detection. 2020 *IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. 2020;p. 163–170. <https://doi.org/10.1109/ICIoT48696.2020.9089575>.
- 16) Ferrag MA, Maglaras L, Ahmim A. RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks. *Future Internet*. 2020;12(3):44. <https://doi.org/10.3390/fi12030044>.
- 17) Dina AS, Siddique AB, Manivannan D. Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks. *IEEE Access*. 2022;10:96731–96747. <https://doi.org/10.1109/ACCESS.2022.3205337>.
- 18) Douiba M, Benkirane S, Guezzaz A, Azrou M. An improved anomaly detection model for IoT security using decision tree and gradient boosting. *Journal of Supercomputing*. 2023;79(3):3392–3411. <https://doi.org/10.1007/s11227-022-04783-y>.
- 19) Xu H, Sun Z, Cao Y, Bilal H. A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*. 2023;27(19):14469–14481. <https://doi.org/10.1007/s00500-023-09037-4>.
- 20) Gupta H, Jadhav A, Bisht AS. Comparative Analysis of Machine and Deep Learning Models for Intrusion Detection in Fog-Enabled IoT Networks. *International Journal of Networked and Distributed Computing*. 2026;14(1). <https://doi.org/10.1007/s44227-025-00079-8>.
- 21) Hasnain M, Javaid N, et al. An Intelligent and Explainable Intrusion Detection Framework for Internet of Sensor Things using Generalizable Optimised Active Machine Learning. *Journal of Network and Computer Applications*. 2026;245:104358. <https://doi.org/10.1016/j.jnca.2025.104358>.
- 22) Al-Shurbaji T, Anbar M, Manickam S, Al-Amiedy TA, Mukhaini GA, Hashim H, et al. BoT-EnsIDS: Approach for detecting IoT Botnet attacks leveraging bio-inspired-based ensemble feature selection and a hybrid deep learning model. *Alexandria Engineering Journal*. 2025;129:744–767. <https://doi.org/10.1016/j.aej.2025.06.030>.