Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.076413

ARTICLE

An Adaptive Intrusion Detection Framework for IoT: Balancing Accuracy and Computational Efficiency

Abdulaziz A. Alsulami[1,*], Badraddin Alturki[2], Ahmad J. Tayeb[2], Rayan A. Alsemmeari[2] and Raed Alsini[1]

[1]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[2]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

*Corresponding Author: Abdulaziz A. Alsulami. Email: aaalsulami10@kau.edu.sa

Received: 20 November 2025; Accepted: 26 January 2026; Published: 09 April 2026

ABSTRACT: Intrusion Detection Systems (IDS) play a critical role in protecting networked environments from cyberattacks. They have become increasingly important in smart environments such as the Internet of Things (IoT) systems. However, IDS for IoT networks face critical challenges due to hardware constraints, including limited computational resources and storage capacity, which lead to high feature dimensionality, prediction uncertainty, and increased processing cost. These factors make many conventional detection approaches unsuitable for real-time IoT deployment. To address these challenges, this paper proposes an adaptive intrusion detection framework that intelligently balances detection accuracy and computational efficiency. The proposed framework integrates mutual information (MI) feature selection model, deep contextual embeddings, and an adaptive decision mechanism. The MI model identifies and retains the most informative features, which reduces dimensionality while maintaining high detection accuracy. The adaptive decision dynamically selects between multiple inference paths to ensure that additional computation is needed only when the uncertainty level is high. Experimental evaluations on benchmark IoT datasets namely RT-IoT-2022, CIC-IoT-2023 and CIC-IoMT-2024 show that the proposed framework achieves F1-score of 99.92%, 96.66%, and 99.84%, respectively, with an average inference time of approximately 0.105 ms per sample. These results demonstrate that the framework effectively adapts inference complexity to data uncertainty, which provides an intelligent, interpretable and efficient solution for real-world IoT intrusion detection.

KEYWORDS: Adaptive IDS; cyberattack detection; computational cost; IoT security

1 Introduction

The Internet of Things (IoT) is an emerging technology that is already gaining acceptance and has been used in a variety of application areas including wearables, smart homes, healthcare services, agriculture, logistics, smart cities, and automotive [1]. Because it supports efficient communication of information across real-world objects and accomplishes this without the need of user intervention. IoT is a ubiquitous, self-organized network that has the ability to significantly change our way of interaction with others and our environment [2]. The ability to remotely monitor and manage the environment has been made feasible by the connectivity of IoT devices, which improves the ability to make decisions and thus improves the life of individuals and communities [3].

There are around 20 billion connected devices around the world, and by the end of 2030, there will be around 31 billion [4]. Additionally, by 2030, the IoT is projected to generate 684.1 billion dollars in income yearly worldwide [5]. Individuals, businesses and community sectors could benefit from the widespread implementation of IoT in several asymmetrical environments and application domains. While IoT devices are becoming more accessible, they often lack strong security features [6]. The existing constraints on IoT devices cause them to be a target for attackers, which leads to a range of attacks such as man-in-the-middle attacks, and network scanning, spear phishing, malware infiltration, eavesdropping, selective forwarding, keystroke logging, SQL injections and tampering, physical damage [7]. A Denial of Service (DoS) attack can be of the most harmful attack type to IoT devices as it restricts genuine users from reaching services. A cyberattack of this kind can cause considerable damage to IoT services and smart ecosystem applications that operate within a network of IoT devices. As a result, protecting IoT systems has become a growing challenge [8].

In the context of IoT, intrusion is defined as any attack, malicious activity, or unauthorized access that threatens the availability, integrity, or confidentiality of connected devices or data across consumer and enterprise networks [9,10]. Because IoT systems are diverse, resource-constrained and reliant on lightweight standards, they offer a broader attack surface since they are widely deployed in critical sectors such as production, energy, healthcare, and transportation [11,12].

Conventional security systems were initially created for static, standard, and high power systems, but they are often not effective in the constantly evolving attack environment of IoT [13]. The IoT infrastructures have constraints including processing power, memory and real-time responsiveness that differentiate them against traditional networks. They are therefore particularly vulnerable to adaptive cyber threats, which provide a risk and have the ability to compromise system security and interrupt with important operations of the company. Furthermore, the growing number of autonomous and sensitive IoT systems requires the use of creative, intelligent, and context-aware security mechanisms [14,15].

The security of IoT networks and systems requires intrusion detection systems (IDSs) and intrusion prevention systems (IPSs). These systems are necessary to protect data, prevent cyberattacks, and respond to security regulations [16]. Utilizing a variety of techniques, including anomaly detection, behavioral analysis, and investigating existing attack patterns, IDSs identify potential security breaches and immediately alert users of any threats to ensure rapid responses of security concerns [17]. Since that many IoT devices lack adequate security features, IPSs helps in network protection by examining data packets and preventing known threats. Security of IoT devices can also be ensured by IPSs, which address vulnerabilities, encrypt communication for added safety, and act as an essential defensive mechanism in the complicated IoT system [16].

Because IoT environments are dynamic, distributed and diverse, intrusion detection is a challenging but essential task [18]. Since these platforms comprise a variety of devices with different processing capabilities, protocols of communication and data formats, which makes performing cybersecurity analysis a challenging task in such contexts. Furthermore, the large amount of real-time data and the absence of annotated attack instances particularly for undetected attacks, which limits the ability to identify threats in an adaptive way [19]. Class imbalance, noise and low-latency computing requirements make it more difficult to use traditional deep learning (DL) techniques that emphasizes the need for more specialized, flexible and resource-effective methods [20].

Sensor data reliability is a critical aspect in IoT security systems because IDS performance is heavily dependent on the quality and accuracy of data generated by heterogeneous sensors such as vibration, temperature, humidity and motion sensors. Faulty, noisy, drifting or spoofed sensor readings can introduce false patterns into the network, which affects both anomaly detection accuracy and model stability.

Recent studies like [21] highlight that unreliable sensors significantly degrade the trustworthiness of IoT operations and can even mimic intrusion signatures. Therefore, incorporating reliability assessment or fault tolerant preprocessing techniques is essential for robust intrusion detection.

Existing machine learning (ML) and DL based intrusion detection systems are resilient and provide high accuracy in classification in conventional Internet environments. However, existing IDS methods can consume more resources and can be considered as unsuitable for IoT [22]. Furthermore, the latest developments in ML based IDS require more resources and computing power which can help improve accuracy. As a result, these methods need to be resource effective to be appropriate for IoT devices with limited resources [23]. Due to all the challenges that are above mentioned, there is a need to have an adaptive and cost effective IDS in IoT environments.

Our study aims to address the challenges of IoT environments characterized by limited computational resources, high feature dimensionality, and heterogeneous architectures. To overcome these limitations, we propose an adaptive intrusion detection framework that integrates mutual information (MI) model for feature selection, calibrated random forest classifiers, and deep contextual embeddings from the efficiently learning an encoder that classifies token replacements accurately (ELECTRA) transformer. The framework employs a Linear Upper Confidence Bound (LinUCB) algorithm as an adaptive decision mechanism that dynamically selects between multiple inference paths, effectively balancing detection accuracy and computational efficiency.

Our main contribution in this work is summarized as follows:

- We propose an adaptive IDS framework that dynamically balances detection accuracy and computational efficiency for IoT environments, supported by a mutual-information (MI) feature selection model that reduces dimensionality by removing redundant and less-informative features while preserving important content.
- We implement a parallel pipeline architecture using random forest classifiers: a Base Model trained on all features and a Reduced Model trained on the top MI-selected features. This setup enables adaptive decision-making during inference.
- We integrate a pretrained ELECTRA language model to generate deep contextual embeddings, which are combined with a logistic regression Teacher Probe to refine predictions under high uncertainty and enhance the understanding of complex IoT traffic patterns.
- We use a contextual LinUCB bandit algorithm to select among four inference actions namely Reduced Model, Base Model, large language model (LLM) Refinement and Escalation, which optimizes the trade-off between accuracy and computational cost. The framework is evaluated on three IoT datasets—RT-IoT-2022, CIC-IoT-2023, and CIC-IoMT-2024—achieving F1-scores of 99.94%, 94%, and 98%, respectively, with significant reduction in computational overhead.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the proposed method, including the system architecture and feature extraction. Section 4 presents the experimental results, compares them with existing benchmark models, and discusses the findings. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Work

In this section, we discuss an overview of recent related works that design advanced IDS models for IoT setting. Then, we highlight the used datasets, learning methods, feature selection techniques, processes, and findings. The discussion emphasizes the way that the previous research has attempted to improve detection accuracy, computational effectiveness, and flexibility of integrating the DL and ML methods. Table 1 discusses

recent IDS methods in IoT environments including datasets utilized, approaches and techniques used, the accuracy achieved, the computational cost and research gap.

The authors in [24] proposed an anomaly IDS architecture that uses a Multi-Layer Perceptron (MLP) classifier on the reduced set of features and a combined feature selection strategy to find the optimal features for the detection of IoT attacks. They used information gain, gain ratio, correlation-based feature subset selection, pearson's correlation analysis, symmetric uncertainty and MLP as a classifier. They evaluated their model using the RT-IoT-2022 dataset. The MLP achieved an accuracy rate of 93.5% with the original data with 83 features and 96.4% with all FS methods with 16 features. The study in [25] improved intrusion detection by introducing a semi supervised learning technique that uses a self-training loop and uncertainty filtering based on entropy. They have used XGBoost, random forest, extremely randomized trees (XRT), gradient boosting classifier (GBC), decision tree (DT), and entropy-based dynamic threshold filtering with self-training. In their evaluation, they used three datasets namely RT-IoT-2022, CICIoT-2023, and CICIoMT-2024 datasets. They achieved the highest overall accuracy of 100% with DT using the RT-IoT-2022 dataset, and the highest overall accuracy of 93% with XGBoost and XRT using the CICIoT2023 dataset, and achieved accuracy of 98% with random forest and XGB using CICIoMT2024. The researchers in [26] introduced an approach to maximize feature dimensionality for real-time intrusion detection in IoT contexts by integrating five feature selection techniques including gain ratio, correlation based feature subset selection, Pearson analysis, information gain, and symmetrical uncertainty with PCA and classifiers used such artificial neural networks (ANNs), deep neural networks (DNNs) and TabNet. They evaluated their approach using the RT-IoT-2022 dataset. They have achieved the highest accuracy rate of 99.7% by using Pearson–PCA with ANN that are applied on the RT-IoT-2022 dataset. According to the comparison table provided by the authors that their work outperformed other related works in the field. The work in [27] presented a method named cosine similarity based majority class reduction (CSMCR). It reduces duplicated majority class instances while maintains the integrity of the dataset. Despite traditional methods like synthetic minority over sampling technique (SMOTE) or random undersampling, they used CSMCR to ensure the retained majority samples remain diverse by evaluating similarity across features. This reduces information loss and avoids unnecessary data duplication. In addition, they created a hybrid DL model that improved feature extraction and classification performance by integrating the RegNet and FBNet architectures. In the evaluation process, they used four datasets including IoTID20, N-BaIoT, RT-IoT-2022, UNSW Bot-IoT. Their model achieved accuracy rates of 96.18% on the IoTID20 dataset, 100% on N-BaIoT, 98.40% on RT-IoT-2022, and 91.10% on UNSW Bot-IoT.

Table 1: Summary of related IDS approaches in IoT.

Ref.	Year	Methods	Datasets	Accuracy	Computational Cost	Research Gap
[24]	2024	MLP+ FS (IG, GR, CFS, Pearson, SU)	RT-IoT-2022	96.4%	Runtime reduced by 66.4% via feature selection	Uses static set of 16 features and lacks an adaptive decision layer
[26]	2023	FS + PCA + ANN/DNN/ TabNet	RT-IoT-2022	99.7%	Lightweight computation via reduced features from 83 to 5	Uses static dimensionality reduction (83 to 5 features)

(Continued)

Table 1 (continued)

Ref.	Year	Methods	Datasets	Accuracy	Computational Cost	Research Gap
[27]	2025	CSMCR + RegNet + FBNet	IoTID20, N-BaIoT, RT-IoT-2022, Bot-IoT	91%–100%	Training time reduced by 53% via CSMCR	High computational costs without a cost-aware mechanism
[28]	2024	Lightweight Host IDS + XGBoost + Entropy	CIC-IoT-2022, Bot-IoT	99.97%	99.7% reduction in processing time; 86.4% in memory	Primarily host-based
[29]	2024	CNN+LSTM Stacked + Random Forest Meta-Classifier	CIC-IoT-2022 + NF datasets	91%–99.99%	Average inference time of 1.5 ms	High inference latency
[30]	2024	CNN-LSTM + Random Forest + Transfer Learning + SHAP/LIME	CIC-IoT-2022, CIC-IoT-2023, Edge-IIoT	98.2%	Handled big data by Spark optimization and reduced the size of the network by data preprocessing techniques	Lacks an intelligent bandit-based controller for real-time resource optimization
[31]	2025	Incremental Learning + Blockchain + Encryption	CIC-IoT-2023, CIC-DDoS2019, TON-IoT2020	98%–99.89%	Low encryption and decryption overhead 0.0001 s–0.0003 s and transaction time 2–3 s	Blockchain integration introduces high latency
[32]	2025	RF, DT, KNN, SVM, AdaBoost	CIC-IoT-2023	99.125%	Optimized for resource efficiency via feature selection	Evaluation is limited to a single dataset
[33]	2025	Correlation + SFS + Cascaded LSTM + NB	CIC-DDoS2019, CIC-IoT-2023, CIC-IoV-2024	99.7%–99.9%	Fastest intrusion detection time is 0.041 s	Employs a static hybrid classification model

(Continued)

Table 1 (continued)

Ref.	Year	Methods	Datasets	Accuracy	Computational Cost	Research Gap
[34]	2025	RF, DT, XGB, FNN	CICIoT-2023, CICIoMT-2024	99.85%	Resource efficiency by optimizing the dataset and hyperparameter tuning	Lacks an intelligent decision layer
[35]	2023	XGBoost + SHAP + Voting Fusion	General IoT threat data	97%	Bandwidth reduced by 73% and detection latency below 200 ms	Relies on post-hoc SHAP values for interpretability
[36]	2023	UNet++ + LSTM for IoMT	CICIoMT-2024	87.96%–99.92%	The inference speed is 166.12 µs per sample and the model size is 8.04 MB	Uses a complex UNet++/LSTM stack that is computationally expensive and difficult to reduce

The article in [28] proposed a lightweight host-based intrusion detection system that can characterize communication activities with a variety of entropies. They used methods namely XGBoost, host-based intrusion detection systems, data aggregation and several entropies. They evaluated their methods using the Bot-IoT Dataset and the CIC IoT 2022 Dataset. The accuracy rating achieved by their approaches was 99.97%. According to the findings, the proposed approach can reduce the processing time by 99.7% (2916 ms) and memory usage by 86.4% (633 MiB). The study in [29] proposed a stacking ensemble IDS model called (StaEn-IDS). A CNN and a LSTM model are integrated with a Deep Intrusion Detection System (DIDS). To increase accuracy, the random forest algorithm is used as a meta-classifier throughout the stacking procedure. Their StaEn-IDS was evaluated toward other models in the literature using a range of datasets including NF-ToN-IoT, NF-BoT IoT, NF-UNSW NB15, NF-CSE CIC IDS 2018 and NF-UQ NIDS. They achieved accuracies of 91.58% on NF-ToN-IoT, 95.82% on NF-BoT IoT, 99.49% on NF-UNSW NB15, 98.44% on NF-CSE CIC IDS 2018, 97.72% on NF-UQ NIDS and approximately 99.99% on CIC-IoT IDS 2022. The results show promising accuracy among all benchmark datasets. The authors in [30] present an enhanced IDS for IoT security that uses transfer learning and multimodal big data representation. They have used LIME and SHAP, Deep IDS (DIDS), CNN and LSTM as base learners. In addition, they utilized random forest as a meta-classifier and for Feature Selection they used Correlation and MI. The CIC-IoT 2022, CIC-IoT 2023 and Edge-IIoT are three benchmark IoT-based datasets that are utilized to evaluate the proposed approach. The proposed CNN-LSTM method achieves an accuracy of 98.2% on CIC-IoT dataset 2022.

The researchers in [31] used blockchain, hybrid encryption and incremental learning in three steps to create an integrated framework that secures and enhances IDS performance in the IoT. First, a model was created using incremental learning including SGD Classifier to identify cyberattacks. In the second step, the model has the ability to retrain itself on new data. The information has signed and encrypted in

digital manner. The encrypted data is stored using a blockchain model. The incremental model is retrained using new data taken from the blockchain in the third step. They used CIC-IoT-2023 dataset to train their model. After retraining, the model evaluated with data that can be considered as unseen data for the model namely CIC IDS2019 and TON IoT2020 datasets. Their method achieved accuracy rates of 99.7% on CIC-DDoS2019 dataset, 98.27% on TON-IoT2020 dataset and 99.89% on CIC-IoT-2023 dataset. The research in [32] introduces an IDS method that improves cyber threat recognition and mitigation in smart cities by using ML. They created and evaluated a number of models including random forest, decision tree, k-nearest neighbors (KNN), support vector machine (SVM), and adaptive boosting (AdaBoost) by utilizing the CIC-IoT-2023 Dataset. Their proposed IDS system focuses on real-time detection of threats and ensures to obtain a high accuracy and minimal latency. Their solution performs well in cyberthreat detection and prevention due to robust data preparation and precise model training. The results of this study demonstrate the power of using AI techniques by presenting significant improvements in privacy and security of smart city in IoT architectures. They achieved an accuracy rate of 99.125% with the KNN model in binary classification. The authors in [33] proposed a hybrid DL and ML approach that helps to identify DDoS and spoofing threats, lower false alarms and then put the required security in action. The first step of the proposed approach is to provide a feature selection that uses a correlation coefficient and a sequential feature selector. The second step involves a combination of DL neural networks with a cascaded LSTM and a Naive Bayes classifier. In the third stage, ports are restricted and network security measures are enhanced. The three datasets namely CIC-DDoS2019, CIC-IoT2023 and CIC IoV2024 were used in training and evaluating the performance of the proposed method and were balanced to provide useful outcomes. The method achieved an accuracy rates of 99.91% on CIC-DDoS 2019, 99.88% on CIC-IoT 2023 and 99.77% CIC IoV 2024 dataset. The test data was validated with cross-validation technique to ensure there was no overfitting.

The authors in [34] compared the performance of ML models that were trained on two datasets namely CICIoT2023 and CICIoMT2024, and they emphasized the importance of using domain specific data. Their results show that when models trained on single dataset are evaluated on another, the F1-score decreased significantly by around 66.87%. They evaluated in CICIoMT2024 and suggested baseline optimization methods such as appropriate train,validation and test splits, uniform windowing, modifying temporal dependencies for time series data and balance the imbalance dataset. For intrusion detection, they used three tree based ML classification methods: extreme gradient boosting (XGB), random forest (RF) and decision support (DT). They extended the scope of the research and emphasized the performance differences that a DL model could have in their situation, they also included a Feedforward Neural Network (FNN). Compared to other methods in the study, they achieved an accuracy rate of 99.85% and considered this a notable gain in IDS performance. The study in [35] proposed a high-performance cybersecurity framework that is interpretable and uses a precisely tuned XGBoost classifier to identify malicious attacks with high prediction accuracy. Their study compares the proposed approach with a baseline of regular logistic regression. They used SHAP (SHapley Additive exPlanations) to find important features that influenced predictions and examined the security cost trade-off while building ML systems for threat detection. They introduced a fusion technique based on max voting to effectively combine the advantages of the models. XGBoost outperforms Logistic Regression in terms of accuracy by achieving 97% and recall 100%, the results showed that the fusion model provided a more balanced performance with enhanced precision 98% and fewer false negatives. The researchers in [36] presented a DL method for analyzing Internet of Medical Things (IoMT) traffic and detecting malicious activity. They enhanced the prediction accuracy of current neural network models by using the CIC IoMT dataset. LSTM models and UNet++ are combined in their method to efficiently extract network traffic characteristics. The proposed approach outperformed conventional

algorithms, according to experimental data, they achieved an accuracy rate of 87.96% in attack classification and 99.92% in anomaly detection.

Most existing studies on IDS for the IoT focused on improving performance by optimizing classifiers through techniques such as undersampling including CSMCR and dimensionality reduction such as PCA. However, these approaches have not fully explored the integration of deep contextual embeddings or adaptive decision mechanisms, such as bandit-based controllers. The reviewed papers employ ensemble tree models such as random forest, XGBoost or DL architectures like ANN, DNN, RegNet, FBNet and CNN without leveraging deep contextual embeddings such as ELECTRA for feature representation or utilizing adaptive selection strategies including LinUCB for dynamic model control. While existing studies achieve high accuracy, they typically rely on static computational costs, lack semantic context from deep embeddings and provide no adaptive paths for uncertain cases.

However, our study introduces an adaptive intrusion detection framework that works in IoT environments. The framework dynamically adjusts its behavior based on contextual factors such as computational power, classification accuracy, and security requirements. The adaptability behavior of our framework continuously optimizes the performance in real time by using deep contextual embeddings for feature representation and a bandit-based controller for classifier selection. This helps maintain high detection accuracy under the computational constraints of IoT devices.

3 Methodology

In this section, we present the research methodology. We propose an adaptive intrusion detection framework for IoT environments that integrates feature learning, deep embeddings, and intelligent decision control. Within this framework, multiple models run in parallel, allowing the system to choose dynamically among four possible actions: Reduced Model, Base Model, LLM Refinement, or Escalation based on current conditions. This helps to balance high detection accuracy with computational efficiency. The framework uses an MI-based feature selection model with dynamic cumulative fraction policies to keep the most informative features. This reduces redundancy while preserving essential information. Calibrated random forest classifiers operate in both base and reduced configurations. The ELECTRA embedding pipeline is combined with a logistic regression (Teacher Probe) to refine predictions in uncertain cases. During inference, a LinUCB algorithm dynamically chooses the best action to balance accuracy and computational cost.

3.1 Proposed Framework

The overall architecture of the proposed adaptive intrusion detection framework is illustrated in Figs. 1 and 2, which depict the training pipeline and the adaptive inference and decision-making process, respectively.

As shown in Fig. 1, the framework begins with data preprocessing, where the dataset is divided into training, validation, and test subsets. Two parallel pipelines are then constructed for model training. In the first pipeline, MI with a cumulative fraction policy is applied to rank features according to their information contribution. Only features that collectively explain 90% of the total information are retained, and these features are used to train the Reduced Model, a lightweight random forest classifier designed to minimize computational cost during inference.

In parallel, the second pipeline trains the Base Model using the complete feature set. This model is implemented as a calibrated random forest to ensure reliable probability estimates, which are critical for downstream uncertainty assessment. In addition to these pipelines, both training and test samples are transformed into dense vector representations using the ELECTRA model [37]. These embeddings

capture richer contextual relationships in the IoT traffic data and are used to train a logistic regression classifier, referred to as the Teacher Probe. The Teacher Probe serves as a refinement component that assists decision-making when model uncertainty is high.

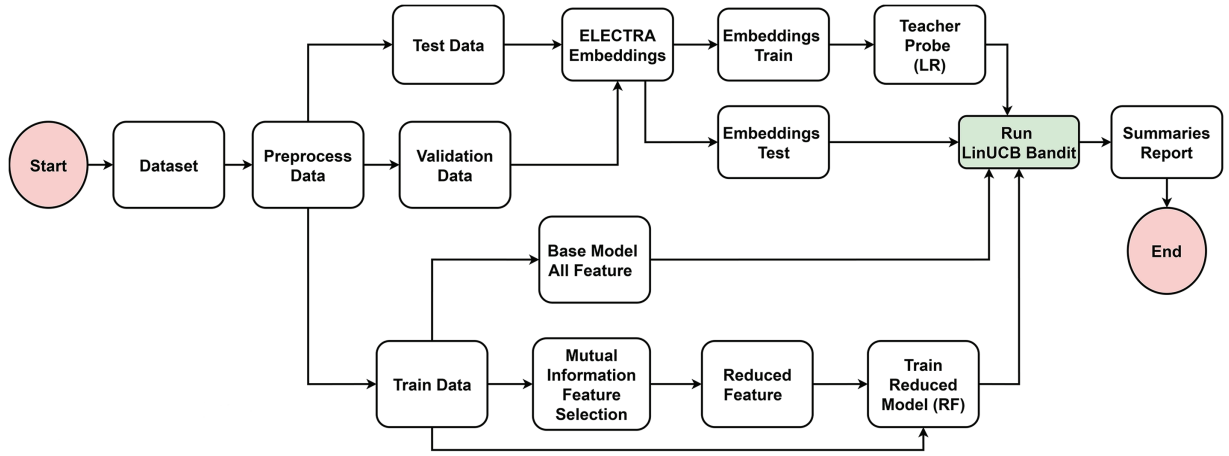


Figure 1: Architecture of the proposed IDS showing data preprocessing, MI-based feature selection, and training of the reduced and base models.

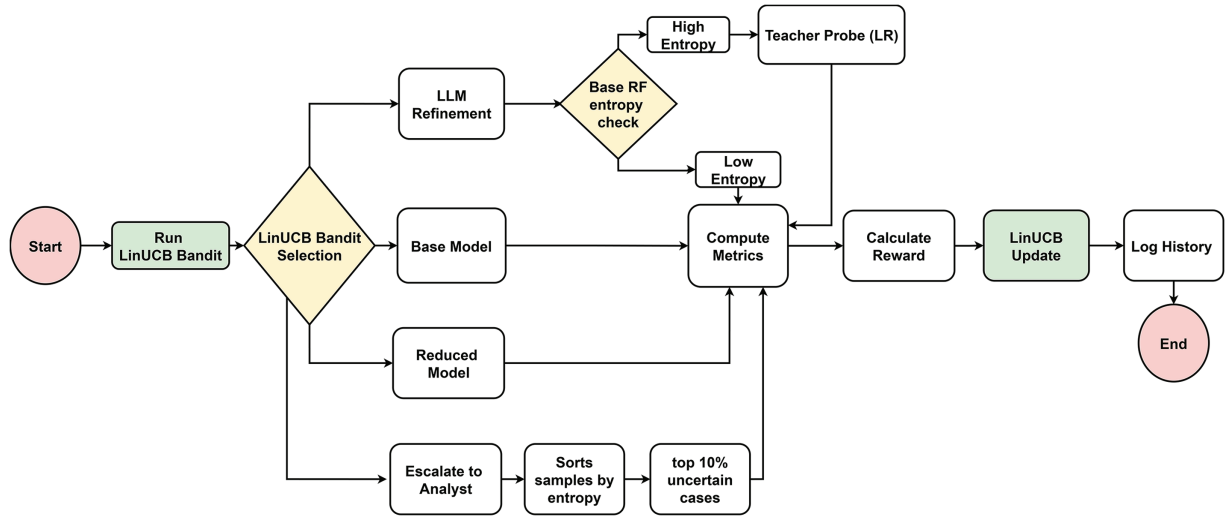


Figure 2: Adaptive inference workflow using a LinUCB controller to select inference actions based on uncertainty and computational cost.

Fig. 2 illustrates the adaptive inference process during testing. Instead of relying on a single static classifier, the framework employs a LinUCB contextual bandit to dynamically select the most suitable inference action for each batch of test samples. This ensures a balance between detection accuracy and computational cost. For each batch, a context vector is constructed using batch-level statistics, such as the mean and standard deviation of the input features, and provided to the LinUCB controller.

Based on this context, the controller selects one of four possible actions: (i) the Reduced Model for efficient low-cost inference, (ii) the Base Model when higher reliability is required, (iii) LLM-based refinement using the Teacher Probe when uncertainty remains high, or (iv) escalation for manual review

under extreme uncertainty. Each selected action produces a prediction output, which is evaluated using accuracy and F1-score metrics.

The selected action, prediction performance, and associated computational cost are then used to compute a reward signal, defined as performance minus action cost. This reward is used to update the LinUCB controller, completing the feedback loop. Through this iterative process, the controller learns to select inference actions that achieve an optimal trade-off between accuracy and efficiency under varying operational conditions.

The overall operation of the proposed framework is summarized in Algorithm 1, which presents the end-to-end training and adaptive inference procedure.

Algorithm 1: Adaptive intrusion detection framework

Input: IoT dataset D, cumulative MI fraction threshold τ, batch size B
Output: Predicted labels $\hat{y}$ for test samples
Process:

Split dataset D into training, validation, and test subsets
Rank features using MI
Select top features using MI based on the cumulative fraction threshold τ
Train Reduced Model M_r using selected features
Train Base Model M_b using full feature set
Generate ELECTRA embeddings and train Teacher Probe M_t
for each test batch B **do**
    Construct context vector c from batch-level statistics
    Select inference action a using LinUCB controller given context vector c
    if a = Reduced Model **then**
        $\hat{y} \leftarrow M_r(B)$
    else if a = Base Model **then**
        $\hat{y} \leftarrow M_b(B)$
    else if a = Refinement **then**
        $\hat{y} \leftarrow M_t(B)$
    else
        Escalate batch for manual review
    end if
    Compute reward as performance minus action cost
    Update LinUCB controller
end for

3.2 Proposed Hybrid IDS Components

3.2.1 SMOTE

Balancing the classes of an imbalance dataset is a crucial step to eliminate biased decisions caused by major classes over minority ones. In this work, SMOTE is applied only to the training subset in order to avoid information leakage and preserve the integrity of the validation and test distributions. As illustrated in Fig. 3, the process begins with loading the training set, then class distribution is examined to detect imbalance between majority and minority classes. The dataset is then separated into features x and labels y, and SMOTE is applied to generate synthetic samples for the minority classes. This step balances the dataset by ensuring that all classes are more evenly represented. The resampled features and labels are recombined

into a new balanced dataset, and the class distribution is rechecked to confirm the effect of SMOTE. Finally, the balanced dataset is saved as a CSV file for use in subsequent model training. This process ensures that the learning algorithm does not become biased toward the majority class and can achieve more reliable and fair performance across all classes.

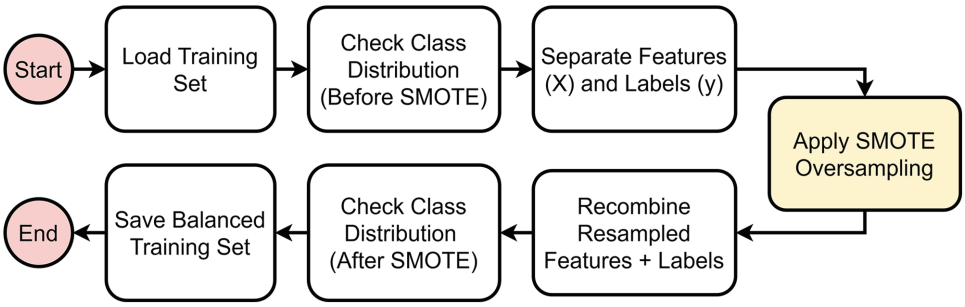


Figure 3: Data preprocessing workflow.

3.2.2 Mutual Information (MI) Feature Selection

MI between a feature X_j and the label Y is:

$$I(X_j; Y) = \sum_{x_j \in X_j} \sum_{y \in Y} p(x_j, y) \log \frac{p(x_j, y)}{p(x_j)\, p(y)} \tag{1}$$

Eq. (1) defines the standard MI feature selection, which measures how much knowing a feature X_j helps in predicting the class label Y. In simple terms, MI evaluates the dependency between a feature and the label. The term $p(x_j, y)$ represents the joint probability of the feature value and the label occurring together. However, $p(x_j)$ and $p(y)$ represent their individual probabilities. Therefore, X_j and Y are independent, then $p(x_j, y) = p(x_j)p(y)$, and the logarithmic term becomes zero. This means that the feature provides no useful information about the label. Larger MI values indicate that the feature carries more information about the label, making it more valuable for classification tasks. Features are ranked by $I(X_j; Y)$ and the top-K are selected.

The cumulative fraction policy (cumfrac) in Eq. (2) selects the smallest number of top features K such that their combined MI accounts for at least 90% of the total MI across all features. In practice, this means we first sort features by how much information they share with the label. Then we keep adding features, one by one, until the fraction of total information they explain reaches the threshold $\tau = 0.90$. This ensures that most of the useful information is kept while removing redundant or less important features [38].

$$K = \min \left\{ k : \frac{\sum_{i=1}^{k} I(X_{(i)}; Y)}{\sum_{j=1}^{n} I(X_{(j)}; Y)} \geq \tau \right\}, \quad \tau = 0.90 \tag{2}$$

where $X_{(i)}$ are features sorted by decreasing MI score.

3.2.3 Base and Reduced Models

The framework employs two random forest classifiers, which are a base model and a reduced model. The base model uses the entire feature set with dimensionality d, whereas the reduced model is restricted to only the top-K most informative features selected using MI. This allows the reduced model to achieve faster predictions with lower computational cost, while still maintaining competitive accuracy. The mathematical representations of the two models are given below:

The base model uses all d features is given in Eq. (3):

$$\hat{y}_{\text{base}} = f_{\text{RF}}(X), \quad X \in \mathbb{R}^d \tag{3}$$

While the reduced model uses the top-K features is represented in Eq. (4):

$$\hat{y}_{\text{reduced}} = f_{\text{RF}}(X_{1:K}), \quad K \ll d \tag{4}$$

Fig. 4 demonstrates the workflow of the base and reduced models. The base model directly uses all available input features, training a random forest followed by probability calibration without applying feature selection.

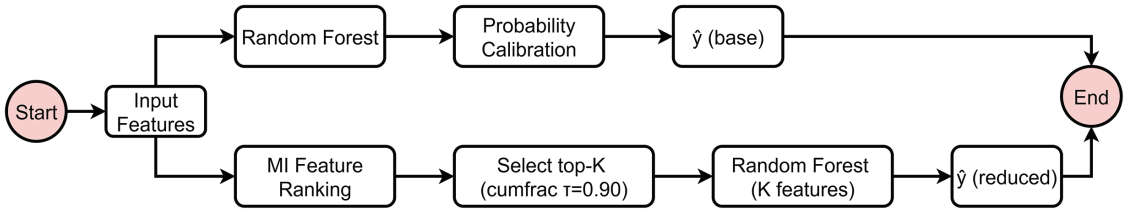


Figure 4: Workflow of base and reduced models.

In contrast, the reduced model applies MI feature ranking to the input features, retaining the top-K subset based on a cumulative fraction threshold of $\tau = 0.90$, which ensures that at least 90% of the overall information content is preserved. A random forest classifier is then trained on this reduced feature set. This dual setup, one model using all features and another using only the most informative subset, allows a clear comparison of their performance. This shows how the MI method helps remove redundancy, reduce dimensionality, and minimize noise.

3.2.4 ELECTRA Embeddings and Teacher Probe

Although IoT features are numeric, transforming each feature vector into a tokenized text sequence allows ELECTRA to model dependencies between features that traditional tabular methods treat independently. The transformer captures contextual relationships, feature combinations, and non-linear interactions that often characterize malicious IoT traffic. Using a lightweight logistic regression probe on top of these embeddings provides an efficient refinement step when the base model is uncertain, enabling richer representation learning with minimal additional cost. As shown in Eq. (5), each feature vector X_i is first converted into a text string and then passed through the ELECTRA model, which produces a dense embedding $z_i \in \mathbb{R}^h$. These embeddings capture richer semantic and structural patterns from the input features, which effectively maps raw numerical data into a representation space where complex relationships are easier to learn.

Features are transformed to text and embedded with ELECTRA:

$$z_i = \text{ELECTRA}\big(\text{Tokenizer}\big(\text{features}(X_i)\big)\big), \quad z_i \in \mathbb{R}^h \tag{5}$$

After the embeddings are created the Teacher Probe is implemented as a logistic regression classifier as represented in (6). The probe is trained on embeddings derived from the training set, making it computationally efficient while still leveraging the representational power of ELECTRA. At test time, when the base random forest model is uncertain, the Teacher Probe predicts labels using the test embeddings. This combination allows the framework to integrate deep language model representations into a classical ML pipeline without requiring fine-tuning of the large transformer itself.

A teacher probe (logistic regression) is trained on embeddings:

$$\hat{y}_{\text{probe}} = f_{\text{LR}}(z_i) \tag{6}$$

3.2.5 Entropy and Adaptive Thresholding

Entropy is used to to measure the uncertainty of a model or dataset [25]. Given a probability distribution $p = (p_1, p_2, \ldots, p_i)$ over i classes, the entropy as shown in Eq. (7), $H(p)$ calculates how confused the model is. If the distribution is sharp for example, one class has probability close to 1, the entropy is low, indicating high confidence. While, if the probabilities are not sharp, the entropy is higher, meaning the model is uncertain.

$$H(p) = -\sum_{i=1}^{n} p_i \log(p_i), \tag{7}$$

where p_i is the predicted probability for class i.

To decide when to trust the base model and when to move to a more expensive option (such as the Teacher Probe), an adaptive threshold T is used as in (8). This threshold is computed using the z-score of the entropy values across a batch. Specifically, the dynamic threshold is set to the mean entropy μ_H plus a multiple λ of the standard deviation σ_H.

$$\mathrm{T} = \mu_{H(p)} + \lambda \cdot \sigma_{H(p)}, \tag{8}$$

where $\mu_{H(p)}$ and $\sigma_{H(p)}$ are the mean and standard deviation of entropy values, and λ is a scaling factor.

3.2.6 Action Definitions

The agent has four possible actions, each representing a different trade-off between accuracy and computational cost.

Reduced Model: The first action is using random forest that is trained only on the top-K features selected using the MI cumfrac policy. By focusing on the most informative features, the reduced model is faster and simpler, with no added cost. However, it may sacrifice some accuracy compared to the full model. Reduced Features action can be represented as in Eq. (9)

$$\hat{y} = f_{\text{red}}(X_{1:K}) \tag{9}$$

Base Model: This is the second action as defined in (10), which uses the calibrated random forest model trained on all d features. It provides strong predictive performance since no information is discarded. Importantly, this option is considered cost-free, making it a reliable baseline.

$$\hat{y} = f_{\text{base}}(X) \tag{10}$$

LLM Refinement: In this third action, the Base Model initially performs the prediction, and its confidence is quantified using the entropy of the predicted probability distribution, denoted as $H(p)$. If the entropy is below the adaptive threshold T (indicating low uncertainty), the prediction from the Base Model is accepted directly. Conversely, if the entropy exceeds the threshold (indicating high uncertainty), the system activates the Teacher Probe, a logistic regression classifier trained on ELECTRA embeddings.

This mechanism refines the decision for ambiguous samples by leveraging deep contextual representations. The decision rule for this action is formally expressed in Eq. (11):

$$\hat{y}_i = \begin{cases} f_{\text{base}}(X_i), & \text{if } H(p_i) < T, \\ f_{\text{probe}}(z_i), & \text{if } H(p_i) \geq T, \end{cases} \tag{11}$$

Escalate to Analyst: The fourth action is used when predictions are highly uncertain. The top K samples with the highest entropy values are forwarded to an analyst. In practice, this means that the system does not assign a predicted label for these cases and instead marks them as unresolved pending manual verification. As shown in Eq. (12), if the sample i is not in the Top-K uncertain samples ($i \notin \text{TopK}(H(p))$), the prediction is made by the base model $f_{\text{base}}(X_i)$. However, if sample i is in the Top-K uncertain samples ($i \in \text{TopK}(H(p))$), the output is assigned an UNRESOLVED label to reflect escalation to the analyst.

$$\hat{y}_i = \begin{cases} f_{\text{base}}(X_i), & \text{if } i \notin \text{TopK}(H(p)), \\ \text{UNRESOLVED}, & \text{if } i \in \text{TopK}(H(p)). \end{cases} \tag{12}$$

3.2.7 Adaptive Decision Agent Workflow

Fig. 5 illustrates the workflow of the adaptive decision approach, which controls the dynamic action selection process during inference. The input batch, which comes from the test set, is passed to MI model to perform feature selection using a cumulative fraction policy to identify the most informative subset of features. This step ensures that redundant or less relevant features are excluded.

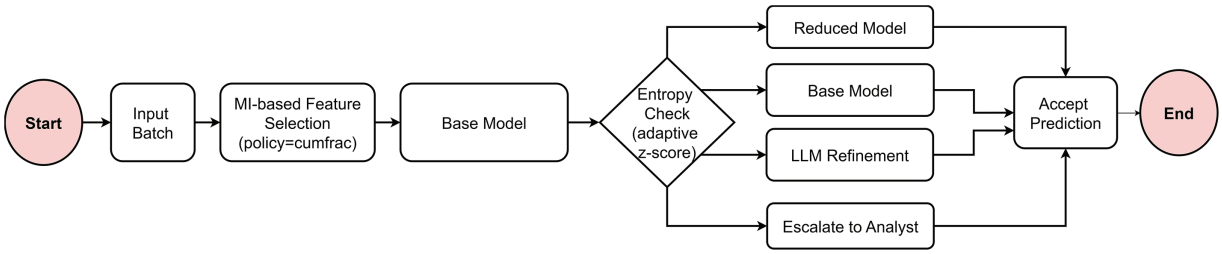


Figure 5: Adaptive decision controller.

The selected features are then submitted to the Base Model to produce class predictions along with confidence scores. These scores are checked using an adaptive entropy method to decide how reliable each prediction is.

As the figure shows, the model dynamically selects one of four decision pathways based on the evaluated entropy level. For instance, in case of low-entropy, the Reduced Model path is used after selecting the top-K most informative features. It is worth noting that this option is computationally efficient and suitable when predictions are already confident. The system switches to the Base Model when the entropy level increased to operate on the complete feature set. This enhances reliability in uncertain cases. The LLM Refinement process is activated by the framework when the entropy rises. This uses deep contextual representations together with logistic regression and ELECTRA embeddings to enhance predictions.

Finally, the agent initiates the Escalate to Analyst pathway when the entropy stays over a certain threshold. For manual verification, it sends the top-K highly uncertain instances. The process concludes at the Accept Prediction module where the result of the action selected is used to make the final decision of the system. This adaptive control improves accuracy, interpretability and computational efficiency by intelligent allocation of computational resources based on the uncertainty of each input sample.

Overall, these four steps allow the model to effectively balance computational costs toward predictive performance. In order to maintain efficiency without reducing accuracy, the framework is encouraged by the cost structure to keep the more costly choices (Actions 3 and 4) to only highly uncertain samples.

3.2.8 Reward Function

The reward function is designed to balance prediction accuracy with the computational or operational penalty of each action. The reward calculates the difference between the performance score of the model and the penalty of the chosen action as defined in Eq. (13). We use accuracy as the scoring function that measures the proportion of correctly classified samples. Actions that achieve high accuracy with minimal penalty will receive higher rewards. However, actions that are more expensive or less accurate will produce lower rewards. For instance, there is no penalty associated with the usage of the Reduced or Base models, therefore the accuracy is equivalent to the reward. By contrast, utilizing the Teacher Probe or forwarding the Top-K uncertain samples to the analyst incurs additional penalties, which reduces the reward even when accuracy is high. This design ensures that the system learns to prefer strategies that are not only correct but also efficient, reflecting a realistic trade-off between accuracy and resource usage.

$$r = \text{Score}(y, \hat{y}) - \text{Penalty}(a) \tag{13}$$

3.2.9 LinUCB Algorithm

Each action in the LinUCB algorithm keeps its own parameters, which are represented as a vector b_a and a matrix A_a. The matrix $A_a \in \mathbb{R}^{d\times d}$ captures how much information has been gathered about the action based on past feature contexts, while $b_a \in \mathbb{R}^d$ accumulates the rewards observed [39]. Using these values, the algorithm estimates the parameter vector $\theta_a = A_a^{-1} b_a$, which can be interpreted as the model's best guess about the relationship between the context and the expected reward for action a. When deciding which action to take, LinUCB computes an upper confidence bound (UCB) score for each action as written in Eq. (14):

$$p_a(x) = \theta_a^\top x + \alpha\sqrt{x^\top A_a^{-1} x}. \tag{14}$$

The first term, $\theta_a^\top x$, represents the predicted reward for the action given the current context (features) x. The second term adds an exploration bonus, scaled by the parameter α. This bonus is larger when the algorithm has less certainty about the action (i.e., when A_a^{-1} is large), encouraging exploration of under-tested actions.

The action with the highest score is then selected as shown in Eq. (15):

$$a_t = \arg\max_a p_a(x_t), \tag{15}$$

where $p_a(x_t)$ represents the estimated upper confidence bound for action a given context x_t. This means the agent selects the action that maximizes the combination of the predicted reward and the uncertainty bonus.

Once the reward r_t is observed, the algorithm updates its internal parameters for the chosen action as in Eq. (16):

$$A_a \leftarrow A_a + x_t x_t^\top, \quad b_a \leftarrow b_a + r_t x_t. \tag{16}$$

These updates allow LinUCB to continuously refine its confidence estimates, gradually learning which actions yield higher rewards under different contexts.

3.2.10 Datasets

In this research, we evaluated our proposed adaptive intrusion detection framework on three benchmark datasets including RT-IoT-2022 [40], CIC-IoT-2023 [41] and CIC-IoMT-2024 [42]. We chose these datasets because they represent modern network behavior, diverse device types and unseen attacks in IoT and IoMT environments.

The first dataset used is RT-IoT-2022 and contains attack and normal traffic from a real-time IoT setup. It was designed for intrusion and anomaly detection research and provides 83 flow-based features such as packet statistics, payload details, timing and TCP flags [40]. The dataset has nine attack types that are SYN flooding, Slowloris distributed denial of service (DDoS), ARP poisoning and Nmap scans as well as three types of benign traffic. This variety makes this dataset a strong test for classification methods, particularly given the limited resources common in IoT systems.

The second dataset is CIC-IoT-2023 that also supports research on intrusion and anomaly detection and was developed using a network of 105 real IoT devices [41]. It has 33 attack types from compromised devices that are grouped into seven categories namely DDoS, DoS, reconnaissance, web attacks, brute force, spoofing and Mirai botnet variants. The dataset provides 46 features describing network flows, packet statistics and timing. This range enables us to evaluate our detection methods on a variety of new IoT attacks and takes various devices and protocols into account.

The third dataset is CIC-IoMT-2024 that was collected from a simulated healthcare network with typical medical IoT devices and protocols [42]. The dataset includes both benign and malicious traffic such as resource loss attacks, illegal access, and probing. It provides 44 flow-based features on protocol use, packet statistics and timing. This setup lets us test our detection methods in healthcare where device variety and the need for constant service add extra challenges.

Overall, the use of the aforementioned datasets to evaluate our intrusion detection framework helps to demonstrate its effectiveness and adaptability to diverse IoT and IoMT environments. The variety of attack types, device behaviors and protocol characteristics in RT-IoT-2022, CIC-IoT-2023, and CIC-IoMT-2024 enables a thorough evaluation of the classification performance of our model under realistic conditions. Table 2 summarizes the key characteristics of the datasets used in this study, highlighting their differences in environment, feature dimensionality, attack diversity, and application domain.

4 Results and Discussion

In this section, we present and discuss the experimental results of our proposed adaptive intrusion detection framework. We used three benchmark IoT datasets to evaluate the proposed framework that are RT-IoT 2022, CIC-IoT 2023 and CIC-IoMT 2024 datasets. First, we show the MI distribution of each dataset to highlight the most informative features selected for analysis. Next, we examine the usage proportions of the four decision pathways namely Reduced Model, Base Model, LLM Refinement and Escalate to Analyst to understand the way that the adaptive policy allocates computational effort. After that, we evaluate the per-action performance of the proposed framework by computing the well known evaluation metrics and calculating the corresponding reward for each action. Finally, we analyze the trade-off between accuracy and computational cost in the four decision actions of the proposed framework.

4.1 Evaluation on RT-IoT-2022 Dataset

4.1.1 Mutual Information (MI) Result on RT-IoT-2022 Dataset

MI was employed to estimate the relevance of each feature to the target class and to guide dimensionality reduction. Fig. 6 shows the ranked MI scores for all features in the RT-IoT-2022 dataset. Each bar represents a

feature's MI value, where higher scores indicate stronger dependence on the class label. Using the cumulative-fraction policy with a threshold of $\tau = 0.90$, the cutoff indicated by the red dashed line retains K = 58 out of d = 82 features. These 58 features account for at least 90% of the total MI, while the remaining variables contribute only marginal information and are removed. This reduction eliminates redundant or low-impact features and lowers computational cost by focusing the model on the most informative subset. To justify $\tau = 0.90$, we evaluated thresholds $\tau = 0.70, 0.80, 0.90, 0.95$, and 0.99 as presented in Table 3. The MI values drop quickly, so a small number of features carry most of the useful information. At $\tau = 0.70$ and $\tau = 0.80$, only 41 and 49 features are needed to cover most of the MI. The reference threshold $\tau = 0.90$ selects 58 features, providing a stable balance between information coverage and dimensionality. Increasing the threshold beyond this point brings only minor benefit. For example, $\tau = 0.99$ increases the count to 72 features, but the extra features add almost no meaningful information.

Table 2: Summary of datasets used in this study.

Dataset	Environment	# Features	Attack Diversity	Traffic Representation	Application Domain
RT-IoT-2022	Real-time IoT network	83	Multiple DoS, scanning, poisoning, and benign traffic types	Flow-based	General IoT security
CIC-IoT-2023	Large-scale IoT testbed	46	33 attacks grouped into 7 categories (DDoS, DoS, reconnaissance, web, brute force, spoofing, Mirai)	Flow-based	Heterogeneous IoT networks
CIC-IoMT-2024	Simulated healthcare network	44	Multiple resource abuse, illegal access, and probing attacks	Flow-based	Medical IoT (IoMT)

Jaccard similarity analysis also supports this choice. When using $\tau = 0.70$ or $\tau = 0.80$, the selected features are simply smaller versions of the same important set obtained at $\tau = 0.90$. When increasing the threshold to $\tau = 0.95$ or $\tau = 0.99$, the extra features added are mostly low-importance variables that do not change the core set. In short, $\tau = 0.90$ preserves the meaningful features while avoiding unnecessary, low-impact variables.

4.1.2 Actions Usage and Performance Analysis on RT-IoT-2022 Dataset

The percentage of each action is calculated to determine action usage in the RT-IoT-2022 dataset. This helps identify which action dominates when the adaptive framework is used. As shown in Fig. 7a, the LinUCB controller can dynamically select among four possible inference actions based on prediction confidence and computational cost. There are four actions as mentioned in the methodology section named Reduced Model, Base Model, LLM Refinement, and Escalate to Analyst. The overall results demonstrate that the Reduced Model and Base Model together recorded 50% of all decisions. The LLM Refinement was used in 27.55% of the total actions. The Escalate to Analyst accounted for 22.45%, which is desirable since relying less on this costly action improves overall efficiency.

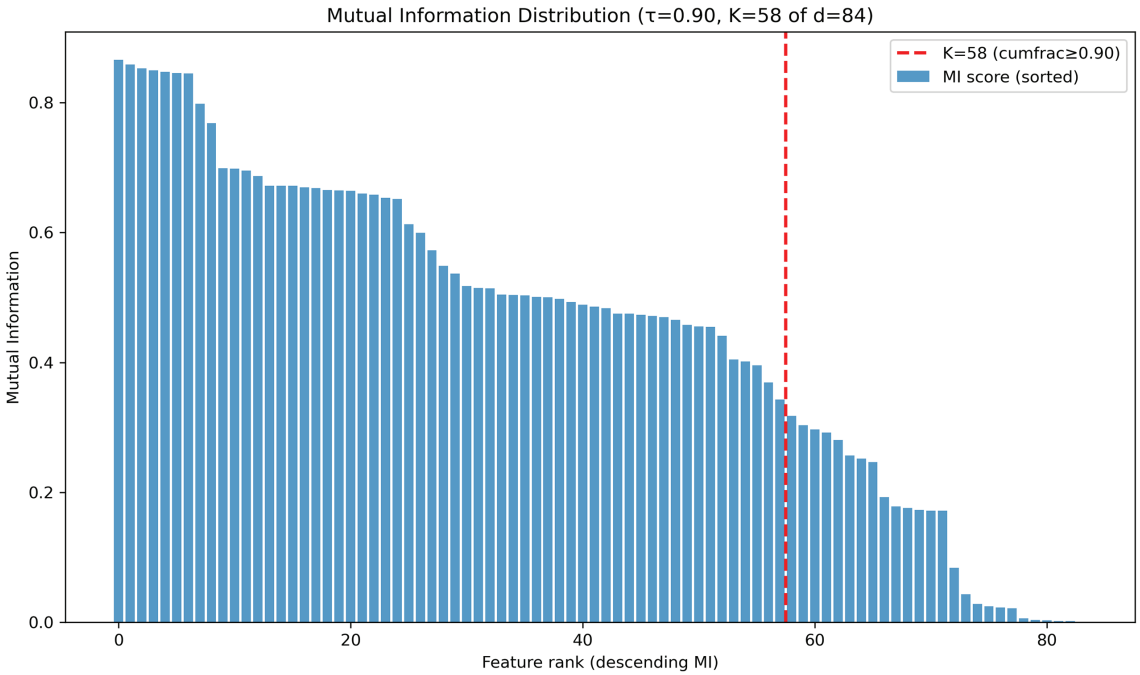
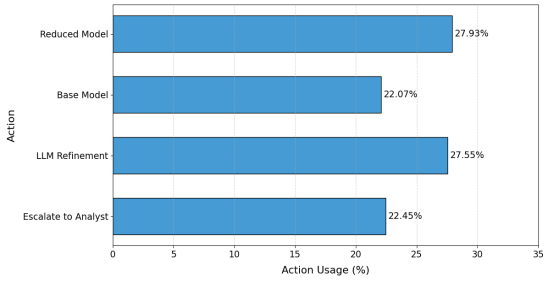


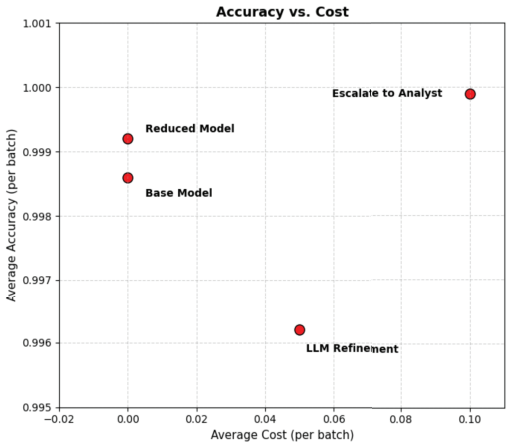
Figure 6: MI distribution of RT-IoT-2022 dataset.

Table 3: Effect of different cumulative MI thresholds on the number of selected features for the RT-IoT-2022 dataset.

Threshold τ	Selected Features (K)	Percentage of Features	Jaccard vs. τ = 0.90
0.70	41	48.81%	0.707
0.80	49	58.33%	0.845
0.90	58	69.05%	1.000
0.95	64	76.19%	0.906
0.99	72	85.71%	0.806



(a) Action usage for the adaptive IDS on the RT-IoT-2022 Dataset.



(b) Accuracy vs. Penalty of the RT-IoT-2022 Dataset.

Figure 7: Adaptive IDS behavior on the RT-IoT-2022 dataset.

Therefore, the proposed framework reduces computation, as half of the actions involve no extra processing cost. The results also show that the Reduced Model being the most frequently chosen 27.93%, which reflects the system's preference for computational efficiency inference when confidence is high.

The per-action performance on the RT-IoT-2022 dataset using 5-fold average is shown in Table 4. The Reduced Model and Base Model achieve the best overall balance, with high accuracy at 0.9992 and 0.9986, zero penalty, and the lowest inference time per sample at 0.105 and 0.107 ms. These results confirm that lightweight actions handle most samples efficiently without compromising accuracy. For uncertain cases, the framework triggers LLM Refinement. ELECTRA reaches 0.9962 accuracy and 0.9981 F1-score, while DistilBERT achieves 0.9973 accuracy and 0.9986 F1-score. The two models deliver comparable refinement performance, with only minor differences in accuracy and F1, and both incur the same five percent penalty and inference time of about 0.238 ms per sample. Escalate to Analyst achieves the highest accuracy at 0.9999 and F1-score at 0.9948 but carries the highest penalty (10%) and therefore the lowest reward among the correct actions. Overall, the results show that the framework prioritizes fast, zero-penalty actions and uses costly steps only when uncertainty is high.

Table 4: Per-action performance on the RT-IoT-2022 dataset (5-fold average).

Action	Accuracy	F1-Score	Penalty	Reward	Inference Time (ms)
Reduced Model	0.9992	0.9995	0%	0.9992	0.105
Base Model	0.9986	0.9990	0%	0.9986	0.107
LLM Refinement (ELECTRA)	0.9962	0.9981	5%	0.9462	0.238
LLM Refinement (DistilBERT)	0.9973	0.9986	5%	0.9473	0.238
Escalate to Analyst	0.9999	1.0000	10%	0.8999	0.243

It is worth mentioning that each experiment uses 5-fold cross-validation. We used 5-fold cross-validation to balance statistical robustness and computational efficiency, as higher fold counts (e.g., 10-fold) substantially increase training and evaluation cost and are commonly reported to provide only marginal performance gains when sufficient data is available.

In each fold, the dataset is first split into 80% training and 20% testing. The training portion is then further divided into 80% training and 20% validation, resulting in final proportions of 64% training, 16% validation, and 20% testing. The validation set is used for model calibration, feature selection, and adaptive decision tuning, while the test set remains completely unseen until final evaluation.

Different split ratios mainly affect the bias–variance trade-off. Increasing the training portion may slightly improve fitting but reduces the reliability of validation and test estimates, whereas smaller training sets increase variance, particularly in imbalanced IoT traffic. The chosen configuration yields stable and consistent performance across folds.

We also analyzed the trade-off between accuracy and the action penalty across the four decision actions of the proposed framework on the RT-IoT-2022 dataset. The x-axis represents the assigned penalty, while the y-axis shows the average accuracy, as illustrated in Fig. 7b. The optimal region appears on the far left, where high accuracy aligns with minimal penalty. Conversely, the far-right region reflects actions that achieve high accuracy but incur higher penalties.The Reduced Model and Base Model appear in the far-left region and achieve near-perfect accuracy with zero penalty, confirming their efficiency for standard predictions. The LLM Refinement action introduces a moderate penalty with a slight decrease in accuracy, and is triggered only for uncertain samples. The Escalate to Analyst action, which offers the highest accuracy, carries the highest penalty and appears in the upper-right corner, reflecting the cost of manual verification. Overall, the

figure shows that the framework balances accuracy and efficiency. Zero-penalty actions handle most inputs reliably, while higher-penalty actions are used only when uncertainty is high.

4.2 Evaluation on CIC IoT 2023 Dataset

4.2.1 Mutual Information (MI) Result on CIC IoT-2023 Dataset

The MI distribution for the CIC-IoT-2023 dataset, which contains $d = 46$ features. At this point, $K = 23$ features are retained, while the remaining 23 features, which contribute only marginally to the predictive signal, are excluded. This selection balances dimensionality reduction with information retention, ensuring that the model leverages the most informative subset of features while discarding less relevant ones to reduce complexity and noise.

The MI distribution of the CICIoT-2023 dataset reveals that among the total $d = 46$ extracted features, the informative content is highly concentrated within the top-ranked subset. As presented in Fig. 8, the features are arranged in descending order of their MI scores, showing a rapid decline after the leading features. This indicates that only a limited number of them significantly contribute to class discrimination. Using the cumulative fraction policy with a threshold, a total of $K = 23$ features are retained, capturing 90% of the total MI, while the remaining features are discarded due to their marginal relevance.

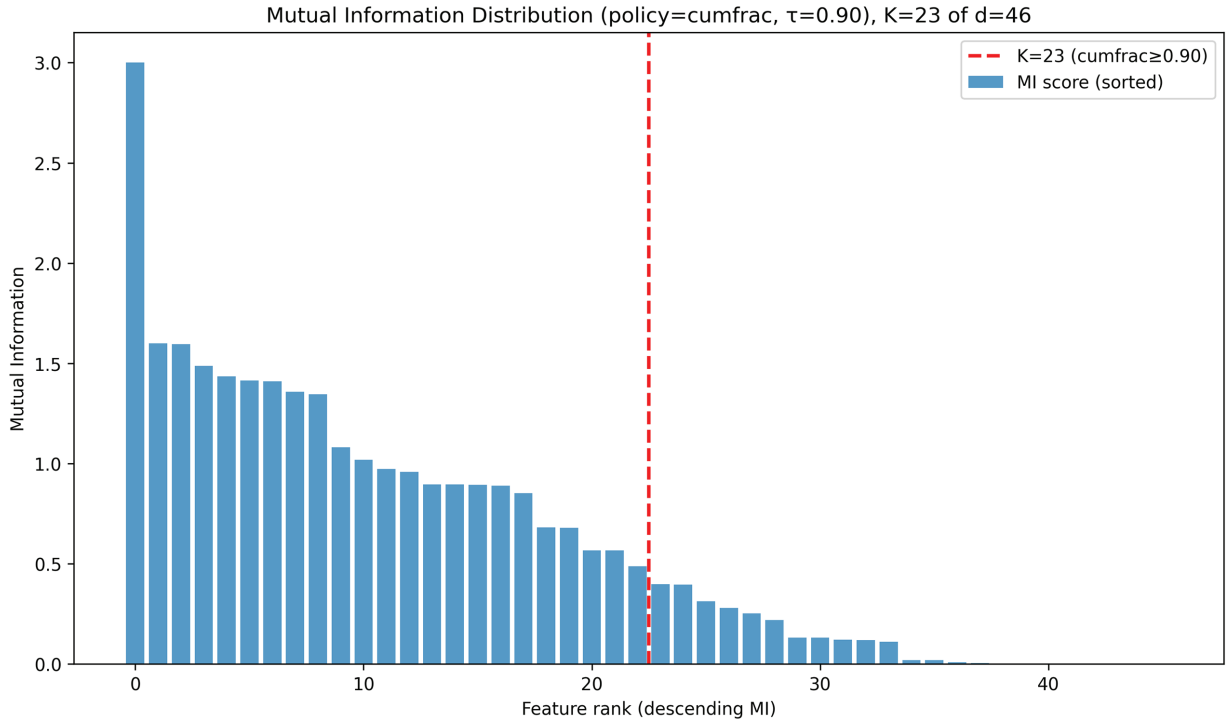
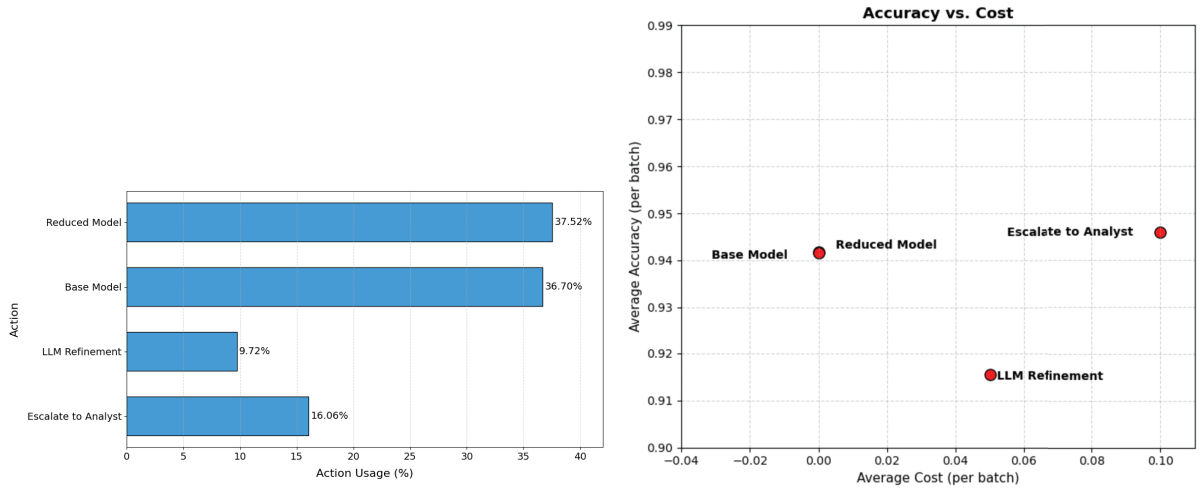


Figure 8: MI distribution of CIC IoT 2023 dataset.

4.2.2 Actions Usage and Performance Analysis on CIC IoT 2023 Dataset

The distribution of action usage within the proposed framework on the CIC-IoT-2023 dataset is analyzed in this research. Fig. 9a illustrates that the Reduced Model and Base Model account for around 37.52% and 36.70% of all forecasts, respectively and therefore lead the decision-making process. This highlights the preference of the framework for low-cost inference pathways, which confirms that most network traffic can be accurately classified using lightweight or full-feature models without the need for additional computation.

The LLM Refinement action, which involves deeper semantic processing through ELECTRA embeddings and logistic regression, is used for only 9.72% of samples. This indicates that uncertainty levels in the baseline predictions are relatively low and the refinement mechanism is triggered selectively for uncertain cases. Finally, the Escalate to Analyst action was used for 16.06% of the cases, which represents that a small portion of samples had the highest prediction entropy. Overall, the adaptive framework successfully used the Reduced and Base Models for 74.22% of the cases and only about 25.78% were handled by LLM Refinement and Escalate to Analyst. This emphasizes that the model primarily relies on low-cost actions.



(a) Action usage for the adaptive IDS on the CIC-IoT-2023 Dataset.

(b) Accuracy vs. Penalty of the CIC-IoT-2023 Dataset.

Figure 9: Adaptive IDS behavior on the CIC-IoT-2023 dataset.

On the CIC-IoT-2023 dataset,the Reduced Model and Base Model deliver almost identical performance with accuracies of 0.9417 and 0.9416 and very low inference time in the range of 0.1020 to 0.1110 ms as reported in Table 5. This indicates that the framework maintains strong predictive capability under both lightweight and full-feature inference without any penalty. The LLM Refinement actions powered by ELECTRA and DistilBERT produce closely matched results with accuracies near 0.916 to 0.919 and weighted F1-scores near 0.951 to 0.953. Both actions incur a five percent penalty, leading to rewards in the range of 0.865 to 0.869, and introduce a slightly higher latency of about 0.24 ms. The Escalate to Analyst action achieves the highest accuracy of 0.9459 and a weighted F1-score of 0.9698, though its ten percent penalty reduces the reward to 0.8459. Overall, the controller maintains a stable detection performance while using higher-cost actions only when uncertainty makes them necessary.

Table 5: Per-action performance on the CIC-IoT-2023 Dataset (5-fold average).

Action	Accuracy	F1-Score	Penalty	Reward	Inference Time (ms)
Reduced Model	0.9417	0.9666	0%	0.9417	0.102
Base Model	0.9416	0.9663	0%	0.9416	0.111
LLM Refinement (ELECTRA)	0.9155	0.9509	5%	0.8655	0.245
LLM Refinement (DistilBERT)	0.9189	0.9532	5%	0.8689	0.239
Escalate to Analyst	0.9459	0.9698	10%	0.8459	–

Fig. 9b shows the trade-off between accuracy and action penalty on the CIC-IoT-2023 dataset. The Reduced Model and Base Model appear in the top-left region, reaching about 0.94 accuracy with zero penalty, confirming their efficiency for most samples. LLM Refinement lies in the middle with a 5% penalty and lower accuracy (around 0.92), reflecting its selective use for uncertain cases. Escalate to Analyst action achieves accuracy around 0.945 but at the highest penalty (10%), indicating that escalation is reserved for the most difficult, high-risk samples.

4.3 Evaluation on CIC IoMT 2024

4.3.1 Mutual Information (MI) Result on CIC IoMT 2024 Dataset

Fig. 10 presents the MI distribution for the CIC-IoMT-2024 dataset, which has $d = 45$ features. The blue bars correspond to MI scores and the solid line represents cumulative MI. The use of the $\tau = 0.90$ cutoff, $K = 25$ features are selected, as shown by the vertical dashed red line. The retained subset accounts for at least 90% of the overall information, while 20 less-informative features are removed. This selection strategy reduces computational complexity while preserving the predictive richness of the dataset.

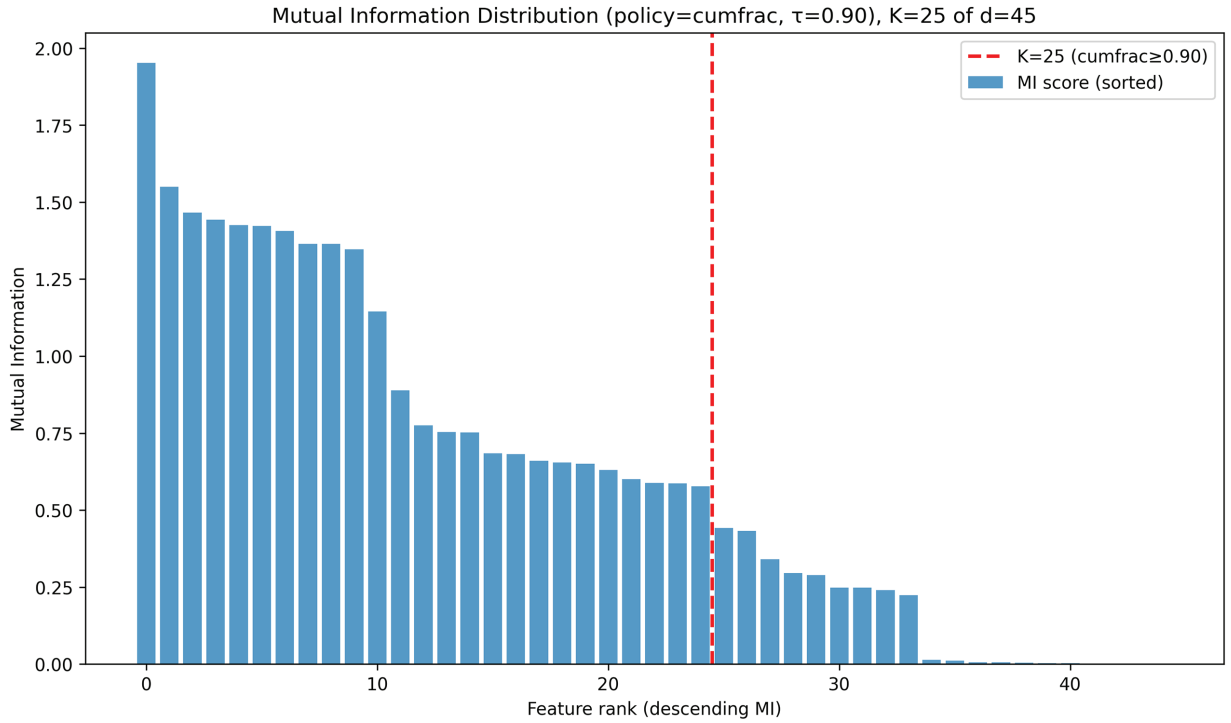
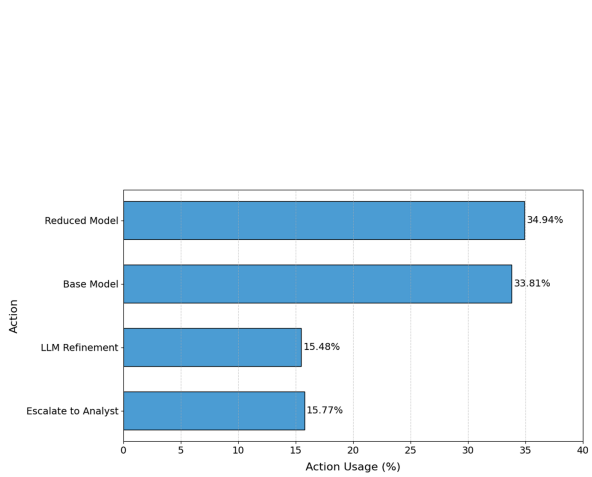


Figure 10: MI distribution of CIC IoMT 2024 dataset.

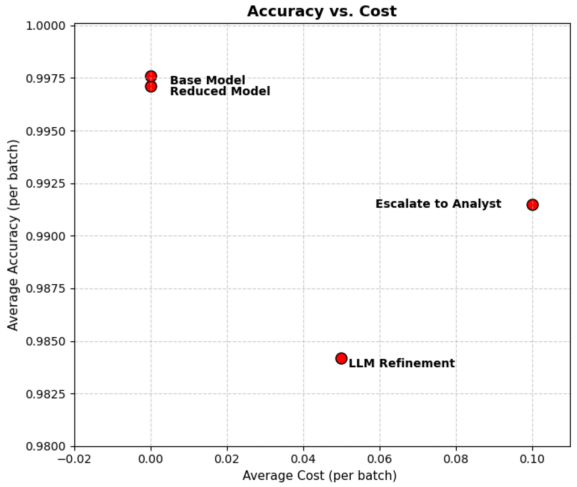
4.3.2 Actions Usage and Performance Analysis on CIC IoMT 2024 Dataset

The distribution of action usage for the proposed framework in the CIC-IoMT-2024 dataset is shown in Fig. 11a. The results show that the Reduced Model and Base Model together dominate the adaptive decisions as they accounted for 68.75% of all predictions. This reflects the framework's strong preference for computationally efficient because most samples are confidently classified without the need for additional processing. The LLM Refinement action was triggered for 15.48% of instances, which represents uncertain samples where the confidence of the primary models falls below the adaptive entropy threshold. Meanwhile, the Escalate to Analyst action was invoked for 15.77% of the inputs, which corresponds to the most complex

samples with the highest uncertainty levels. These cases are routed for expert-level or external analysis to ensure accuracy and reliability. The near-balanced distribution between low-cost and high-cost actions confirms that the adaptive policy dynamically optimizes the trade-off between efficiency and precision, which ensures that the system maintains high detection accuracy while minimizing unnecessary computation.



(a) Action usage for the adaptive IDS on the CIC-IoMT-2024 Dataset.



(b) Accuracy vs. Penalty of the CIC-IoMT-2024 Dataset.

Figure 11: Adaptive IDS behavior on the CIC-IoMT-2024 dataset.

The framework was also evaluated on CIC-IoMT-2024 dataset using 5-fold cross-validation, and the averaged per-action results as shown in Table 6. The Reduced Model and Base Model actions provide the strongest overall balance, reaching accuracies of 0.9971 and 0.9976 with weighted F1-scores of 0.9984 and 0.9987, zero penalty, and small latency of about 0.105 to 0.109 ms.

Table 6: Per-action performance on the CIC-IoMT-2024 dataset (5-fold average).

Action	Accuracy	F1-Score	Penalty	Reward	Inference Time (ms)
Reduced Model	0.9971	0.9984	0%	0.9971	0.105
Base Model	0.9976	0.9987	0%	0.9976	0.109
LLM Refinement (ELECTRA)	0.9842	0.9913	5%	0.9342	0.239
LLM Refinement (DistilBERT)	0.9835	0.9909	5%	0.9335	0.240
Escalate to Analyst	0.9915	0.9955	10%	0.8915	–

The LLM Refinement action achieves accuracies of 0.9842 with ELECTRA and 0.9835 with DistilBERT, with corresponding weighted F1-scores of 0.9909 and 0.9909. Both models deliver comparable accuracy, and the five percent penalty reduces the rewards to 0.9342 and 0.9335. Latency increases slightly due to the refinement step, reaching roughly 0.239 to 0.240 ms per sample. The Escalate to Analyst action attains 0.9915 accuracy and 0.8999 F1-score but carries the highest ten percent penalty.

The relationship between accuracy and action penalty on the CIC IoMT 2024 dataset is shown in Fig. 11b. The Reduced Model and Base Model actions appear at the zero penalty region with accuracies of 0.9971 and 0.9976. They deliver the strongest efficiency to accuracy value and remain the dominant choices for most traffic. The LLM Refinement action carries a five percent penalty and achieves an accuracy of about

0.984. This reduction is expected because the action is triggered only for high-uncertainty samples, which are naturally more difficult to classify. The Escalate to Analyst action appears at the ten percent penalty level and reaches an accuracy of about 0.991. While not the highest, it provides the most reliable option for the hardest cases because it represents human verification.

To further validate the findings of our research, we compare the classification accuracy and F1-score of the proposed framework with existing intrusion detection approaches, as summarized in Table 7. Unlike prior studies that report a single static model, the proposed framework is adaptive and operates through multiple inference paths. Depending on prediction uncertainty and computational constraints, the controller dynamically selects between lightweight (Reduced Model), full feature (Base Model), refinement, or escalation strategies. Therefore, Table 7 reports the performance of the proposed framework under its two primary zero-penalty inference paths (Reduced and Base Models), which represent the dominant operating modes during deployment.

Table 7: Comparison of the proposed adaptive IDS framework with existing intrusion detection approaches on IoT benchmark datasets.

Study	Dataset	Method	Accuracy	F1-Score	Computational Cost
Hafid et al. (2025)	CIC-IoMT-2024	XGBoost+Logistic Regression (Late Fusion)	97%	98%	Bandwidth reduced by 73%; latency < 200 ms
Houichi et al. (2025)	CIC-IoT-2023	ML-Based IDS (KNN, RF, AdaBoost, SVM)	99%	91%	—
Almohaimeed et al. (2024)	RT-IoT-2022	Feature Selection + MLP	96%	92%	Runtime reduced by 66.4%
Prasad et al. (2025)	CIC-IoT-2023	RegNet + FBNet + CSMCR Balancing	98%	98%	Training time reduced by 53%
Mahdi et al. (2025)	CIC-IoT-2023	Incremental Learning + Blockchain	99.53%	–	Encryption overhead (0.0001–0.0003 s); transaction time 2–3 s
Proposed Framework (ours)	RT-IoT-2022	Reduced Model	99.92%	99.95%	0.105 ms
	CIC-IoT-2023	Reduced Model	94.17%	96.66%	0.102 ms
	CIC-IoMT-2024	Reduced Model	99.74%	99.84%	0.105 ms
	RT-IoT-2022	Base Model	99.86%	99.90%	0.107 ms
	CIC-IoT-2023	Base Model	94.16%	96.63%	0.111 ms
	CIC-IoMT-2024	Base Model	99.76%	99.87%	0.109 ms

Hafid et al. [35] reported strong performance on CIC-IoMT-2024 using a late-fusion XGBoost and Logistic Regression model, achieving 97% accuracy and 98% F1-score. While both their approach and the proposed framework incorporate adaptivity, their interpretability relies on SHAP values and their evaluation follows a static cost-performance analysis. Their method reduces bandwidth usage by 73% with detection latency below 200 ms, whereas our framework achieves substantially lower inference time in the sub-millisecond range.

Houichi et al. [32] reported 99% accuracy and a lower F1-score of 91% on CIC-IoT-2023 using traditional ML classifiers. However, their evaluation was limited to a single dataset and lacked adaptive decision control, limiting generalizability.

Prasad et al. [27] achieved 98% accuracy and a 98% F1-score on CIC-IoT-2023 using a DL model with cosine-similarity–based balancing. Their balancing method mitigated the inherent class imbalance of the dataset. In addition, their deep-learning pipeline followed a static inference process with no adaptive or cost-aware control. In contrast, the proposed adaptive framework maintains consistently strong performance across three benchmark datasets (RT-IoT-2022, CIC-IoT-2023, CIC-IoMT-2024) and introduces a LinUCB-based decision layer that dynamically balances accuracy and penalty under varying network conditions. They reported a 53% reduction in training time, whereas our framework delivers fast inference with an average cost near 0.1 ms per sample.

Almohaimeed and Albalwy [24] applied multiple feature selection methods to RT-IoT-2022 followed by an MLP classifier, achieving 96% accuracy and a 92% F1-score. While effective at reducing dimensionality, their approach does not incorporate adaptive or cost-aware decision logic. Our framework surpasses these results, achieving up to 99.92% accuracy and 99.95% F1-score on RT-IoT-2022 using the Reduced Model. Their method reduced runtime by 66.4%, whereas our framework additionally achieves sub-millisecond inference with higher predictive performance.

Finally, Mahdi et al. [31] used the CIC-IoT-2023 dataset, which is inherently imbalanced, with attack traffic dominating normal flows. The authors did not apply any explicit data-balancing method, instead relying on the incremental learning process to adapt to changing distributions. Consequently, the reported 99.53% accuracy may be influenced by class imbalance, as no F1-score or weighted metrics were provided. Their system introduces blockchain-related overhead (0.0001 s–0.0003 s per operation and 2–3 s per transaction), whereas the proposed IDS maintains inference latency near 0.1 ms.

5 Conclusions

In conclusion, we proposed an adaptive intrusion detection framework designed for IoT environments. This asymmetrical nature of IoT systems involves limited computational resources, high feature dimensionality, and prediction uncertainty. The proposed framework has the ability to intelligently balance detection accuracy and computational cost. This is accomplished by applying MI as a feature selection model, which effectively reduces the dimensionality of the data and keeps only the most informative features while eliminating redundancy. The adaptive decision dynamically selects between multiple inference paths, which ensures that additional computation is used only when uncertainty is high. Performance evaluation on the RT-IoT-2022, CIC-IoT-2023, and CIC-IoMT-2024 datasets indicates that the framework achieves F1-scores of 99.92%, 96.66%, and 99.84%, respectively, with an average inference time of approximately 0.105 ms per sample. A limitation of this work is that it focuses on binary intrusion classification. This choice was made to clearly study the balance between detection accuracy and computational cost. In future work, the framework will be extended to multiclass intrusion detection to support more detailed attack classification. In addition, future studies will evaluate the framework in real IoT environments and improve the adaptive controller by considering additional factors such as energy consumption, network latency, and device diversity.

Acknowledgement: Not applicable.

Funding Statement: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia under grant no. (IPP: 753-611-2025). The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Author Contributions: Conceptualization, Abdulaziz A. Alsulami, Badraddin Alturki and Ahmad J. Tayeb; methodology, Abdulaziz A. Alsulami and Rayan A. Alsemmeari; software, Abdulaziz A. Alsulami; validation, Abdulaziz A. Alsulami, Rayan A. Alsemmeari and Ahmad J. Tayeb; formal analysis, Ahmad J. Tayeb; investigation, Badraddin Alturki; resources, Abdulaziz A. Alsulami; data curation, Abdulaziz A. Alsulami; writing—original draft preparation, Abdulaziz A. Alsulami, Badraddin Alturki and Ahmad J. Tayeb; writing—review and editing, Rayan A. Alsemmeari, Raed Alsini and Badraddin Alturki; visualization, Badraddin Alturki; supervision, Abdulaziz A. Alsulami; project administration, Badraddin Alturki, Ahmad J. Tayeb and Raed Alsini; funding acquisition, Abdulaziz A. Alsulami. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The benchmark datasets used in this study (RT-IoT2022, CIC-IoT2023, and CIC-IoMT2024) are publicly available from their respective sources.

Ethics Approval: This study does not involve human participants, animal subjects, or the use of personal or sensitive data. Therefore, ethical approval was not required.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sadeghi-Niaraki A. Internet of Thing (IoT) review of review: bibliometric overview since its foundation. Future Gener Comput Syst. 2023;143(5):361–77. doi:10.1016/j.future.2023.01.016.
2. Ahmetoglu S, Che Cob Z, Ali N. A systematic review of Internet of Things adoption in organizations: taxonomy, benefits, challenges and critical factors. Appl Sci. 2022;12(9):4117. doi:10.3390/app12094117.
3. Jouhari M, Saeed N, Alouini MS, Amhoud EM. A survey on scalable LoRaWAN for massive IoT: recent advances, potentials, and challenges. IEEE Commun Surv Tutor. 2023;25(3):1841–76. doi:10.1109/comst.2023.3274934.
4. Vailshery LS. Number of IoT connections worldwide 2022–2034. 2025 [cited 2025 Oct 10]. Available from: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.
5. Vailshery LS. IoT global annual revenue 2020–2034. 2025 [cited 2025 Oct 10]. Available from: https://www.statista.com/statistics/1194709/iot-revenue-worldwide/.
6. Li J, Othman MS, Chen H, Yusuf LM. Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning. J Big Data. 2024;11(1):36. doi:10.1186/s40537-024-00892-y.
7. Merlino V, Allegra D. Energy-based approach for attack detection in IoT devices: a survey. Internet Things. 2024;27(4):101306. doi:10.1016/j.iot.2024.101306.
8. Altulaihan E, Almaiah MA, Aljughaiman A. Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. Sensors. 2024;24(2):713. doi:10.3390/s24020713.
9. Hossain MA. Deep Q-learning intrusion detection system (DQ-IDS): a novel reinforcement learning approach for adaptive and self-learning cybersecurity. ICT Express. 2025;11(5):875–80. doi:10.1016/j.icte.2025.05.007.
10. Hossain MA, Islam MS. A novel feature selection-driven ensemble learning approach for accurate botnet attack detection. Alex Eng J. 2025;118(13):261–77. doi:10.1016/j.aej.2025.01.042.
11. Bakhsh SA, Khan MA, Ahmed F, Alshehri MS, Ali H, Ahmad J. Enhancing IoT network security through deep learning-powered intrusion detection system. Internet Things. 2023;24(10):100936. doi:10.1016/j.iot.2023.100936.
12. Orman A. Cyberattack detection systems in industrial Internet of Things (IIoT) networks in big data environments. Appl Sci. 2025;15(6):3121. doi:10.3390/app15063121.
13. Salayma M. Risk and threat mitigation techniques in Internet of Things (IoT) environments: a survey. Front Internet Things. 2024;2:1306018. doi:10.3389/friot.2023.1306018.
14. Zhukabayeva T, Zholshiyeva L, Karabayev N, Khan S, Alnazzawi N. Cybersecurity solutions for industrial Internet of Things—edge computing integration: challenges, threats, and future directions. Sensors. 2025;25(1):213. doi:10.3390/s25010213.
15. Wakili A, Bakkali S. Privacy-preserving security of IoT networks: a comparative analysis of methods and applications. Cyber Secur Appl. 2025;3(4):100084. doi:10.1016/j.csa.2025.100084.

16. Khayat M, Barka E, Serhani MA, Sallabi F, Shuaib K, Khater HM. Reinforcement learning with deep features: a dynamic approach for intrusion detection in IoT networks. IEEE Access. 2025;13:92319–37. doi:10.1109/access.2025.3569312.
17. Chiba Z, Abghour N, Moussaid K, Lifandali O, Kinta R. A deep study of novel intrusion detection systems and intrusion prevention systems for Internet of Things networks. Procedia Comput Sci. 2022;210(15):94–103. doi:10.1016/j.procs.2022.10.124.
18. Mallidi SKR, Ramisetty RR. Advancements in training and deployment strategies for AI-based intrusion detection systems in IoT: a systematic literature review. Discov Internet Things. 2025;5(1):8. doi:10.1007/s43926-025-00099-4.
19. Usama Tanveer M, Munir K, Amjad M, Ali Jafar Zaidi S, Bermak A, Rehman AU. Ensemble-guard IoT: a lightweight ensemble model for real-time attack detection on imbalanced dataset. IEEE Access. 2024;12:168938–52. doi:10.1109/access.2024.3495708.
20. Musthafa MB, Huda S, Kodera Y, Ali MA, Araki S, Mwaura J, et al. Optimizing IoT intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques. Sensors. 2024;24(13):4293. doi:10.3390/s24134293.
21. Shafin SS, Karmakar G, Mareels I, Balasubramanian V, Kolluri RR. Sensor self-declaration of numeric data reliability in Internet of Things. IEEE Trans Reliab. 2025;74(2):2751–65. doi:10.1109/tr.2024.3416967.
22. Chen X, Wang P, Yang Y, Liu M. Resource-constraint deep forest-based intrusion detection method in Internet of Things for consumer electronic. IEEE Trans Consum Electron. 2024;70(2):4976–87. doi:10.1109/tce.2024.3373126.
23. Simioni JA, Viegas EK, Santin AO, de Matos E. An energy-efficient intrusion detection offloading based on DNN for edge computing. IEEE Internet Things J. 2025;12(12):20326–42. doi:10.1109/jiot.2025.3544060.
24. Almohaimeed M, Albalwy F. Enhancing IoT network security using feature selection for intrusion detection systems. Appl Sci. 2024;14(24):11966. doi:10.3390/app142411966.
25. Alturki B, Alsulami AA. Semi-supervised learning with entropy filtering for intrusion detection in asymmetrical IoT systems. Symmetry. 2025;17(6):973. doi:10.3390/sym17060973.
26. Albalwy F, Almohaimeed M. Advancing artificial intelligence of things security: integrating feature selection and deep learning for real-time intrusion detection. Systems. 2025;13(4):231. doi:10.3390/systems13040231.
27. Prasad A, Mohammad Alenazy W, Ahmad N, Ali G, Abdallah HA, Ahmad S. Optimizing IoT intrusion detection with cosine similarity based dataset balancing and hybrid deep learning. Sci Rep. 2025;15(1):30939. doi:10.1038/s41598-025-15631-3.
28. Katsura Y, Endo A, Arai I, Fujikawa K. Efficient IDS for IoT networks using host-based data aggregation and multi-entropy analysis. IEEE Access. 2025;13(21):125406–19. doi:10.1109/access.2025.3589057.
29. Vishwakarma M, Kesswani N. StaEn-IDS: an explainable stacking ensemble deep neural network-based intrusion detection system for IoT. IEEE Access. 2025;13:109713–28. doi:10.1109/access.2025.3582391.
30. Ullah F, Turab A, Ullah S, Cacciagrano D, Zhao Y. Enhanced network intrusion detection system for Internet of Things security using multimodal big data representation with transfer learning and game theory. Sensors. 2024;24(13):4152. doi:10.3390/s24134152.
31. Mahdi ZS, Zaki RM, Alzubaidi L. A secure and adaptive framework for enhancing intrusion detection in IoT networks using incremental learning and blockchain. Secur Priv. 2025;8(4):e70071. doi:10.1002/spy2.70071.
32. Houichi M, Jaidi F, Bouhoula A. Enhancing smart city security: an intrusion detection system using machine learning methods with the UNB CIC IoT 2023 dataset. IET Smart Cities. 2025;7(1):e70014. doi:10.1049/smc2.70014.
33. Mahdi ZS, Zaki RM, Alzubaidi L. Advanced hybrid techniques for cyberattack detection and defense in IoT networks. Secur Priv. 2025;8(2):e471. doi:10.1002/spy2.471.
34. Doménech J, León O, Siddiqui MS, Pegueroles J. Evaluating and enhancing intrusion detection systems in IoMT: the importance of domain-specific datasets. Internet Things. 2025;32(1):101631. doi:10.1016/j.iot.2025.101631.
35. Hafid A, Rahouti M, Aledhari M. Optimizing intrusion detection in IoMT networks through interpretable and cost-aware machine learning. Mathematics. 2025;13(10):1574. doi:10.3390/math13101574.
36. Mezina A, Nurmi J, Ometov A. Novel hybrid UNet++ and LSTM model for enhanced attack detection and classification in IoMT traffic. IEEE Access. 2025;13(1):57589–603. doi:10.1109/access.2025.3553966.

37. Haq MIU, Mahmood K, Li Q, Das AK, Shetty S, Hussain M. Efficiently learning an encoder that classifies token replacements and masked permuted network-based BIGRU attention classifier for enhancing sentiment classification of scientific text. IEEE Access. 2024;12(3):190240–54. doi:10.1109/access.2024.3516946.
38. Regler B, Scheffler M, Ghiringhelli LM. TCMI: a non-parametric mutual-dependence estimator for multivariate continuous distributions. Data Min Knowl Discov. 2022;36(5):1815–64. doi:10.1007/s10618-022-00847-y.
39. Su Y, Xiong D, Wan Y, Shi C, Zeng Q. LinFuzz: program-sensitive seed scheduling greybox fuzzing based on LinUCB algorithm. IEEE Access. 2024;12(10):74843–60. doi:10.1109/access.2024.3404918.
40. Sharmila B, Nagapadma R. Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset. Cybersecurity. 2023;6(1):41. doi:10.1186/s42400-023-00178-5.
41. Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.
42. Dadkhah S, Neto ECP, Ferreira R, Molokwu RC, Sadeghi S, Ghorbani AA. CICIoMT2024: a benchmark dataset for multi-protocol security assessment in IoMT. Internet Things. 2024;28:101351.