Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082207

ARTICLE

Lightweight AI-Powered Intrusion Detection via Edge Computing

Jackson Diaz-Gorrin[1,*], Candido Caballero-Gil[1], Pino Caballero-Gil[1] and Joanna Kolodziej[2,3]

[1]Department of Computer Engineering and Systems, University of La Laguna, La Laguna, Spain
[2]Department of Computer Sciences, Cracow University of Technology, Cracow, Poland
[3]Naukowa i Akademicka Sieć Komputerowa-Państwowy Instytut Badawczy (NASK-PIB), ul. Kolska 12, Warszawa, Poland

*Corresponding Author: Jackson Diaz-Gorrin. Email: jhdiazgo@ull.edu.es

Received: 12 March 2026; Accepted: 28 May 2026; Published: 23 July 2026

ABSTRACT: A lightweight flow-based intrusion detection system is proposed for identifying Mirai-based distributed denial-of-service attacks in Internet of Things (IoT) environments. Efficient intrusion detection at the network edge is essential for resource-constrained IoT deployments, where devices operate with limited processing, memory, and energy resources, making centralized or computationally intensive solutions impractical in real-world scenarios. Network traffic is represented using statistical and temporal features extracted from unidirectional flows constructed from the TII-SSRC-23 dataset. A balanced subset of 10,000 samples is used for training and evaluation, ensuring balanced data distribution and improving generalization across different traffic conditions. Three machine learning models, a multilayer perceptron, a support vector machine, and LightGBM, are investigated to evaluate trade-offs between detection performance, complexity, and suitability for deployment in resource-constrained edge environments. Experimental results show that LightGBM achieves the best performance, obtaining an accuracy of 0.99, an F1-score of 1.00, and an AUC of 1.00, while consistently maintaining a low false positive rate. The selected model is deployed on the NVIDIA Jetson Orin Nano platform for real-time inference under resource constraints and evaluated for continuous operational performance. The system operates with low latency and reduced memory and computational requirements, making it highly suitable for edge IoT security scenarios.

KEYWORDS: Cybersecurity; intrusion detection; machine learning; internet of things; edge computing; artificial intelligence

1 Introduction

The rapid proliferation of devices on the Internet of Things (IoT) has significantly transformed modern digital ecosystems, enabling applications such as smart homes, industrial automation, intelligent transportation systems, and healthcare monitoring. Despite their widespread adoption, many IoT devices are deployed with limited security mechanisms and are often directly exposed to the Internet. These limitations make them attractive targets for cyber attacks and, in many cases, effective tools for launching large-scale malicious campaigns [1,2].

Among the most impactful cyber threats to networked systems are Denial-of-Service (DoS) attacks, which aim to disrupt service availability by exhausting critical resources such as bandwidth, memory, or processing capacity. When such attacks are executed simultaneously from multiple compromised devices, they evolve into Distributed Denial-of-Service (DDoS) attacks, significantly amplifying their effectiveness and complicating mitigation efforts. In IoT-enabled infrastructures, DDoS attacks can cause severe service degradation, loss of availability, and cascading failures across edge and cloud components [3–5].

Advances in edge and fog computing have enabled distributed processing closer to IoT devices, allowing for low-latency analysis, local decision-making, and real-time security enforcement [6–8]. By offloading computation from centralized cloud servers to edge and fog nodes, these paradigms can support scalable intrusion detection system (IDS) capable of monitoring vast numbers of heterogeneous IoT devices while reducing network congestion and response times. Several fog-based cybersecurity frameworks have demonstrated the effectiveness of integrating machine learning and flow-based traffic analysis at intermediate nodes to detect and mitigate attacks before they propagate through the network. This distributed approach provides a complementary foundation for our edge-deployed IDS, ensuring rapid threat detection and mitigation while maintaining system efficiency in resource-constrained environments.

A prominent example of IoT-driven DDoS attacks is the Mirai malware [9]. Mirai specifically targets vulnerable IoT devices by exploiting weak or default authentication credentials and incorporates them into large botnets capable of generating massive volumes of malicious traffic [10]. These botnets have demonstrated the ability to launch various DDoS attack variants, including volumetric, protocol-based, and application-layer attacks. The scale, heterogeneity, and geographic distribution of compromised IoT devices make Mirai-based DDoS attacks particularly challenging to detect and mitigate [11].

This work addresses the problem of detecting Mirai-based DDoS attacks using a machine-learning-based IDS designed for deployment at the network edge [12]. By identifying anomalous traffic patterns in real time, the proposed system enables early detection and timely countermeasures, reducing the impact of attacks before service availability is compromised. Although IDS is primarily focused on Mirai-induced DDoS traffic, its design allows it to be generalized to other DDoS attack scenarios with similar network characteristics.

Traditional endpoint-based intrusion detection systems, as commonly described in the literature on IoT security frameworks [13,14], typically operate at the device level and rely on host-based monitoring mechanisms or centralized post-facto analysis. In contrast, the proposed framework performs flow-based detection directly at the network edge. This design eliminates the need for endpoint instrumentation, reduces computational overhead on constrained IoT devices, and enables earlier detection by analyzing aggregated traffic flows rather than individual host behavior. Furthermore, the system is specifically optimized for real-time execution on resource-constrained embedded hardware, such as the NVIDIA Jetson Orin Nano, allowing continuous monitoring and classification of network traffic without requiring cloud connectivity or centralized processing.

To evaluate the feasibility of deploying such a system in resource-constrained environments, the IDS is implemented on an edge computing platform. The NVIDIA Jetson Orin Nano provides a compact and energy-efficient hardware solution capable of supporting real-time machine learning inference, making it well-suited for continuous traffic monitoring in IoT and edge computing scenarios [15].

This paper is organized as follows. Section 2 presents the preliminaries, including the Mirai-based DDoS attack model, the network protocols exploited in such attacks, the embedded edge platform used for deployment, and the evaluation metrics. Section 3 reviews related work on intrusion detection systems for IoT and Mirai-driven DDoS attacks. Section 4 describes the proposed methodology, detailing the flow-based traffic analysis, machine learning models, and edge deployment strategy. Section 5 reports the experimental setup, dataset, implementation, and performance evaluation of the proposed system. Section 6 discusses and analyzes the obtained results, highlighting performance and deployment considerations. Finally, Section 7 concludes the paper and outlines directions for future work.

The main contributions of this work are summarized as follows:

- A lightweight, flow-based IDS designed for detecting Mirai-based DDoS attacks in IoT environments, optimized for deployment on resource-constrained edge devices.
- A real-time flow-based traffic analysis pipeline enabling efficient extraction of statistical and temporal features for low-latency intrusion detection.
- An edge-based implementation and performance evaluation on the NVIDIA Jetson Orin Nano, analyzing throughput, latency, CPU utilization, memory usage, and energy consumption.
- An integrated IDS architecture that combines flow-based machine learning detection with adaptive response mechanisms, such as traffic steering and rate limiting, for edge-based IoT environments.

This study investigates the end-to-end design and deployment of a flow-based intrusion detection pipeline optimized for real-time execution on resource-constrained edge hardware, bridging the gap between offline model evaluation and practical edge deployment in IoT environments.

2 Preliminaries

This section presents the technical background required to understand the detection approach proposed in this work. Rather than redefining general denial-of-service concepts, the focus is placed on the Mirai-based DDoS attack model, the protocol-level characteristics of the generated traffic, and the system assumptions relevant to intrusion detection in IoT environments.

2.1 Mirai-Based DDoS Attack Model

Mirai-based DDoS attacks are characterized by the coordinated behavior of a large number of compromised IoT devices acting as traffic sources under centralized control. Once infected, these devices participate in attack campaigns by generating malicious traffic toward designated targets, often at relatively low individual transmission rates that aggregate into high-impact distributed attacks.

From a network monitoring perspective, Mirai botnets exhibit heterogeneous traffic patterns because they can launch multiple attack types across different network protocols and layers. The large scale of IoT deployments, combined with the limited resources and homogeneous behavior of embedded devices, results in traffic dynamics that are difficult to distinguish from legitimate activity without fine-grained analysis.

In this work, Mirai-based DDoS attacks are modeled as distributed traffic anomalies originating from IoT devices and targeting network bandwidth, protocol state resources, or application-layer services. While the IDS is primarily designed for Mirai-induced traffic, the underlying detection principles are applicable to other DDoS attacks with comparable behavioral characteristics.

Fig. 1 illustrates a conceptual Mirai-based DDoS attack scenario in which multiple compromised IoT devices generate coordinated malicious traffic toward a target system. This model highlights the importance of early detection mechanisms deployed at the network edge.

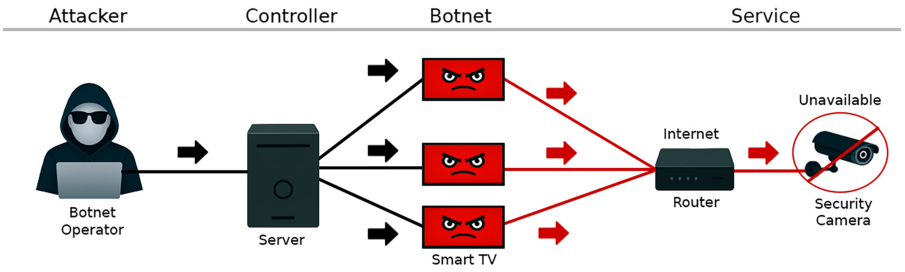


Figure 1: Conceptual illustration of a Mirai-based DDoS attack leveraging compromised IoT devices.

2.2 Network Protocols Exploited in DDoS Attacks

Mirai-based DDoS attacks exploit characteristics of network, transport, and application-layer protocols to maximize their impact. Understanding these protocol-level behaviors is essential for extracting discriminative features used by the IDS.

2.3 GRE-Based Flooding Attacks

Generic Routing Encapsulation (GRE) is a tunneling protocol that encapsulates network-layer packets within other protocols to establish virtual point-to-point links [16]. In DDoS scenarios, GRE can be abused to generate large volumes of encapsulated traffic, bypassing certain filtering mechanisms and overwhelming network-level resources.

2.4 HTTP Application-Layer DDoS Attacks

Hypertext Transfer Protocol (HTTP) operates at the application layer and is commonly targeted by DDoS attacks that issue a high volume of seemingly legitimate requests. These attacks primarily exhaust server-side processing resources rather than network bandwidth, making them difficult to mitigate using traditional volumetric defenses [17].

2.5 DNS Infrastructure DDoS Attacks

The Domain Name System (DNS) is a critical component of the Internet, responsible for resolving domain names to IP addresses. DDoS attacks targeting DNS infrastructure generate excessive query traffic, disrupting name resolution services and indirectly affecting all dependent applications [18].

2.6 Volumetric UDP Flooding Attacks

User Datagram Protocol (UDP) is a connectionless transport-layer protocol with minimal overhead and no built-in flow control. These properties enable high-throughput communication but also make UDP susceptible to volumetric flooding attacks that saturate network bandwidth and processing resources [19].

2.7 TCP SYN and ACK Flood Attacks

Transmission Control Protocol (TCP) ensures reliable, ordered communication through connection management. DDoS attacks exploit these mechanisms through:

- **SYN Flood:** A large number of connection initiation requests are sent without completing the handshake, exhausting server-side connection resources [20].
- **ACK Flood:** High volumes of TCP ACK packets are transmitted to overload the target's processing and memory resources.

2.8 Jetson Orin Nano Platform

The Jetson Orin Nano is an embedded computing platform designed for edge artificial intelligence applications, delivering up to 67 INT8 TOPS of AI performance, which enables real-time execution of demanding deep learning workloads such as computer vision and robotics. It integrates a GPU based on the NVIDIA Ampere architecture, featuring 1024 CUDA cores and 32 Tensor Cores, providing efficient acceleration for AI inference, while a 6-core Arm Cortex-A78AE v8.2 64-bit CPU ensures robust general-purpose and deterministic processing. The system is equipped with 8 GB of 128-bit LPDDR5 memory, offering a high memory bandwidth of 102 GB/s, which is critical for data-intensive applications. Storage flexibility is supported through an SD card slot and external NVMe devices, allowing scalable data handling.

Additionally, its configurable power envelope ranging from 7 to 25 W enables deployment across a wide range of embedded scenarios, balancing energy efficiency and computational performance in compact edge systems.

Several studies on edge-based artificial intelligence and embedded systems have analyzed the performance of resource-constrained platforms, highlighting key trade-offs between computational efficiency, power consumption, and inference capability. Benchmarking analyses of single-board devices such as NVIDIA Jetson Nano, Jetson TX2, and Raspberry Pi have shown that while low-cost platforms enable the deployment of deep learning models, their performance is strongly influenced by hardware constraints, including CPU, GPU, memory usage, and power consumption. In particular, the availability of GPU acceleration has been identified as a critical factor for improving inference speed and enabling more complex models in embedded environments. More recent comparative studies of edge AI platforms further demonstrate that newer-generation devices significantly improve inference throughput and efficiency, achieving higher processing rates and more stable thermal behavior under sustained workloads [21]. These findings highlight the importance of selecting appropriate hardware to balance latency, energy consumption, and computational performance in real-time applications. In this context, the NVIDIA Jetson Orin Nano builds upon these advancements by offering increased processing capability while maintaining energy-efficient operation, enabling faster flow classification and supporting more complex machine learning models for scalable IoT intrusion detection deployments.

2.9 Evaluation Metrics

To evaluate the performance of the proposed network traffic classification model in distinguishing between *Benign* and *Malicious* traffic, the following standard metrics are used:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall (Detection Rate)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Fl-Score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

where:

- TP (True Positives): Malicious network traffic correctly classified as malicious by the model.
- TN (True Negatives): Benign network traffic correctly classified as benign.
- FP (False Positives): Benign traffic incorrectly classified as malicious.
- FN (False Negatives): Malicious traffic incorrectly classified as benign.

In the context of network traffic classification, Recall (Detection Rate) is particularly important because false negatives correspond to undetected attacks. Precision is also essential to minimize false alarms and ensure reliable threat detection.

3 Related Work

In [22], a hybrid IDS for IoT networks is proposed that integrates Support Vector Machine (SVM) and Genetic Algorithm (GA) for feature selection and parameter optimization. An experimental evaluation

on a benchmark dataset demonstrates high detection accuracy with a low false-positive rate, including effective identification of complex attack types. The results further show that the approach scales well while maintaining acceptable latency and computational overhead in large-scale IoT deployments.

The study reported in [23] analyzes the evolution of the Mirai botnet over a six-year period (2016–2022), focusing on its scanning behavior and exploitation of network device vulnerabilities. The analysis reveals that Mirai's signature remains active, with an increasing number of hijacked devices and TCP SYN packets over time. It also identifies the primary targets, such as Telnet ports 23 and 2323, and highlights the emergence of new variants affecting previously unused ports, providing insights into the botnet's long-term adaptation and growth.

A flow-based botnet detection approach based on a hybrid CNN-LSTM model is proposed in [24] to capture both spatial and temporal patterns in network traffic. By combining convolutional neural networks for spatial feature extraction with LSTM for temporal sequence learning, the method provides a more comprehensive analysis of botnet behavior. The study further evaluates the approach against adversarial attacks, demonstrating its robustness and effectiveness in accurately identifying malicious network traffic across different datasets.

An intelligent IDS for IoT networks is developed in [25] using deep learning to address diverse security and privacy challenges. The study evaluates multiple neural network models and an ensemble learning approach on large-scale IoT data, showing that deep learning techniques can significantly improve the detection of network intrusions and enhance overall IoT network security.

A review of intrusion detection in the Internet of Medical Things (IoMT) with a focus on Digital Twins (DTs) is presented in [26]. The study highlights how DTs, by simulating device behavior and network interactions, provide a proactive and adaptive platform for early threat detection in complex IoMT environments. The review identifies current trends, challenges, and opportunities in integrating IDS and DT technologies, emphasizing their potential to enhance scalability, improve detection accuracy, and mitigate evolving security threats in healthcare networks.

A comprehensive review of Artificial Intelligence (AI) techniques for intrusion detection in Internet of Things (IoT) ecosystems is presented in [27]. The work systematically reviews state-of-the-art methodologies, including machine learning, deep learning, and anomaly detection algorithms, highlighting their effectiveness in identifying and mitigating diverse cyber threats. It further examines the unique characteristics and vulnerabilities of IoT environments that necessitate specialized IDS solutions, while evaluating the performance, accuracy, and scalability of existing approaches. Additionally, the study discusses key implementation challenges, such as resource constraints, device heterogeneity, and dynamic network topologies, and outlines future research directions to enhance the resilience and security of IoT systems against evolving cyber threats.

A novel approach to enhance the adversarial robustness of deep neural networks (DNNs) in Internet of Things (IoT) environments through self-supervised adversarial learning is proposed in [28]. The study addresses key limitations of traditional adversarial training, particularly its dependence on labeled data and its reduced effectiveness against unseen attacks. To overcome these challenges, the authors introduce adversarial data augmentations in the latent space combined with contrastive learning techniques, along with new loss functions designed to prevent representation collapse. Building on these contributions, the paper presents an Adversarial Robust Learning (ARL) framework that enables label-free adversarial training and produces more generalized and robust feature representations. Experimental results on benchmark datasets demonstrate that the proposed method achieves competitive robustness against diverse adversarial attacks while improving clean prediction accuracy, outperforming existing self-supervised adversarial learning approaches.

An online ensemble learning framework for anomaly detection in Internet of Things (IoT) systems is proposed in [29], addressing the challenges of concept drift and data imbalance in dynamic environments. The study introduces the Adaptive Exponentially Weighted Average Ensemble (AEWAE), a drift-aware model composed of data preprocessing, base learner training, and online ensembling stages, integrating multiple advanced online learning algorithms. To optimize its performance, key parameters are fine-tuned using Particle Swarm Optimization (PSO). Experimental evaluations on several public datasets demonstrate that the proposed approach effectively detects anomalies and adapts to evolving data distributions, outperforming baseline methods in terms of accuracy, F1 score, false alarm rate, and latency, making it suitable for real-time IoT applications.

While the aforementioned works demonstrate significant advances in intrusion detection for IoT environments, several limitations remain. Many approaches focus primarily on improving detection accuracy through complex models such as deep learning or ensemble techniques, often assuming resource-rich or centralized deployment settings. Other studies address adversarial robustness or adaptive learning, but do not consider the constraints of real-time execution in edge devices. Additionally, limited attention has been given to lightweight, low-power solutions capable of performing continuous traffic analysis and mitigation directly at the network edge. These limitations highlight the need for efficient and deployable intrusion detection mechanisms tailored to resource-constrained IoT environments.

The novelty of this work lies in the design of a low-power, real-time IDS based on a machine learning model deployed on the NVIDIA Jetson Orin Nano, which is used as the edge computing platform for real-time inference in IoT networks. By performing real-time network traffic filtering and analysis directly at the edge, the proposed system enables early detection and mitigation of Mirai-based DDoS attacks, reducing the exposure of IoT devices to large-scale compromise. This proactive approach helps prevent common consequences of botnet propagation, such as service disruption, resource exhaustion, and unauthorized participation in distributed attacks. Furthermore, the low-power edge-based design minimizes communication overhead and latency, allowing timely defensive actions while preserving the operational stability and longevity of IoT deployments.

4 Methodology

For the implementation of the detection system on the NVIDIA Jetson Orin Nano, three machine learning models were considered: a multilayer perceptron (MLP), a linear support vector machine (SVM), and LightGBM. These models were selected because they provide complementary learning capabilities while maintaining relatively low computational requirements, making them suitable for deployment in resource-constrained embedded environments.

The models were implemented and trained in Python using the Scikit-learn library, while Scapy was employed for real-time packet capture and preprocessing. After training on a standard workstation, the resulting models were serialized in pickle format and deployed on the NVIDIA Jetson Orin Nano for real-time inference.

Rather than performing packet-level classification, the proposed system adopts a flow-based traffic analysis approach. Captured packets are aggregated into unidirectional flows according to source and destination IP addresses, transport-layer ports, and protocol type. For each completed flow, or when a predefined timeout is reached, a set of statistical and temporal features is extracted to characterize communication behavior over time.

Each feature vector is then processed by a pre-trained supervised classifier, which labels the flow as either benign or malicious. This continuous process enables real-time detection of Mirai-based distributed denial-of-service (DDoS) attacks while maintaining low computational overhead on the embedded platform. Fig. 2 presents an overview of the proposed detection pipeline.

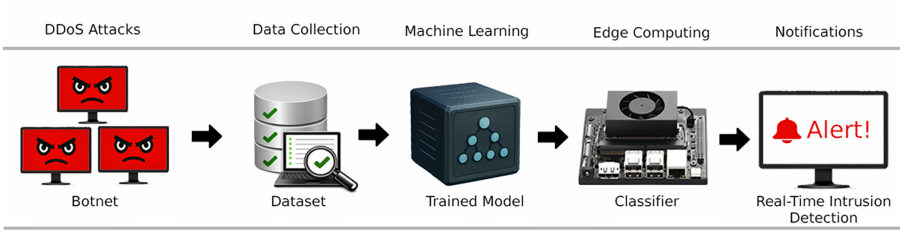


Figure 2: Overview of the real-time flow-based intrusion detection pipeline deployed on the Jetson Orin Nano.

The proposed intrusion detection framework follows a two-stage workflow consisting of an offline training phase and a real-time deployment phase. In the offline stage, the TII-SSRC-23 dataset is used to train and evaluate the candidate models, and the best-performing model is selected together with its preprocessing pipeline. In the deployment stage, the trained model is integrated into the NVIDIA Jetson Orin Nano for real-time flow-based inference on live network traffic. The system continuously classifies flows and generates alerts upon detection of malicious activity, ensuring consistency between development and deployment while enabling efficient intrusion detection on a resource-constrained edge platform.

Fig. 3 illustrates the deployment scenario used to protect IoT devices from Mirai-based DDoS attacks. Traffic directed toward the monitored device is analyzed by the Jetson Orin Nano, which operates as an inline IDS. When malicious activity is detected, the system can generate alerts and trigger mitigation actions before the attack disrupts the target device.

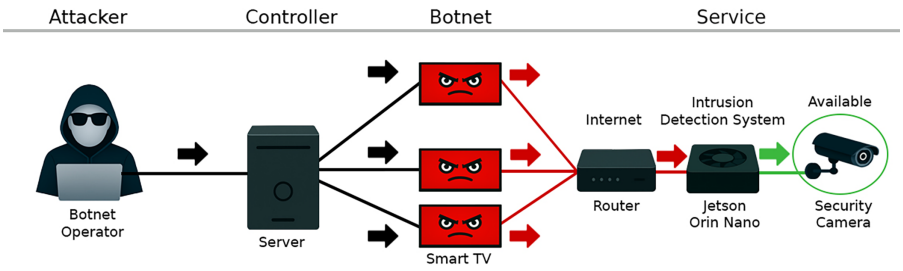


Figure 3: Mirai DDoS attack scenario and early mitigation using a Jetson Orin Nano–based intrusion detection system.

4.1 Mitigation

In addition to real-time intrusion detection, the proposed framework includes a design-level mitigation component that operates on the output of the flow-based IDS. Once a network flow is identified as malicious, the system defines a set of possible response actions intended to reduce the impact of the attack while preserving the availability of legitimate services.

The mitigation process follows a decision-driven approach based on the confidence score provided by the machine learning classifier, which determines the severity of the response. Low-confidence detections are associated with lightweight mitigation actions, whereas high-confidence detections are mapped to stronger response policies. This adaptive strategy is presented as a conceptual mechanism to handle classification uncertainty and to minimize potential disruption to legitimate traffic.

Traffic redirection using Generic Routing Encapsulation (GRE) tunnels is considered as a potential mechanism for isolating suspicious flows by encapsulating and redirecting them to dedicated processing endpoints for filtering or inspection.

Segment routing (SR) policies are also considered as a possible traffic engineering mechanism, where flows could be directed toward different paths based on classifier outputs, allowing separation between suspicious and legitimate traffic in a future deployment scenario.

An adaptive throttling mechanism with feedback control is also defined as part of the proposed design, where rate limits could be dynamically adjusted based on classifier confidence and observed traffic characteristics, with the aim of reducing attack impact while preserving legitimate traffic performance.

All mitigation operations are part of the proposed system design and are conceptually integrated with the edge-based detection pipeline running on the NVIDIA Jetson Orin Nano. The framework therefore represents a unified detection-and-response architecture intended for IoT security scenarios.

To ensure long-term adaptability, the framework supports periodic retraining using newly collected flow data, as well as potential extensions based on online learning and federated learning approaches. These mechanisms are intended to improve robustness against evolving attack patterns and to support distributed IoT deployments while preserving data privacy.

To extend the framework to larger and more complex IoT environments, a distributed multi-edge architecture is considered, where local nodes perform flow-based detection and a central coordinator aggregates alerts and threat intelligence. This hierarchical design supports scalable monitoring and coordinated response strategies across large-scale IoT deployments.

4.2 Threat Model and System Assumptions

The proposed intrusion detection system is designed for deployment in an edge-based IoT security scenario, where the NVIDIA Jetson Orin Nano is placed at the network edge between the local IoT environment and the upstream gateway. In this configuration, the device is able to observe all traffic passing through this point and perform real-time analysis of network flows generated by IoT devices and external sources.

The system operates under a flow-based observation model, where only network-level information is used for detection. Packet payloads are not considered at any stage of the analysis, and the decision process relies exclusively on flow-derived statistical, temporal, and transport-layer header information as defined in the feature extraction stage.

Regarding the threat model, the attacker is assumed to operate as an external adversary controlling compromised IoT devices within a Mirai-like botnet. The attacker's objective is to generate attack traffic targeting IoT services in the monitored network. The attacker is assumed to act in a black-box setting, with no knowledge of the internal structure of the detection system, including model parameters, feature representation, or deployment configuration.

Under these assumptions, the proposed IDS focuses on identifying malicious behavior based on observable flow characteristics such as traffic volume, temporal patterns, and protocol-level interactions.

4.3 Feature Importance and Model Interpretability

To enhance the interpretability of the trained models and better understand which traffic attributes contribute most to attack detection, an additional analysis was conducted based on both intrinsic feature importance from the LightGBM model and SHAP (Shapley Additive Explanations). This was carried out after model training as part of the evaluation of model behavior.

For the LightGBM model, feature importance was derived from the total information gain accumulated across all decision trees. This metric reflects how frequently and effectively each feature contributes to reducing uncertainty during split decisions, resulting in a global ranking of feature relevance.

To complement this global view, SHAP values were used to quantify the marginal contribution of each feature to individual predictions. SHAP is grounded in cooperative game theory, and the TreeExplainer method was applied to efficiently compute values for the trained model using the validation dataset and the final trained model. Global interpretability was assessed through mean absolute SHAP values, while local interpretability was examined by analyzing SHAP value distributions at the instance level.

This analysis provides both global and local interpretability, enabling a clearer understanding of the most influential flow-based features for Mirai-based DDoS detection and how variations in these features affect individual classification outcomes.

5 Experiments

This section presents the experimental setup, including the dataset, preprocessing steps, and the machine learning models employed for detecting Mirai-based DDoS attacks in IoT networks. The primary objective of these experiments is to evaluate and compare the performance of three classifiers, MLP, SVM, and LightGBM, in distinguishing between benign and malicious network flows, using a representative subset of IoT traffic.

5.1 Dataset

The experiments are conducted using the TII-SSRC-23 dataset [30], a large-scale IoT network traffic dataset containing benign and malicious flows generated under realistic Internet of Things scenarios. The dataset includes traffic collected under multiple IoT communication patterns and diverse Distributed Denial-of-Service (DDoS) attack conditions, making it suitable for evaluating intrusion detection systems in edge environments. From the full dataset, a subset was selected focusing specifically on Mirai-based DDoS attack traffic and corresponding benign IoT communication flows.

The original dataset contains millions of raw packet-level records, which are transformed into flow-level representations prior to analysis. From these flow-level records, 10,000 samples were initially selected to ensure a controlled and computationally feasible experimental setup. To construct a balanced dataset, an equal number of benign and malicious flows were subsequently selected, resulting in a final dataset of 10,000 samples (5000 samples per class). This dataset is used for model training and evaluation. To avoid data leakage, the Synthetic Minority Over-sampling Technique (SMOTE) is applied only to the training set after the train-test split.

The class labels used for supervised learning were obtained directly from the annotations provided in the dataset. Each flow record includes a Label attribute indicating whether the traffic is categorized as *Benign* or *Malicious*, together with Traffic Type and Traffic Subtype attributes that provide a more detailed description of the corresponding traffic scenario. During preprocessing, the Traffic Subtype attribute was used to select the benign traffic categories and the Mirai-based DDoS attack variants considered in this study. The original Label values were then mapped to a binary representation, where *Benign* was encoded as 0 and *Malicious* as 1. No manual labeling was performed. Label consistency was verified by confirming that all selected traffic subtypes matched their corresponding annotations in the original dataset.

Benign traffic includes various legitimate network activities, such as audio streaming, background services, text exchanges, and different types of video traffic. Table 1 summarizes the benign traffic subtypes and their corresponding descriptions used for training and evaluation.

Table 1: Benign traffic subtypes considered in the study.

Traffic Subtype	Description
Audio	Multimedia audio streaming traffic.
Background	Network traffic from system services or background applications.
Text	Low-bandwidth textual communication, e.g., messaging.
Video HTTP	Video streaming over HTTP.
Video RTP	Real-time video streaming via RTP.
Video UDP	Video transmitted over raw UDP without higher-layer protocols.

Malicious traffic in the selected subset includes multiple Mirai-based DDoS attack types, namely UDP flooding, TCP SYN flooding, TCP ACK flooding, GRE-based attacks, HTTP flooding, and DNS flooding attacks, which collectively cover volumetric, protocol-level, and application-layer attack vectors.

Table 2 lists these attack subtypes.

Table 2: Mirai-based DDoS attack variants.

Attack Type	Description
Mirai DDoS UDP	High-rate UDP flooding attack.
Mirai DDoS ACK	TCP ACK flooding to exhaust server resources.
Mirai DDoS SYN	TCP SYN flooding targeting connection establishment.
Mirai DDoS GREIP	GRE-over-IP flooding attack.
Mirai DDoS GREETH	GRE-over-Ethernet flooding attack.
Mirai DDoS HTTP	Application-layer HTTP DDoS.
Mirai DDoS DNS	DNS flooding to disrupt name resolution.

5.2 Feature Extraction Process

Traffic flows are constructed using a flow-based aggregation mechanism implemented with Scapy, where packets are grouped into unidirectional flows based on a shared 5-tuple: source IP address, destination IP address, source port, destination port, and transport-layer protocol. A flow is considered complete either after a timeout period or when no additional packets are observed within a predefined inactivity threshold.

For each flow, statistical and temporal features are extracted to characterize communication behavior. These include packet-level statistics such as packet count, mean and variance of packet sizes, temporal dynamics such as flow duration and inter-arrival time statistics, and protocol-level indicators such as TCP flag counts (SYN, ACK, RST). This abstraction enables the system to capture behavioral patterns of DDoS attacks at the flow level, rather than relying on individual packet-level anomalies, thereby improving robustness in real-world IoT environments.

Because the proposed IDS relies exclusively on flow-level statistical, temporal, and protocol-header features, it does not require access to packet payload contents. Consequently, the system remains applicable when communications are protected using encryption protocols such as Transport Layer Security (TLS), since metadata including packet-size statistics, timing-related information, transport-layer ports, and protocol flags remain accessible without decrypting the traffic.

5.3 Dataset Subset Selection

The subset of the TII-SSRC-23 dataset used in this study was selected to focus specifically on Mirai-based DDoS attack scenarios, which are representative of real-world IoT botnet behavior. This selection reduces dataset complexity while preserving the most relevant attack patterns required for effective intrusion detection.

Furthermore, the use of a reduced and balanced dataset ensures computational feasibility for both training and deployment on resource-constrained edge devices such as the NVIDIA Jetson Orin Nano. By operating on flow-level representations and enforcing class balance during dataset construction, the dataset enables efficient model learning while maintaining realism in terms of IoT network traffic behavior and attack diversity.

To improve representativeness, the selected subset was constructed following a stratified sampling strategy, ensuring coverage across different Mirai-based DDoS attack variants and benign IoT traffic categories present in the TII-SSRC-23 dataset. This approach ensures that the reduced dataset retains the diversity of traffic behaviors required for intrusion detection while enabling a computationally feasible experimental setup for edge deployment.

This design choice follows standard practices in intrusion detection literature, where balanced and reduced datasets are commonly used to mitigate computational constraints while preserving the discriminative characteristics of network traffic. Although the full dataset is not used in its entirety, the inclusion of multiple attack vectors ensures that the learned models capture representative patterns of both normal and malicious IoT communications.

5.4 Machine Learning Models

The proposed intrusion detection system employs three complementary machine learning models: Multilayer Perceptron (MLP), Support Vector Machine (SVM) with linear kernel, and LightGBM. These models were specifically selected to capture different perspectives on the classification problem while remaining suitable for deployment on resource-constrained edge devices. The hyperparameter configurations used for all models are summarized in Table 3.

Table 3: Hyperparameter settings of the evaluated machine learning models.

Model	Hyperparameter	Value
MLP	Hidden layers	(50, 50)
	Activation function	ReLU
	Optimizer	Adam
	Learning rate	0.001
	Max iterations	500
SVM	Kernel	Linear
	C	1.0
	Probability estimates	True
	Tolerance	0.001

(Continued)

Table 3 (continued)

Model	Hyperparameter	Value
LightGBM	Objective	Binary
	Metric	binary_logloss
	Number of estimators	100
	Learning rate	0.1
	Max depth	−1
	Number of leaves	31
	Minimum data in leaf	20

The MLP provides the ability to model complex nonlinear relationships in network traffic, allowing the system to detect subtle patterns associated with Mirai-based DDoS attacks. In contrast, the SVM with linear kernel offers a simple, interpretable approach with low computational overhead, ensuring efficient real-time inference on embedded platforms. Finally, LightGBM, a gradient boosting framework, combines high predictive accuracy with fast training and inference, and can handle large volumes of flow-based network features while remaining lightweight enough for low-power devices such as the Jetson Orin Nano.

By incorporating these three distinct approaches, neural networks, SVM with linear kernel, and tree-based boosting, the system achieves a balance between expressiveness, efficiency, and interpretability. This diversity allows for robust detection of malicious traffic while ensuring practical feasibility for real-time edge deployment. The following subsections describe each model in detail, including theoretical foundations, implementation choices, and experimental performance results.

To further evaluate the robustness and generalization capability of the proposed models, stratified 5-fold cross-validation was performed on the balanced dataset. Stratification ensured that each fold preserved the benign-to-malicious traffic ratio during training and validation. The results indicate consistent performance across folds for all evaluated models, with low variation in accuracy and F1-score. In particular, the LightGBM classifier consistently achieved the highest performance across all folds, supporting the stability of the model and reducing the likelihood that the reported results are due to overfitting or a favorable train-test split.

5.4.1 Multilayer Perceptron (MLP)

The Multilayer Perceptron is a feedforward neural network capable of approximating complex nonlinear mappings between input features and target labels. It consists of multiple layers of interconnected neurons, where each neuron applies a nonlinear activation to a weighted sum of its inputs. This structure allows MLPs to capture intricate relationships in high-dimensional data, making them suitable for traffic classification where patterns may be subtle or distributed across features.

For layer l, the transformation is given by:

$$\mathbf{h}^{(l)} = \phi\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \tag{5}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases, and $\phi(\cdot)$ is a nonlinear activation function such as ReLU or sigmoid. The network is trained using cross-entropy loss and optimized with the Adam algorithm, including regularization to reduce overfitting.

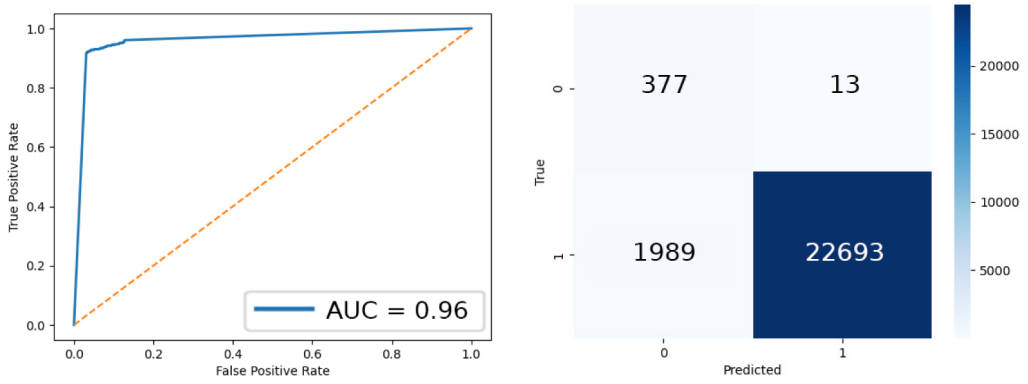
In this study, the MLP architecture comprises two hidden layers with 50 neurons each. It was trained for 500 iterations with a fixed random seed to ensure reproducibility. Table 4 summarizes the classification

results, demonstrating high precision and good overall accuracy, although slightly lower recall indicates some malicious flows were missed.

Table 4: MLP performance metrics.

Metric	Value
Accuracy	0.92
Precision	1.00
Recall	0.92
F1-score	0.96
AUC	0.96

The confusion matrix and ROC curve for the MLP show that while the model correctly classifies most benign and malicious flows, a small number of malicious flows are misclassified as benign (1989 false negatives), leading to a slightly reduced recall (see Fig. 4). The ROC AUC of 0.96 indicates strong overall discrimination between classes.



a ROC Curve **b** Confusion Matrix

Figure 4: MLP classifier evaluation: (**a**) ROC curve, (**b**) confusion matrix.

For the MLP model, a set of candidate architectures was explored by varying both the depth and width of the network, including different combinations of hidden layer sizes. The final configuration, consisting of two hidden layers with 50 neurons each, was selected based on its ability to achieve stable convergence and strong generalization without incurring excessive computational cost. The hidden layers employ the ReLU activation function, which improves gradient propagation and accelerates convergence compared to saturating nonlinearities. The model was trained using the Adam optimizer, which provides adaptive learning rate adjustment through first- and second-order moment estimates, contributing to efficient and stable optimization. Early stopping was enabled with a validation fraction of 0.1 to monitor generalization performance and prevent overfitting, halting training when no improvement was observed on the validation set. The maximum number of iterations was set to 500, which was empirically sufficient for convergence across multiple runs. This configuration ensures an appropriate balance between representational capacity and training efficiency, allowing the model to capture nonlinear relationships in the data while maintaining robustness and reproducibility.

5.4.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM) with a linear kernel is a discriminative classifier that seeks an optimal hyperplane separating benign and malicious flows with maximum margin. This margin maximization promotes robust generalization to unseen data, while the use of a linear kernel yields a simple and interpretable decision boundary, remaining computationally efficient and well suited for edge deployment.

Given training samples $\{(\mathbf{x}_i, y_i)\}$ with $y_i \in \{0,1\}$, mapped to $y_i' = 2y_i - 1$, the optimization problem is formulated as:

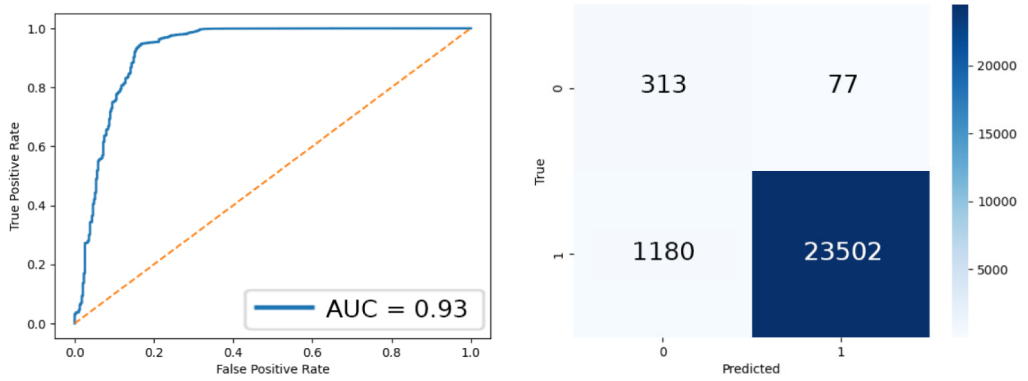
$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i'(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i \end{aligned} \tag{6}$$

To handle non-linearly separable data, a soft-margin formulation with slack variables and regularization parameter C is employed, allowing controlled misclassification while preserving margin maximization. Table 5 summarizes the experimental results obtained using the SVM with a linear kernel.

Table 5: SVM with linear kernel performance metrics.

Metric	Value
Accuracy	0.95
Precision	1.00
Recall	0.95
F1-score	0.97
AUC	0.93

The confusion matrix and ROC curve reveal that the SVM with a linear kernel correctly identifies most flows but misclassifies a higher number of malicious flows as benign (1180 false negatives), reducing recall compared to LightGBM (see Fig. 5). The ROC AUC of 0.93 indicates good discrimination capability, although it remains lower than that achieved by the MLP and LightGBM models.



a ROC Curve **b** Confusion Matrix

Figure 5: SVM with linear kernel classifier evaluation: (**a**) ROC curve, (**b**) confusion matrix.

For the SVM model, multiple kernel functions were systematically evaluated, including linear, radial basis function (RBF), polynomial, and sigmoid kernels, to assess the impact of nonlinear mappings on

the separability of the feature space. The experimental results showed that the linear kernel consistently achieved the best performance in terms of both classification metrics and computational efficiency. This suggests that the extracted flow-based features provide a sufficiently discriminative representation, enabling effective separation without requiring higher-dimensional transformations. The regularization parameter C was maintained at its default value to balance margin maximization and misclassification tolerance without introducing additional tuning complexity. Feature scaling was applied prior to training using standard normalization, which is critical for SVM stability and ensures that all features contribute proportionally to the optimization process. Probability estimates were enabled to allow consistent evaluation using ROC curves and AUC metrics. This configuration was selected as it minimizes computational overhead while preserving strong generalization capability, making it particularly suitable for real-time edge deployment.

5.4.3 LightGBM

LightGBM is a gradient boosting framework that iteratively builds an ensemble of decision trees to minimize prediction errors. By fitting each new tree to the negative gradient of the loss function, the model progressively improves its accuracy while remaining computationally efficient. LightGBM further optimizes performance using leaf-wise tree growth, histogram-based feature discretization, Gradient-based One-Side Sampling (GOSS), and Exclusive Feature Bundling (EFB).

The update at iteration m for sample i is defined as:

$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta h_m(\mathbf{x}_i), \tag{7}$$

where h_m is the m-th weak learner and η is the learning rate, with residuals computed as:

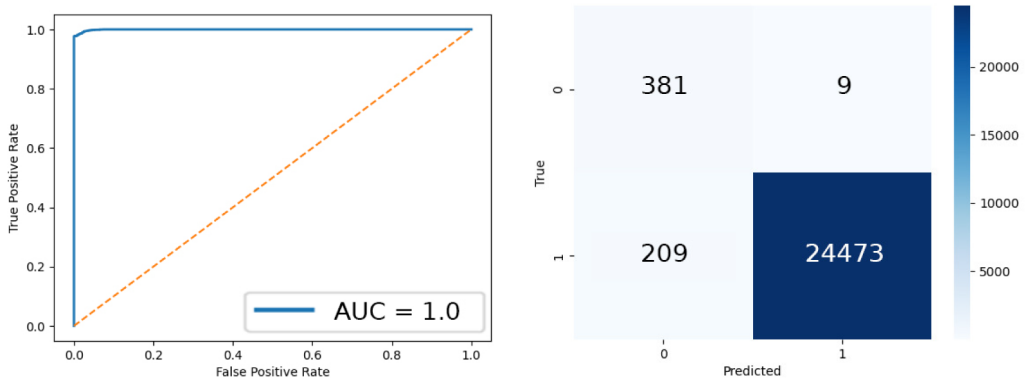
$$g_i^{(m)} = -\left.\frac{\partial \mathcal{L}(y_i, \hat{y}_i)}{\partial \hat{y}_i}\right|_{\hat{y}_i = \hat{y}_i^{(m-1)}}. \tag{8}$$

Table 6 presents the experimental results.

Table 6: LightGBM performance metrics.

Metric	Value
Accuracy	0.99
Precision	1.00
Recall	0.99
F1-score	1.00
AUC	1.00

The confusion matrix and ROC curve for LightGBM show near-perfect performance (see Fig. 6). Only 209 malicious flows were misclassified as benign, and the ROC AUC is 1.0, indicating excellent discrimination between benign and malicious traffic.



a ROC Curve **b** Confusion Matrix

Figure 6: LightGBM classifier evaluation: (**a**) ROC curve, (**b**) confusion matrix.

For the LightGBM model, a standard configuration was adopted with the binary classification objective and logloss evaluation metric, together with a fixed random state to ensure full reproducibility. Default hyperparameters were initially retained, including the learning rate, number of boosting iterations, maximum tree depth, and number of leaves, as preliminary experiments showed that LightGBM achieves near-optimal performance on the considered dataset without requiring extensive hyperparameter tuning. This behavior is consistent with the efficiency of histogram-based gradient boosting and leaf-wise tree growth, which enable the model to capture complex nonlinear feature interactions with relatively low computational cost. In addition, LightGBM inherently incorporates regularization mechanisms such as shrinkage, feature subsampling, and leaf-wise constraints, which help control model complexity and mitigate overfitting even under default settings. Avoiding aggressive hyperparameter optimization also aligns with the objective of maintaining a lightweight and reproducible solution suitable for edge deployment, where model stability and low computational overhead are critical. The selected configuration therefore provides an effective balance between predictive performance, training efficiency, and deployment feasibility in real-time intrusion detection scenarios.

The experimental results illustrate clear differences in the capabilities of the three models. MLP offers flexibility in modeling complex nonlinear patterns but suffers from higher false negatives, affecting recall. Linear SVM provides a lightweight and interpretable approach with better recall than MLP, though still misclassifying a notable number of malicious flows. LightGBM outperforms both, combining high accuracy, near-perfect recall, and low computational overhead, making it particularly suitable for real-time edge deployment in IoT networks. These observations highlight that while different classifiers have unique strengths, gradient boosting with LightGBM provides the most balanced and reliable solution for flow-based Mirai DDoS detection.

5.5 Reproducibility

Model training and development were performed using Python 3.12.3 within Jupyter Notebook 7.2.2 on a Windows 11 workstation equipped with 32 GB of RAM, an Intel i7 processor, and Intel UHD Graphics 620 with 128 MB of dedicated VRAM. Deployment and inference were carried out on an NVIDIA Jetson Orin Nano embedded edge platform.

The implementation relies on widely used open-source libraries, whose exact versions are reported to ensure full reproducibility: Scikit-learn 1.5.1, LightGBM 4.6.0, Imbalanced-learn 0.12.3, Matplotlib 3.9.2, and Scapy 2.7.0.

A fixed random seed of 42 was used throughout all preprocessing and training stages, including dataset splitting, stratified sampling, SMOTE oversampling, random undersampling, and model initialization, to ensure deterministic and repeatable results. The dataset was divided into training and testing subsets using a stratified 70%/30% split, preserving the class distribution in both sets. Additionally, stratified 5-fold cross-validation was applied to evaluate model stability and generalization performance.

After training, the final model was serialized in pickle format and deployed on the NVIDIA Jetson Orin Nano edge device for real-time inference. The deployment platform features a 6-core ARM CPU and 8 GB of LPDDR5 memory, operating under a low-power configuration. All inference experiments were executed in CPU-only mode to reflect realistic edge constraints without GPU acceleration.

5.6 Real-Time Flow-Based IDS

The performance of the proposed intrusion detection system was evaluated using the experimental dataset, comparing three machine learning models: MLP, Linear SVM, and LightGBM. Table 7 presents the evaluation metrics, including accuracy, precision, recall, F1-score, and AUC. Among the models tested, LightGBM achieves the highest performance, with an accuracy of 0.99, perfect precision of 1.00, a recall of 0.99, an F1-score of 1.00, and an AUC of 1.00. Linear SVM and MLP also achieve perfect precision, but their lower recall and F1-scores indicate that some malicious flows may not be detected, highlighting the advantage of LightGBM in reliably identifying both benign and malicious traffic.

Table 7: Performance metrics of evaluated machine learning models.

Model	Accuracy	Precision	Recall	F1-score	AUC
LightGBM	0.99	1.00	0.99	1.00	1.00
SVM	0.95	1.00	0.95	0.97	0.93
MLP	0.92	1.00	0.92	0.96	0.96

In addition to the proposed LightGBM model, two baseline classifiers were included for comparison: a Multilayer Perceptron (MLP) and a linear Support Vector Machine (SVM). These baselines were selected to represent both neural network-based and linear discriminative approaches, enabling a comprehensive evaluation of the proposed method against widely used machine learning techniques in intrusion detection.

The deployment viability of the proposed intrusion detection approach was examined by implementing the LightGBM classifier on the NVIDIA Jetson Orin Nano and evaluating its behavior under real-time traffic monitoring conditions. The assessment focused on practical execution aspects relevant to edge-based operation, including flow processing capacity, inference delay, processor load, memory usage, and energy consumption. The evaluation was conducted using the Python and Scapy-based implementation described earlier, with all classification tasks executed on the CPU to emulate realistic embedded deployment constraints.

The overall operation of the proposed intrusion detection system is formally described in Algorithm 1. The algorithm outlines the logical sequence executed during real-time operation, beginning with continuous packet capture via Scapy, followed by aggregation into unidirectional flows representing traffic directed toward the monitored device. For each completed or timed-out flow, representative statistical and temporal features are extracted and organized into a feature vector, which is then normalized and evaluated by a pre-trained supervised classifier. Based on this evaluation, each flow is labeled as benign or malicious, enabling prompt alert generation in the presence of Mirai-based DDoS activity while maintaining low latency and computational overhead on the embedded platform.

Algorithm 1: Real-time flow-based DDoS detection using LightGBM

Require: Incoming network traffic stream
Ensure: Flow label $\in \{\text{Benign}, \text{DDoS}\}$ and alert status
Define flow timeout $\tau_f = 60$ s
Define decision threshold $\theta = 0.85$
Load pre-trained LightGBM classifier
Continuously capture packets from the network
for all flow f such that f completes or inactivity exceeds τ_f **do**
    Aggregate packets into flow f
    Extract statistical and temporal features immediately after flow termination
    Construct feature vector $\mathbf{x}_f$
    Normalize $\mathbf{x}_f$
    Evaluate $\mathbf{x}_f$ using the trained classifier to obtain class probability vector $\mathbf{p}$
    Let p_{DDoS} be the probability associated with the DDoS class
    Determine predicted label $\hat{y} \leftarrow \arg\max(\mathbf{p})$
    if $\hat{y}$ = DDoS **and** $p_{DDoS} \geq \theta$ **then**
        Generate high-confidence alert
        Flag flow for mitigation
    else if $\hat{y}$ = DDoS **and** $p_{DDoS} < \theta$ **then**
        Generate low-confidence warning alert
    else
        Label flow as benign
    end if
end for
return Real-time classification results and generated alerts

To explicitly validate the lightweight characteristics of the proposed intrusion detection system, a quantitative analysis of computational efficiency and resource utilization is conducted. In this work, a model is considered lightweight if it satisfies key constraints in terms of computational complexity, memory footprint, inference latency, and energy consumption, enabling real-time execution on resource-constrained edge devices.

The proposed system was deployed and evaluated on the NVIDIA Jetson Orin Nano, and several runtime performance metrics were collected during continuous operation. The system maintains an average CPU utilization of around 15%. Memory consumption remains stable at approximately 200 MB, and the average power consumption of 7 W, which is consistent with the low-power design profile of the Jetson Orin Nano platform. The average inference latency is approximately 0.80 ms per 1000 flows, measured after flow aggregation and feature extraction. A comparison with real-time intrusion detection systems from the literature further contextualizes the performance of the proposed approach. Reported inference latencies for machine learning-based IDS models on edge platforms typically fall within the sub-millisecond range per 1000 flows, depending on model complexity and hardware configuration [31]. These observations indicate that the proposed system operates within a similar latency range while maintaining efficient resource utilization on an embedded edge platform.

These results indicate that the system operates efficiently within the limited computational and energy budgets typical of embedded edge environments. In particular, the relatively low CPU utilization and

moderate memory footprint demonstrate that the model does not impose significant resource demands on the device.

From a model perspective, the selected LightGBM classifier contributes significantly to the lightweight design. Unlike deep neural network-based intrusion detection systems, which typically involve millions of parameters and require GPU acceleration, the LightGBM model relies on an ensemble of shallow decision trees, where inference consists of a sequence of simple threshold comparisons. This results in low computational complexity and fast execution.

Furthermore, compared to the MLP and SVM models evaluated in this study, LightGBM provides a more favorable trade-off between detection performance and computational cost, achieving near-perfect classification results while maintaining efficient runtime behavior.

The combination of low resource utilization, energy efficiency, and real-time processing capability indicates that the proposed framework meets the requirements of a lightweight intrusion detection system suitable for deployment in IoT edge environments. In contrast to deep learning-based intrusion detection systems, which typically involve millions of parameters and require significantly higher memory and computational resources, often relying on GPU acceleration, the proposed approach remains suitable for constrained edge scenarios where efficiency is critical.

6 Results and Discussion

The superior performance of LightGBM can be attributed to its gradient boosting framework and optimized tree construction strategies, which allow it to efficiently learn complex patterns in high-dimensional traffic data. Its ensemble of weak learners effectively captures both volumetric and temporal characteristics of network flows, enabling strong discrimination capability between benign and malicious traffic. In contrast, Linear SVM, while maintaining high precision, exhibits lower recall and AUC values, indicating a reduced ability to capture all attack patterns. The MLP shows comparatively lower performance, likely due to sensitivity to hyperparameter tuning and the computational complexity of training neural networks on the available dataset.

It is important to contextualize these results with respect to the characteristics of the experimental setup. In particular, the use of a balanced dataset during training and evaluation may influence the learning dynamics by reducing the effects of class imbalance, which is commonly observed in real-world network traffic scenarios. This can result in more stable classification metrics across all evaluated models under the experimental conditions.

To provide additional insight into the design choices of the proposed system, a controlled ablation-style analysis was conducted during the development phase to assess the impact of feature representation and flow construction strategy on model performance. The results indicate that the use of flow-based statistical and temporal features is essential for achieving stable classification performance, as configurations that removed or reduced temporal aggregation information consistently led to a noticeable degradation in discrimination capability between benign and malicious traffic. Similarly, different flow construction settings, particularly variations in timeout and aggregation window size, were observed to affect the trade-off between temporal resolution and robustness. Shorter windows limited the ability to capture meaningful traffic dynamics, while excessively large windows introduced aggregation noise and reduced responsiveness. The selected configuration was therefore chosen based on its consistent behaviour across preliminary evaluations, providing a balanced compromise between detection stability and responsiveness in real-time scenarios.

In addition, the characteristics of the considered attack traffic may also contribute to the observed performance. Since the dataset focuses on Mirai-based DDoS attacks, the generated traffic exhibits consistent flow-level behavior, which can facilitate the separability between benign and malicious classes when using statistical and temporal features. This behavior may be particularly beneficial for models capable of capturing nonlinear relationships, such as ensemble-based methods including LightGBM, under the evaluated conditions.

This flow-based design also introduces an inherent trade-off between flow completion and early detection, which is a fundamental aspect of real-time intrusion detection systems operating at the flow level.

Furthermore, the performance of LightGBM can also be associated with the separability of the extracted feature space. The flow-level statistical and temporal features provide a representation in which benign and malicious traffic exhibit distinguishable patterns under the evaluated experimental conditions. This characteristic supports the effectiveness of classifiers capable of learning nonlinear decision boundaries, such as gradient boosting models.

In addition to classification performance, computational efficiency is a critical factor for real-time deployment in resource-constrained edge environments. LightGBM demonstrates a favorable balance of high accuracy and low computational overhead, requiring less memory and shorter inference times than neural networks such as the MLP. Linear SVM is lightweight and interpretable, but its lower recall may limit detection reliability in operational scenarios. Deploying LightGBM on the NVIDIA Jetson Orin Nano enables continuous monitoring of network flows, allowing timely detection and mitigation of Mirai-based DDoS attacks before they impact IoT devices. This edge-based deployment ensures low latency, reduced communication overhead, and preservation of device availability, which are essential for maintaining service continuity in IoT networks.

The results confirm that flow-based feature extraction combined with a robust and computationally efficient machine learning model provides an effective solution for real-time intrusion detection in IoT environments. LightGBM achieves high detection performance, minimizing both false positives and false negatives, while also meeting the practical constraints of embedded deployment. While neural networks offer flexibility for complex modeling and Linear SVMs provide interpretability and efficiency, the combination of high detection performance, computational efficiency, and edge deployment capability makes LightGBM particularly suitable for protecting IoT devices against distributed Mirai-based DDoS attacks.

Although the dataset subset used in this study is representative and widely adopted in related research, its controlled nature may influence the distribution of traffic patterns observed during training and evaluation. This should be considered when interpreting the generalization capability of the proposed approach in real-world IoT environments, where traffic distributions and attack behaviors can be more heterogeneous.

These considerations provide context for interpreting the reported results and delineate the scope of applicability of the proposed approach in practical IoT environments.

A benchmarking study [32] on machine learning-based intrusion detection systems for IoT edge devices highlights that performance reported in controlled or simulation-based environments does not always translate directly to real-world deployment conditions. The study emphasizes that computational constraints of edge hardware significantly affect the feasibility of ML-based IDS solutions, making evaluation on real devices an essential requirement for obtaining reliable and reproducible results. In addition, it reports an inherent trade-off between detection performance and resource efficiency, where high-performing models may incur increased latency, energy consumption, or reduced scalability when deployed on resource-constrained platforms.

In this context, the proposed system directly addresses these challenges by evaluating performance on the NVIDIA Jetson Orin Nano under realistic edge computing conditions. The LightGBM-based model achieves strong classification performance while maintaining low computational overhead in terms of CPU utilization, memory consumption, and power usage, demonstrating a practical balance between detection capability and deployment efficiency in embedded IoT environments.

7 Conclusions

This work presents a flow-based intrusion detection system designed for real-time deployment on the NVIDIA Jetson Orin Nano to protect IoT networks against Mirai-based DDoS attacks. By combining lightweight and high-performance machine learning techniques with edge-based execution, the proposed solution enables timely detection of malicious traffic while maintaining low computational overhead and preserving the availability of constrained IoT devices.

Experimental results show that, among the evaluated classifiers, the gradient boosting approach achieves the best overall detection performance, reaching near-perfect accuracy, precision, recall, F1-score, and AUC. These results confirm its effectiveness in discriminating between benign and malicious network flows while satisfying the strict latency and resource constraints imposed by embedded edge platforms. Furthermore, the adoption of flow-based feature extraction allows the system to capture both volumetric and temporal traffic characteristics, improving robustness against diverse Mirai-driven DDoS attack patterns.

From a practical deployment perspective, implementing the intrusion detection system directly at the network edge enables real-time monitoring and mitigation with reduced communication overhead and faster response times compared to centralized approaches. This is particularly relevant for IoT infrastructures, where latency, bandwidth efficiency, and scalability are critical. The integration of detection and mitigation mechanisms within the same edge device further supports autonomous operation and facilitates deployment in distributed and resource-constrained environments, demonstrating the feasibility of the proposed approach in realistic scenarios.

While the experimental setup focuses on Mirai-based DDoS traffic and a representative subset of IoT network data, the underlying flow-based design and lightweight modeling approach provide a flexible foundation that can be extended to a broader range of attack patterns and deployment conditions. The use of flow-level features enables efficient processing and scalability, although incorporating additional traffic representations may further enhance detection capabilities in more complex scenarios.

Future work will build upon these results by exploring more adaptive and collaborative detection strategies. In particular, federated learning approaches will be investigated to enable distributed model training across multiple edge devices while preserving data privacy. Enhancing adversarial robustness will also be considered to improve resilience against evolving evasion techniques. Additionally, multi-edge collaborative detection architectures will be studied to support coordinated threat intelligence sharing and scalable protection across large-scale IoT environments. These directions aim to further strengthen the applicability, robustness, and scalability of edge-based intrusion detection systems.

Acknowledgement: This research has been possible thanks to support from the SCITALA C064/23 project, the Cybersecurity Chair at the University of La Laguna, INCIBE, and the European Union NextGenerationEU/PRTR.

Funding Statement: This research was supported by the Recovery, Transformation and Resilience Plan, financed by the European Union (NextGenerationEU) under the strategic project C064/23 SCITALA (INCIBE-ULL).

Author Contributions: The authors confirm contribution to the paper as follows. Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation,

writing—review and editing, visualization, supervision, project administration: Jackson Diaz-Gorrin, Candido Caballero-Gil, Pino Caballero-Gil and Joanna Kolodziej. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

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

References

1. Yalli JS, Hasan MH, Jung LT, Yerima AI, Aliyu DA, Maiwada UD, et al. A systematic review for evaluating IoT security: a focus on authentication, protocols and enabling technologies. IEEE Internet Things J. 2025;12(12):18908–28. doi:10.1109/JIOT.2025.3545737.
2. Aarti, Gowroju S, Ansarullah SI. A comprehensive review of developments and challenges in the 6G internet of things. Discov Netw. 2025;1(1):13. doi:10.1007/s44354-025-00013-y.
3. Mandela N, Etyang F. Comparative analysis of deep learning models for effective denial of service (DoS) attack detection in network security. J Electr Syst Inf Technol. 2025;12(1):73. doi:10.1186/s43067-025-00267-0.
4. Hirsi A, Alhartomi MA, Audah L, Salh A, Sahar NM, Ahmed S, et al. Comprehensive analysis of DDoS anomaly detection in software-defined networks. IEEE Access. 2025;13:23013–71. doi:10.1109/ACCESS.2025.3535943.
5. Kumar S, Dwivedi M, Kumar M, Gill SS. A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services. Comput Sci Rev. 2024;53(19):100661. doi:10.1016/j.cosrev.2024.100661.
6. Kaur N. A systematic review on security aspects of fog computing environment: challenges, solutions and future directions. Comput Sci Rev. 2024;54(2):100688. doi:10.1016/j.cosrev.2024.100688.
7. Pimentel EL, de Souza CA, Westphall CB. Detecting attacks in fog and cloud computing environments using deep learning: a systematic literature review. Comput Netw. 2025;264:111269. doi:10.1016/j.comnet.2025.111269.
8. Luo Z. Fog node intrusion detection and response based on SVMIF and INSGA-II algorithm. Syst Softw Comput. 2025;7(8):200330. doi:10.1016/j.sasc.2025.200330.
9. Zhang H. IoT botnet forensics: a comprehensive digital forensic case study on Mirai botnet servers. Forensic Sci Int Digit Investig. 2020;32:300926. doi:10.1016/j.fsidi.2020.300926.
10. Liu S, Fauzi F, Kok VJ. A systematic literature review: classifying IoT botnet data features based on its lifecycle. IEEE Access. 2025;13(12):65040–65. doi:10.1109/ACCESS.2025.3559635.
11. Harada R, Shibata N, Kaneko S, Honda K, Terada J, Ishida Y, et al. Quick suppression of DDoS attacks by frame priority control in IoT backhaul with construction of Mirai-based attacks. IEEE Access. 2022;10:22392–9. doi:10.1109/ACCESS.2022.3153067.
12. Abdulganiyu OH, Ait Tchakoucht T, Saheed YK. A systematic literature review for network intrusion detection system (IDS). Int J Inf Secur. 2023;22(5):1125–62. doi:10.1007/s10207-023-00682-2.
13. Samita. A review on intrusion detection system for IoT based systems. SN Comput Sci. 2024;5(4):380. doi:10.1007/s42979-024-02702-x.
14. Aydin B, Aydin H, Gormus S. Intrusion detection systems in IoT: a detailed review of threat categories, detection strategies, and future technologies. J Inf Secur Appl. 2025;95(4):104291. doi:10.1016/j.jisa.2025.104291.
15. Hua H, Li Y, Wang T, Dong N, Li W, Cao J. Edge computing with artificial intelligence: a machine learning perspective. ACM Comput Surv. 2023;55(9):184. doi:10.1145/3555802.
16. Uddin MR, Evan NA, Alam MR, Arefin MT. Analysis of generic routing encapsulation (GRE) over IP security (IPSec) VPN tunneling in IPv6 network. In: Kumar N, Vinodhini M, Venkatesha Prasad RR, editors. Ubiquitous communications and network computing. Cham, Switzerland: Springer; 2021. p. 3–15. doi:10.1007/978-3-030-79276-3_1.
17. Saleh MA, Abdul Manaf A. Optimal specifications for a protective framework against HTTP-based DoS and DDoS attacks. In: Proceedings of the 2014 International Symposium on Biometrics and Security Technologies (ISBAST); 2014 Aug 26–27; Kuala Lumpur, Malaysia. p. 263–7. doi:10.1109/ISBAST.2014.7013132.

18. Sagatov E, Mayhoub S, Sukhov A, Calyam P. Countering DNS amplification attacks based on analysis of outgoing traffic. J Commun Inf Netw. 2023;8(2):111–21. doi:10.23919/JCIN.2023.10173727.
19. Romadhon FW, Nuha MAU, Adiprawira Y, Sari RF. Comparative analysis of HAProxy and Nginx load balancers in mitigating User Datagram Protocol (UDP) flood attacks. In: Proceedings of the 2024 12th International Conference on Information and Communication Technology (ICoICT); 2024 Aug 7–8; Bandung, Indonesia. p. 354–9. doi:10.1109/ICoICT61617.2024.10698656.
20. Desai A, Brahma S, Khare S. SYN flood attacks in wireless local area networks (WLANs): dataset and analysis. In: Proceedings of the 2025 IEEE 16th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON); 2025 Oct 22–24; Yorktown Heights, NY, USA. p. 79–85. doi:10.1109/UEMCON67449.2025.11267661.
21. Minott D, Siddiqui S, Haddad RJ. Benchmarking edge AI platforms: performance analysis of NVIDIA Jetson and Raspberry Pi 5 with Coral TPU. In: Proceedings of the SoutheastCon 2025; 2025 Mar 22–30; Concord, NC, USA. p. 1384–9. doi:10.1109/SoutheastCon56624.2025.10971592.
22. Aldallal IB, Ibrahim AA, Ahmed SR. An intelligent multi-stage GA-SVM hybrid optimization framework for feature engineering and intrusion detection in internet of things networks. Comput Mater Contin. 2026;87(1):39. doi:10.32604/cmc.2025.075212.
23. Affinito A, Zinno S, Stanco G, Botta A, Ventre G. The evolution of Mirai botnet scans over a six-year period. J Inf Secur Appl. 2023;79(2):103629. doi:10.1016/j.jisa.2023.103629.
24. Asadi M, Heidari A, Jafari Navimipour N. A new flow-based approach for enhancing botnet detection efficiency using convolutional neural networks and long short-term memory. Knowl Inf Syst. 2025;67(7):6139–70. doi:10.1007/s10115-025-02410-9.
25. Zhang H. Development of an intelligent intrusion detection system for IoT networks using deep learning. Discov Internet Things. 2025;5(1):74. doi:10.1007/s43926-025-00177-7.
26. Thomas T, Prakash R, Pal S. Intrusion detection in internet of medical things using digital twins—a review. Comput Mater Contin. 2025;84(3):4055–104. doi:10.32604/cmc.2025.064903.
27. Sharma SB, Bairwa AK. Leveraging AI for intrusion detection in IoT ecosystems: a comprehensive study. IEEE Access. 2025;13(6):66290–317. doi:10.1109/ACCESS.2025.3550392.
28. Chen C, Ye D, He Y, Tang L, Xu Y. Improving adversarial robustness with adversarial augmentations. IEEE Internet Things J. 2024;11(3):5105–17. doi:10.1109/JIOT.2023.3301608.
29. Wu Y, Liu L, Yu Y, Chen G, Hu J. Online ensemble learning-based anomaly detection for IoT systems. Appl Soft Comput. 2025;173(10):112931. doi:10.1016/j.asoc.2025.112931.
30. Herzalla D, Lunardi WT, Andreoni M. TII-SSRC-23 dataset: typological exploration of diverse traffic patterns for intrusion detection. IEEE Access. 2023;11(6):118577–94. doi:10.1109/ACCESS.2023.3319213.
31. Nandihal P, Ashirova A, Sultanov R, Raghava MS, Begaliev A, Ruzieva M. Lightweight dual-stream IDS: feature-pruned CNN-LSTM with fast random-learning Aquila optimisation. In: Proceedings of the 2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3); 2025 Jul 25–26; Bhubaneswar, India. p. 1–6. doi:10.1109/ISAC364032.2025.11156547.
32. Alsharif M, Rawat DB. Benchmarking machine learning based intrusion detection for IoT edge devices. In: Proceedings of the 2024 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI); 2024 Dec 6–8; Kusatsu, Japan. p. 210–6. doi:10.1109/IIKI65561.2024.00044.