Lightweight Machine Learning-Based Intrusion Detection System for IoT Networks

Tanush Gupta Guntha[1], M Reddy Rohith[2], Rishabh Natarajan[3], Dr. R. Kayalvizhi[4]
[1,2,3,4]*Dept. of Networking and Communication,*
SRM Institute of Science and Technology Chennai, India

***Abstract*—The increased usage of Internet of Things (IoT) devices in recent times has resulted in the increase in threat of cyber-attacks like Denial of Service (DoS) which are used to exploit resource constrained devices. Traditional Intrusion Detection Systems (IDS) use models having high computational complexity which makes them unsuitable for use in lightweight IoT systems. In this project, we are working on a lightweight machine learning based DoS detection model which combines efficient feature engineering, a gradient boosting classifier and a real time monitoring interface. It utilizes the UNSW-NB15 dataset to train the model and focuses on behaviour driven feature selection to identify the indicators of DoS like traffic flooding. LightGBM algorithm is used for the ML model because of its low overhead and fast inference capabilities. A pipeline is developed using Pyshark for automated packet parsing and sliding window traffic aggregation. The extracted features are evaluated by the trained model in order to detect patterns of DoS. In addition to detection, there is a web interface in order to provide live visualization of attack probability, inference latency, and traffic status. Experimental results show high detection accuracy with low inference time and low memory usage which makes it suitable for edge level deployment. The compact model size and CPU only execution mode makes it ideal for IoT environments where hardware resources are limited. This approach helps bridge the gap between high accuracy intrusion detection systems and practical real time deployment by offering a compact, fast and efficient solution for DoS detection.**

***Index Terms*—IoT Security, Denial of Service Detection, Intrusion Detection System, LightGBM, Feature Engineering, Real-Time Monitoring**

I. INTRODUCTION

The rapid growth of Internet of Things (IoT) and technology and their application have enabled many devices to communicate and operate across domains like healthcare, smart home systems, automation and transport. At the same time, this has resulted in exposure to many security challenges in IoT devices which operate under strict constraints of processing power, memory and energy consumption. These make it difficult to implement traditional security measures, exposing IoT networks to a wide range of cyber threats. Among these cyber threats, Denial of Service (DoS) is one of the most prevalent and disruptive ones. It works by overwhelming a network through excessive traffic and connection requests which can degrade or even completely disrupt the functionality of the device. Hence, the need for a lightweight intrusion detection system (IDS) tailored for IoT devices arises.

Existing IDS tend to prioritize detection accuracy by using deep learning or complex ensemble models which prove to be effective in centralized computing environments requiring significant computational resources. This proves to be unsuitable for deployment in IoT edge settings which operate under resource constrained conditions. To address this challenge, this project proposes a lightweight machine learning–based DoS detection framework which is designed for real-time application. It does this by using carefully selected behavioural features which are corelated to behaviour exhibited during DoS attacks. A LightGBM classifier is used due to its efficiency and low latency which makes it suitable for IoT environments. The primary contributions of this project are the development of a lightweight DoS detection model using behaviour driven feature engineering, implementation of a real-time detection pipeline through automated packet parsing and Integration of a web-based monitoring interface for live attack visibility. By combining an efficient machine learning

model with real time monitoring capabilities, our aim is to provide a practical and scalable solution for enhancing IoT network security against DoS attacks.

II. LITERATURE SURVEY

The security of Internet of Things (IoT) networks is an active area of research due to the increasing susceptibility of resource constrained devices to cyberattacks such as Denial of Service (DoS). Traditional intrusion detection systems (IDS) rely on signature matching or rule-based detection and are often unsuitable for IoT deployments due to their high computational overhead.

Machine learning (ML) based IDS using behaviour based detection of malicious traffic appears to be a good solution. Classical supervised learning models like Decision Trees, Support Vector Machines (SVM), and Random Forests have been explored for network intrusion detection tasks. A study by Moustafa and Slay [1] demonstrated that statistical traffic features which were derived from datasets like UNSW-NB15 could significantly improve detection accuracy compared to traditional methods. Modern IoT-specific datasets further helped with the development of more realistic IDS models. Research utilizing the UNSW-NB15 dataset shows that ensemble-based classifiers can effectively detect network-layer attacks including DoS and probing activities [2]. Random Forest and boosting techniques provide improved classification performance due to their ability to capture nonlinear relationships in traffic patterns while maintaining their interpretability [3].

Due to advancements in computational capabilities, deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Autoencoders have also been applied to intrusion detection tasks [4]. These approaches demonstrated strong performance in identifying complex attack signatures by learning hierarchical representations of traffic data. However, their practical deployment in IoT environments is limited due to high memory consumption, training complexity, and reliance on GPU acceleration. To overcome these challenges, recent research has focused on lightweight machine learning techniques which balance accuracy and computational efficiency. For this, gradient boosting frameworks like XGBoost and LightGBM have gained attention due to their fast-training speed and low inference latency [5]. Ke et al. [6] introduced LightGBM as a scalable gradient boosting decision tree framework capable of handling large-scale data with reduced computational requirements, which makes it suitable for edge-level deployment scenarios.

Another critical component in reducing computational complexity without sacrificing detection performance is Feature selection. Several studies have demonstrated that behaviour driven feature reduction can help improve IDS efficiency while also preserving the most discriminative traffic characteristics [7]. Despite these advancements, many existing solutions remain confined to testing on static datasets and do not address real-time operational deployment or live monitoring. This highlights a research gap in designing IDS frameworks which are not only accurate but also deployable in real-time IoT environments. This project attempts to address this gap by combining behaviour-based feature engineering, a lightweight gradient boosting model, automated packet parsing using PyShark, and a web-based monitoring interface to enable real time DoS detection and be suitable for constrained edge deployments.

III. SYSTEM OVERVIEW

The proposed framework is designed to provide a lightweight, real time detection mechanism for Denial of Service (DoS) attacks in IoT network environments. Unlike conventional intrusion detection systems which rely on complex and computationally intensive models or offline traffic analysis, this system integrates machine learning based detection with automated packet monitoring and live visualization to improve applicability. Fig 3.1 below depicts the architecture of the system:

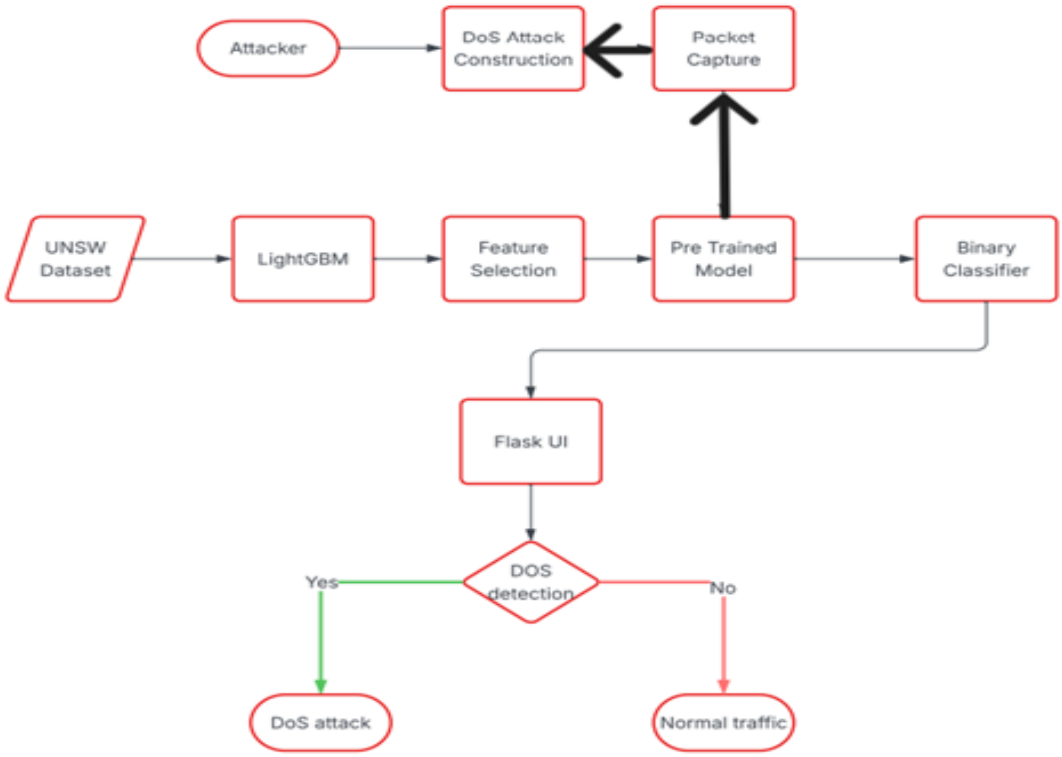

Fig 3.1 High level architecture diagram

During the training phase, network traffic data obtained from the UNSW-NB15 dataset is processed to identify behavioural patterns associated with both normal and DoS traffic. A reduced set of 11 statistically significant features is selected to minimize computational complexity while preserving indicators of DoS attack behaviour. These selected features are then used to train a LightGBM classifier in order to distinguish DoS attack and normal network traffic. After training, the model is serialized and saved in the form of a pkl file.

In the real time phase, live network traffic is continuously captured and processed through automated packet parsing done using PyShark. The incoming packets are filtered and aggregated within short time intervals in order to capture sudden traffic bursts and abnormal connection behaviour which is typically associated with DoS attacks. From the aggregated traffic stream, the same behavioural features used during training are extracted. The resulting feature vector is evaluated using the trained LightGBM model to determine the likelihood of DoS attack. The prediction output is then visualized and displayed on a web-based monitoring interface which has graphs depicting the network conditions, attack probability, and system response latency. By combining efficient feature engineering, automated packet analysis, and a deployable machine learning model, the proposed system establishes a practical pipeline that transitions intrusion detection from static dataset evaluation to real time operational monitoring suitable for IoT environments.

IV. METHODOLOGY

A. Dataset And Attack Modelling

The UNSW-NB15 dataset provides a comprehensive representation of modern network traffic containing both normal and malicious traffic. It captures realistic traffic behaviour by incorporating a variety of attack scenarios and network protocols, which makes it suitable for evaluating intrusion detection systems in present day environments. For our study, was filtered the dataset to focus specifically on Normal and Denial of Service (DoS) traffic categories in order to detect mechanisms for traffic flooding attacks. DoS attacks were selected as the primary focus due to their prevalence and their severe impact on resource constrained IoT devices. These attacks usually aim to

overwhelm network services through excessive traffic or repeated connection attempts, which leads to service degradation or complete disruption. Within the dataset, DoS instances exhibit distinct behavioural characteristics such as elevated packet rates, abnormal traffic load, and reduced server responsiveness. These behavioural patterns serve as the basis for constructing a classification model which is capable of distinguishing between benign and malicious activity.

In real time traffic, DoS activity is commonly seen as rapid bursts of connection requests and minimal response acknowledgment from the target system. Therefore, the modelling approach emphasizes behavioural similarity rather than strict signature matching. This allows the trained system to generalize from dataset-based patterns to live traffic conditions.

B. Feature Engineering for Dos Detection

Detection of Denial-of-Service attacks in resource constrained environments requires the identification of traffic characteristics that are both discriminative and computationally lightweight. Instead of relying on a large number of network attributes, our work focuses on traffic intensity, connection dynamics, and response behaviour. These characteristics are essential in capturing the operational footprint of DoS attacks, which typically involve rapid request bursts and limited server acknowledgment. To identify the relevant attributes, statistical analysis was conducted to compare normal and DoS traffic instances within the dataset. Features exhibiting significant differences in their mean values and strong correlation with attack behaviour were selected for model training to ensure that the final feature set retained meaningful indicators of abnormal activity while also reducing dimensionality to support efficient real-time processing.

The selected features represent three behavioural aspects of network traffic: connection concentration, timing irregularities, and traffic load imbalance. Connection based attributes show repeated interaction with the same service or destination, which is a common pattern in flooding attacks. Timing related features capture disruptions in handshake responses that occur when servers become overwhelmed. Load related attributes represent the volume and rate of transmitted data, which typically increase during attack conditions.

To ensure consistency the same feature definitions were reconstructed during live packet analysis. Traffic captured in real time was aggregated over short time intervals to approximate the statistical representation observed in the dataset. This sliding window aggregation allowed the system to translate raw packet flows into structured behavioural indicators suitable for machine learning inference.

C. ML model design

The machine learning component of the IDS is designed to balance performance with computational efficiency. Instead of using deep neural architectures which demand more memory and processing power, this work uses a gradient boosting decision tree model implemented using LightGBM. The choice of LightGBM is due to its histogram-based tree learning algorithm, leaf wise growth strategy, and optimized memory usage, which enable high accuracy with low inference latency. These characteristics make it suitable for IoT oriented and edge constrained environments. The model operates as a supervised binary classifier distinguishing between Normal and DoS traffic. Training data were obtained from the UNSW-NB15. Only the eleven selected behavioural features were used as input variables to ensure dimensional consistency between training and real time inference stages. An 80/20 split was done for training and testing subsets to mitigate sampling bias and ensures reliable performance evaluation.

Class imbalance was addressed using balanced class weighting during training, ensuring that the minority DoS class contributed proportionally to the loss optimization process. The boosting framework iteratively constructs decision trees that correct errors made by preceding trees, thereby improving classification boundaries across the feature space. During training, the model learns non-linear relationships between behavioural features such as packet rate, service repetition counts, TCP round trip indicators, and traffic load statistics which enable the classifier to distinguish sudden traffic surges and abnormal connection dynamics from legitimate

activity. Unlike detection systems that rely on static thresholds, the boosting approach adapts decision boundaries based on data driven optimization, improving robustness against traffic variability.

After training, the optimized model was serialized into a compact binary file using the Joblib library. This serialized model serves as the inference engine within the real time detection pipeline. During deployment, extracted feature vectors are passed directly to the loaded LightGBM model, which outputs a probabilistic estimate of DoS likelihood. A threshold-based decision mechanism then converts this probability into a binary detection label. Because inference in gradient boosted trees involves traversing shallow decision paths rather than performing matrix multiplications across large neural layers, prediction latency remains within millisecond scale bounds.

D. Real-time detection pipeline

The real time detection pipeline is designed to perform operational traffic monitoring. Unlike traditional intrusion detection approaches which rely on manual packet inspection tools, this system employs automated traffic parsing using the PyShark library, which acts as a Python wrapper over the TShark packet analysis engine. This enables continuous capture and structured interpretation of live network packets without requiring manual intervention. Incoming traffic is filtered to focus on TCP based communication, as DoS attacks frequently occur through repeated connection initiation attempts and excessive service targeting. Captured packets are aggregated within a fixed sliding time window of one second to approximate burst characteristics associated with flooding attacks. This window-based aggregation ensures that traffic spikes are captured. Within each aggregation interval, statistical traffic metrics are computed to reconstruct the same behavioural feature vector used during the training phase. These include indicators of connection repetition, traffic rate amplification, byte load concentration, and imbalance between request and response packets. Maintaining identical feature definitions between training and deployment ensures distributional consistency and reduces inference drift.

Once the feature vector is constructed, it is passed to the pre-trained LightGBM model stored in serialized format. The model outputs a probability score representing the likelihood of DoS activity within the observed time window. A threshold-based decision mechanism converts this probability into a binary classification label. Because the inference process relies solely on CPU execution and a compact tree-based model, prediction latency remains minimal.

The output of the detection module is then transmitted to the web interface through a backend service layer built using Flask and SocketIO, enabling asynchronous communication between the detection engine and the monitoring dashboard. This architecture ensures that traffic capture, feature extraction, and inference operate continuously while allowing users to observe detection outcomes in real time.

E. Web-based monitoring interface

To bridge the gap between offline model evaluation and practical deployment, the intrusion detection framework integrates a real time web-based monitoring interface. The objective of this interface is not just visualization, but also operational validation of the complete detection pipeline under simulated attack and normal traffic conditions. The monitoring system is implemented using a Flask backend combined with Flask-SocketIO to enable bidirectional, low latency communication between the detection engine and the frontend dashboard. The backend operates as an event driven service layer coordinating three primary processes: traffic generation, packet aggregation and feature extraction, and model inference. Detection outputs are broadcast asynchronously via WebSocket events to the frontend, ensuring near real-time updates without page reloads.

The user interface is structured into multiple panels. The Attacker Dashboard allows controlled initiation and termination of DoS traffic, with adjustable parameters such as thread intensity and attack type. The Client Simulator panel generates normal HTTP traffic at configurable rates to emulate benign IoT communication. The Server Monitor panel displays live detection results, including attack status and probability confidence. Finally, the Analytics panel visualizes feature level values and packet statistics over sliding windows, enhancing interpretability.

Packet capture and parsing are automated using the PyShark library (TShark backend), replacing manual Wireshark-based inspection workflows. Live Capture functionality enables the backend to monitor loopback or selected interfaces continuously, aggregate packets into fixed duration windows, and compute the required feature vector in real time. The monitoring system also incorporates an Edge Device Simulation panel that estimates inference latency under constrained computational settings. By adjusting slowdown factors to approximate Raspberry Pi 5 characteristics, the interface provides real time feedback on predicted inference time, model size, and resource utilization. This component supports deployment feasibility analysis without requiring immediate hardware integration.

V. RESULTS

This section evaluates the DoS detection framework from both predictive and computational perspectives, including offline classification performance on benchmark data, real time validation using live traffic simulation, and deployment feasibility assessment under edge constrained conditions. It also emphasizes inference latency, model compactness, and operational responsiveness to ensure practical applicability.

The LightGBM classifier was trained using an 80/20 stratified split of the filtered UNSW-NB15 dataset containing Normal and DoS traffic samples. Stratification ensured proportional representation of both classes in training and testing partitions, preventing evaluation bias. On the test set, the model achieved an accuracy exceeding 97%, with precision and recall values above 96% for the DoS class. The false negative rate remained extremely low, indicating strong sensitivity toward attack behavior. The false positive rate was also minimal, demonstrating that the selected feature subset preserves discriminative capacity without causing excessive alert generation. These results validate that the compact 11-feature behavioral representation retains sufficient statistical divergence between normal and malicious traffic patterns. The 11 selected features and their feature importance is shown below, in Fig 5.1:

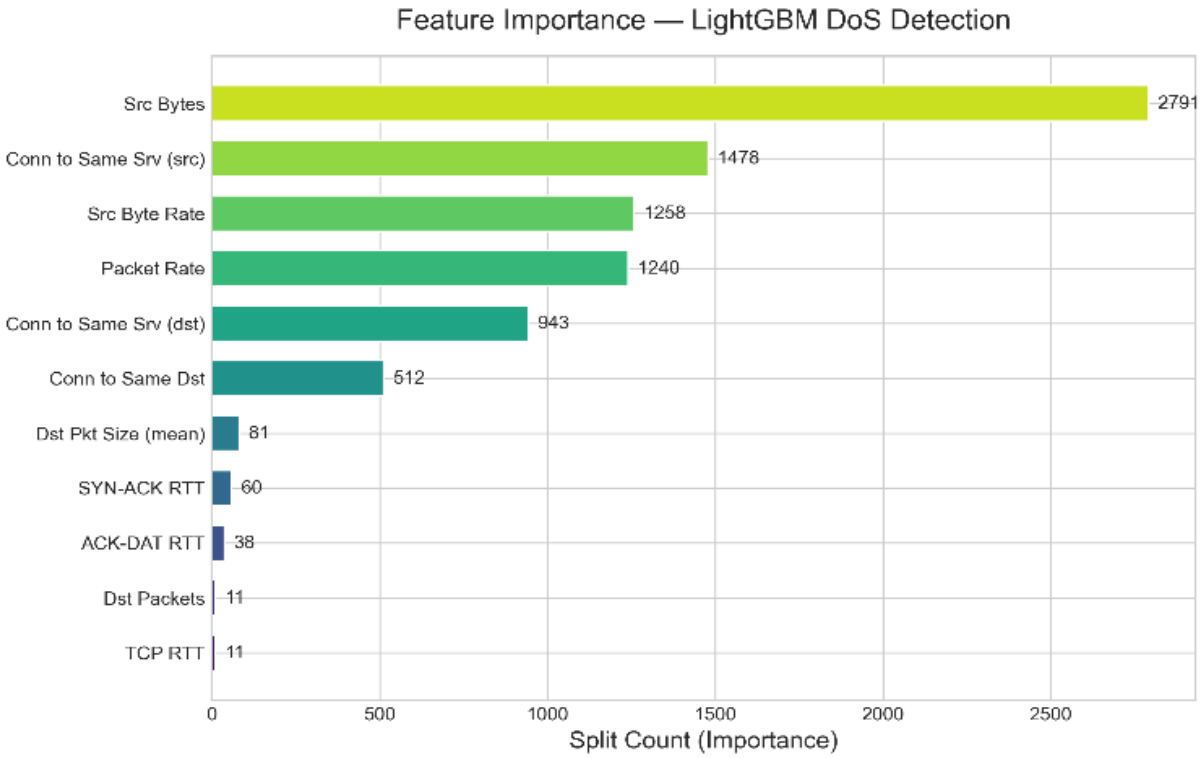

Fig 5.1 Feature importance

To evaluate generalization stability, stratified cross-validation was conducted across multiple folds. The model demonstrated consistently high performance with negligible variance in accuracy, precision, recall, and F1-score across folds. This stability indicates that the boosting architecture and selected features are not overly sensitive to dataset partitioning. The low standard deviation across validation rounds suggests robustness and reliable generalization capability within the defined binary classification scope.

Beyond offline evaluation, the trained model was integrated into the real-time detection pipeline implemented in the web-based monitoring interface. During controlled experiments, baseline HTTP traffic was first generated to emulate normal IoT communication as shown in Fig 5.2. Subsequently, a SYN flood-based DoS attack was initiated through the attacker simulation module as shown in Fig 5.3. Packet streams were aggregated into one-second sliding windows, and the 11 behavioural features were computed dynamically using PyShark based parsing. The LightGBM model processed each feature vector and returned a probabilistic output representing DoS likelihood. During attack onset, observable increases in packet rate, service target repetition counts, and TCP timing degradation were reflected in the extracted features. Correspondingly, the predicted DoS probability rose significantly and triggered detection alerts within a single aggregation window. This behaviour confirms that the framework achieves near real time responsiveness under simulated attack conditions.

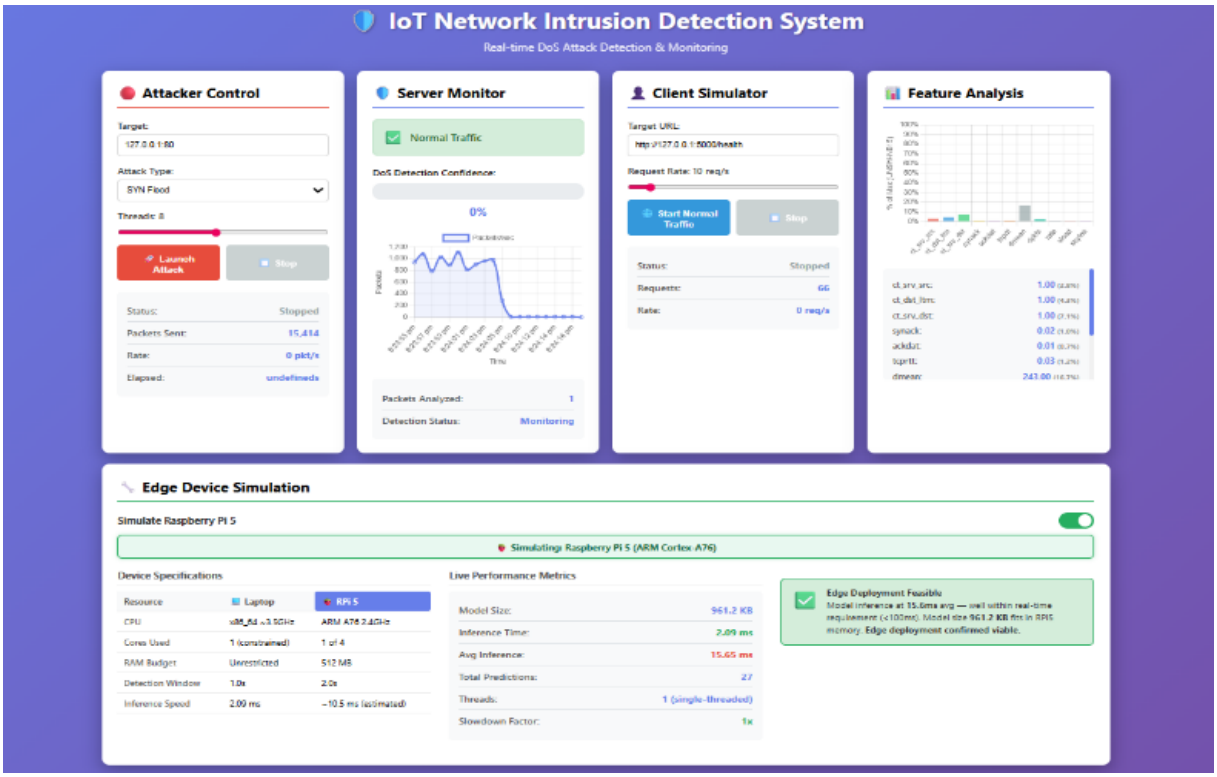

Fig 5.2 Web interface UI, Normal traffic

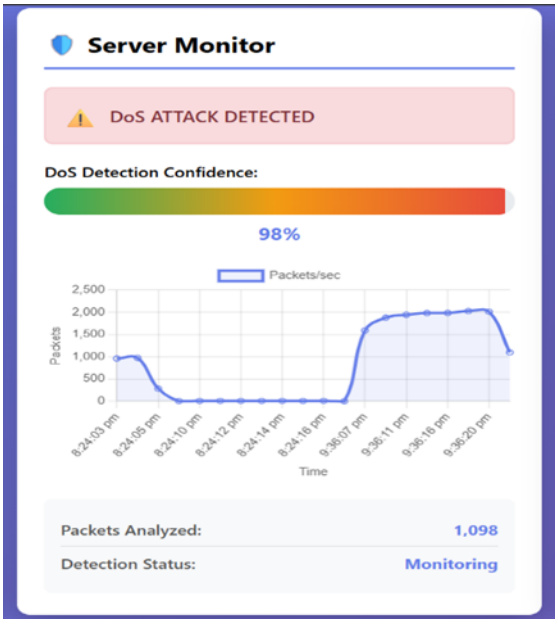

Fig 5.3 SYN flood DoS attack

Computational efficiency was evaluated by measuring inference latency on the development system as seen in Fig 5.4. To approximate edge deployment constraints, a simulated slowdown factor representing Raspberry Pi 5 computational characteristics was applied within the monitoring interface, shown in Fig 5.5.

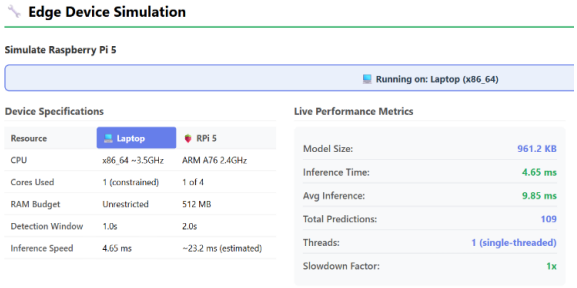

Fig 5.5 Performance on laptop specifications

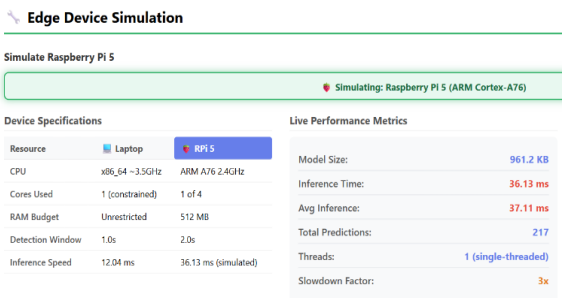

Fig 5.6 Performance on Raspberry Pi 5 specifications

Model footprint was also assessed to determine memory suitability for constrained environments. The serialized LightGBM model occupies around 962KB , which represents a negligible fraction of typical embedded device memory capacity. Compared to deep neural network-based intrusion detection models that may require tens of megabytes or more, the proposed boosting based classifier maintains a substantially smaller footprint while preserving predictive strength.

Overall, the experimental findings demonstrate that behaviour driven feature engineering combined with gradient boosting yields a favorable balance between detection performance and computational efficiency. The system achieves strong classification accuracy, stable generalization, rapid inference latency, and minimal memory usage. These characteristics support the viability of deploying the proposed framework in resource constrained IoT and edge network environments.

VI. CONCLUSION AND FUTURE WORK

This work presented a lightweight machine learning framework for detecting Denial of Service (DoS) attacks in IoT-oriented network environments. Unlike conventional intrusion detection systems that prioritize complex deep learning architectures, the proposed approach emphasizes behavioral feature selection and computational efficiency. By restricting the feature space to eleven statistically and behaviorally significant attributes, the system achieves high classification performance while maintaining minimal computational overhead. Experimental evaluation on the UNSW-NB15 dataset demonstrated that a LightGBM-based classifier can effectively distinguish between normal and DoS traffic. The model achieved high precision and recall, low false negative rates, and stable cross-validation performance, without using high dimensional feature sets or computationally intensive neural architectures.

Beyond testing, the integration of a real time detection pipeline and web-based monitoring interface validated the operational viability of the framework. Live traffic simulation experiments confirmed that the model can detect attack onset within a single aggregation window. Inference latency measurements further demonstrated that the detection process operates

within millisecond scale response times, making it suitable for edge oriented IoT deployment. The compact model size and low memory footprint make it practical to implement this solution on resource constrained devices. Despite these outcomes, several limitations must be acknowledged. First, the system is currently specialized for DoS detection and does not generalize to other intrusion categories such as reconnaissance, exploitation, or data exfiltration. Second, the evaluation relies primarily on benchmark datasets and controlled simulation environments, whereas large-scale real-world deployment scenarios may introduce additional variability which is not captured in the present study. Third, the current implementation assumes fixed time-window aggregation, which may not optimally capture adaptive or low-rate attack strategies.

Future work can be done to incorporate Multi class intrusion detection involving additional attack categories while preserving computational efficiency. Adaptive windowing techniques may be introduced to dynamically adjust aggregation intervals based on traffic behaviour. Further optimization and benchmarking on actual edge hardware platforms will provide deeper insight into deployment scalability. In conclusion, this study demonstrates that behaviour-driven feature engineering combined with gradient boosting can deliver both strong detection performance and practical deployment feasibility. By focusing on efficiency without sacrificing accuracy, the proposed system contributes toward bridging the gap between academic IDS research and real-world IoT security implementation.

REFERENCES

[1] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6.

[2] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2018, pp. 108–116.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

[4] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 3146–3154.

[5] S. M. Pudukotai Dinakarrao, M. S. Iyer, and A. Raghunathan, "EDIMA: Early detection of IoT malware network activity using machine learning techniques," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 39, no. 11, pp. 4288–4299, Nov. 2020.

[6] Y. Meidan et al., "N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, Jul.–Sep. 2018.

[7] H. Hindy et al., "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," IEEE Access, vol. 8, pp. 104650–104675, 2020.

[8] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.