A MACHINE LEARNING FRAMEWORK FOR REAL-TIME DETECTION OF DOS ATTACKS IN IOT ENVIRONMENTS

KOO WAI KIT

A thesis submitted in fulfilment of the
requirements for the award of the degree of
MASTER OF SCIENCE IN CYBER SECURITY

ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION (APU)
GRADUATE SCHOOL OF TECHNOLOGY

JUNE 2025

DECLARATION OF THESIS CONFIDENTIALITY

Author's full name: Koo Wai Kit

Student Number: TP081761

Thesis/Project title: A Machine Learning Framework for Real-Time Detection of DoS Attacks in IoT Environments

---

I declare that this thesis is classified as:

☐ CONFIDENTIAL

☐ RESTRICTED

☑ OPEN ACCESS

I acknowledged that Asia Pacific University of Technology & Innovation (APU) reserves the right as follows:

1. The thesis is the property of Asia Pacific University of Technology & Innovation (APU).
2. The Library of Asia Pacific University of Technology & Innovation (APU) has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Author's Signature: 

Date: 12/6/2025

Supervisor's Signature: ……

Date:

Supervisor's Name: Ts. Dr. Vinesha A/P Selvarajah

DECLARATION OF SUPERVISOR(S)

"I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of
Master of Science in Cyber Security"

Name of Supervisor: Ts. Dr. Vinesha A/P Selvarajah

Signature: ……

Date:

DECLARATION OF ORIGINALITY AND EXCLUSIVENESS

I declare that this thesis entitled
A MACHINE LEARNING FRAMEWORK FOR REAL-TIME DETECTION OF DOS ATTACKS IN IOT ENVIRONMENTS
is the result of my own research work except as cited in the references.
This thesis has not been accepted for any degree and it is not concurrently submitted in candidature of any other degree.

Signature: 

Name: Koo Wai Kit

Date: 12/6/2025

ABSTRACT

The fast growth of Internet of Things (IoT) technologies has significantly increased the vulnerability of networked systems against cyber threats, especially Denial-of-Service (DoS) attacks that exploit the limited resources of IoT devices. This research addresses the challenge by developing a lightweight machine learning (ML) framework for real-time DoS detection in IoT environments using the CICIoT2023 dataset. The study focuses on resource-efficient ML models suitable for constrained IoT systems and evaluates five models, which include Decision Tree, Naïve Bayes, Logistic Regression, Linear Discriminant Analysis (LDA), and Light Gradient Boosting Machine (LGBM). Several tasks were performed before the models were trained and evaluated, including preprocessing the data, balancing classes, and reducing dimensionality from 41 to 20 features through Pearson correlation, Mutual Information (MI) scores, and Random Forest feature importance. The models were assessed using metrics like accuracy, precision, recall, F1-score, training and prediction time, and memory usage. Decision Tree and LGBM emerged as the top-performing models, achieving near-perfect accuracy and F1-scores above 99.9%, and were subsequently selected for deeper analysis. While Decision Tree initially achieved near-perfect performance, its heavy reliance on a single feature (inter-arrival time) caused a severe performance drop when that feature was excluded. In contrast, LGBM demonstrated high robustness and maintained acceptable performance even under feature loss, making it the most suitable model for deployment. The study concludes that LGBM offers the best balance between detection accuracy, robustness, and computational efficiency for real-time DoS detection in IoT systems, contributing valuable insights to the advancement of ML-based intrusion detection systems tailored for IoT security.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DDoS ..... Distributed Denial-of-Service

DoS ..... Denial-of-Service

IoT ..... Internet of Things

k-NN ..... K-Nearest Neighbors

LDA ..... Linear Discriminant Analysis

LGBM..... Light Gradient Boosting Machine

ML ..... Machine Learning

SVM ..... Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 Background

The Internet of Things (IoT) is defined as a flexible ecosystem of interconnected smart devices, which have sensors and communication technologies that enable them to share data in real time via the internet. Its large-scale use has already redefined various fields, including healthcare, smart home, energy, manufacturing, as it has provided automation, efficiency, and data-driven decision-making (Mohan, 2018). Although it has several benefits, the proliferation of IoT has equally increased the attack surface area of cyber threats, especially Denial-of-Service (DoS) attacks, which can greatly impair the operations of a network. As opposed to Distributed Denial-of-Service (DDoS) attacks where multiple sources are used, this study considers only single-source DoS attacks.

A DoS attack is meant to overload a targeted system, network, or service with too much traffic and make it unavailable to legitimate users (Cloudflare, n.d.). The effects of such attacks can be especially harmful in IoT settings, where devices usually are constrained in processing power, memory, and bandwidth. In the context of IoT, DoS attacks can disable communication, slow down network speed, and shut down IoT systems. These issues can result in substantial operational and financial losses (Altulaihan et al., 2024).

Traditional Intrusion Detection Systems (IDS) are usually made to fit in traditional computing architecture and they hardly adjust to the limitations and heterogeneity of IoT networks. Such legacy systems are not able to sustain performance in conditions that are unique to the IoT, like constrained hardware resources, the need to interoperate with a wide variety of communication standards, and extremely dynamic traffic patterns (Elrawy et al., 2018). Consequently, the demand is increasing in terms of lightweight, real-time detection frameworks that would be optimized according to the specific demands of IoT infrastructures.

In order to fill this gap, the research proposes a machine learning (ML) based framework for real-time detection of DoS attacks in IoT environments. The experiment is performed on the UNB CIC IoT 2023 (CICIoT2023) dataset, which is a state-of-the-art benchmark dataset,

specifically made to represent realistic IoT scenarios, involving more than 100 IoT devices and 33 different types of attacks, carried out with malicious IoT nodes (Neto et al., 2023). Notably, in this project, the category of attacks under consideration is limited to the DoS attack and all other kinds of attacks including DDoS, Reconnaissance, or Web-based attacks are lumped together and marked as non-DoS to perform binary classification. The proposed framework is implemented in Kaggle Notebook by following a complete ML pipeline. It includes several steps like data preprocessing, class balancing, feature engineering, and comparison of different ML models.

With the particular emphasis on detecting DoS attacks and the use of a well-structured ML pipeline, the study provides a viable and scalable solution to the problem of improving IoT security via efficient and timely threat detection.

1.2 Problem Statement

Since IoT systems are growing both in size and popularity, they are becoming more appealing to cyberattacks, especially Denial-of-Service (DoS) attacks. They are carried out to overwhelm a system or service with traffic in order to impact communication and performance, which could lead to costly service outages. This threat is particularly severe in IoT settings due to resource constraints and the dynamic nature of IoT traffic. Although a lot of work has been done to identify DoS attacks by employing conventional security tools, such tools often fail in practice when they are applied to actual IoT networks.

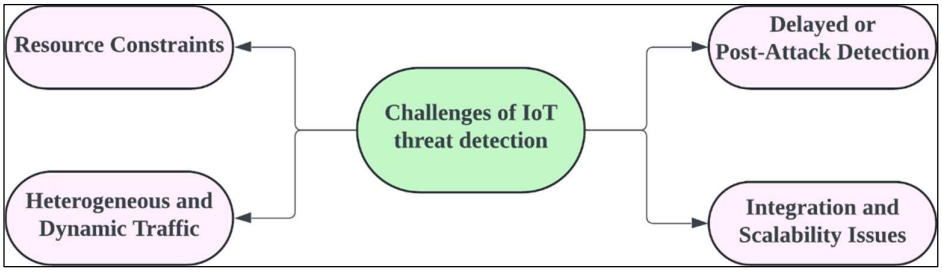

Figure 1: Challenges in IoT threat detection

There are four significant challenges that negatively affect the efficiency of DoS attack detection in IoT setup:

1. **Resource Constraints**: IoT devices are usually characterized by low CPU power, memory and battery life, limiting their capability of running complex security mechanisms. They require basic security solutions that can detect threats swiftly without

demanding high usage of computing power or energy (Agbedanu et al., 2022). However, traditional IDS and real-time monitoring tools need more computational power than what most IoT devices can handle (Altulaihan et al., 2024). This implies that it is important to develop lightweight and efficient detection systems that do not sacrifice on detection accuracy.

2. **Heterogeneous and Dynamic Traffic**: IoT ecosystems consist of a large variety of devices, communication protocols, and traffic patterns. Such variety makes it complicated to develop consistent traffic baselines or signatures to use in anomaly detection (Altulaihan et al., 2024). This means that security systems need to be flexible enough to generalize across device types and use cases.
3. **Delayed or Post-Attack Detection**: Many current detection methods are reactive, as they examine logs or traffic patterns after an incident has already taken place. This post-event analysis increases the response time and exposes IoT environments to vulnerability at critical attack windows (Rani et al., 2024). It has also been found that relying on past data to identify threats is not enough, and this drives security teams to become more proactive (Chawla & Thamilarasu, 2018). Hence, in order to mitigate threats in real-time, real-time detection is necessary, especially in systems where uptime and responsiveness are crucial
4. **Integration and Scalability Issues**: Integration of security solutions for IoT devices remains challenging because various security protocols exist between different systems. Most IoT systems often use a number of security measures simultaneously, including IDS, encryption and hardware segmentation, which may lead to compatibility problems (Hwang & Kim, 2021). Protection efforts become difficult because IoT devices lack uniform security standards which create challenges for implementing a unified security framework (Singh et al., 2023). A universal solution for securing all IoT devices does not exist (Kim, 2024).

While ML has shown promise in detecting DoS attacks, they require significant improvements to work efficiently across IoT devices in real-time settings. Many existing ML-based solutions demand high computational power, which makes them unusable in environments with limited resources. The lack of a lightweight, flexible and scalable ML-driven detection system stands as a major problem in protecting IoT devices (Wardana et al., 2024). Considering such limitations, it becomes evident that a lightweight, flexible, and scalable ML-based framework explicitly optimized to DoS attacks detection in IoT setting is required.

1.3 Research Questions

This study explores the development of a lightweight, real-time ML framework for detecting DoS attacks in IoT environments. The investigation is guided by the following research questions (RQs):

- **RQ1**: How effective are lightweight ML models in differentiating DoS attacks from other benign and malicious traffic in a practical IoT setup?
- **RQ2**: How do feature selection and reduction techniques affect the performance and efficiency of models trained on IoT traffic data?
- **RQ3**: Among some commonly used lightweight models, which model offer the best trade-off between accuracy, detection speed, and model size in detecting DoS attacks?
- **RQ4**: How does the removal of the most important feature affect the robustness and reliability of selected models?
- **RQ5**: How can a lightweight DoS detection framework, trained and tested on publicly available IoT dataset demonstrate the potential for future real-world application in IoT security?

1.4 Aim and Objectives

1.4.1 Aim

The research aims to develop and evaluate a real-time ML framework for DoS attack detection in IoT environments using a large-scale, publicly available IoT dataset. The research outcomes will assist the advancement of IoT security strategies while guiding developments of new intrusion detection systems.

1.4.2 Objectives

The research will achieve its aim through the following research objectives (ROs):

- **RO1**: To preprocess and explore the CICIoT2023 dataset by separating and labelling DoS traffic for binary classification.
- **RO2**: To apply and assess feature selection and reduction techniques for improving model performance.
- **RO3**: To compare multiple lightweight machine learning models based on their performance metrics.
- **RO4**: To evaluate the robustness of the top-performing models by analyzing the impact of removing the most important feature on their performance.

1.5 Significance of the Study

As IoT devices are being broadly used in various sectors including healthcare, smart homes and industrial automation, it has become more critical to secure the networks against cyber threats. This study contributes to IoT security by addressing the vulnerabilities of DoS attacks and proposing a real-time detection framework.

For traditional IDS, their high resource requirements and incapability in adapting to dynamic network conditions make them unsuitable for IoT environments (Elrawy et al., 2018). This research addresses these limitations through the use of lightweight ML models that provide effective security without overwhelming the IoT devices (Altulaihan et al., 2024). Studies have shown that ML models like Random Forest (RF) could achieve more than 99% accuracy in detecting DoS attacks, which highlights the potential of ML-based approaches in this study (Alsulaiman & Al-Ahmadi, 2021). By testing on the CICIoT2023 dataset, the research demonstrates how models like Decision Tree and Light Gradient Boosting Machine (LGBM) could be trained to perform well in DoS attacks detection with low latency.

In addition, real-time detection is crucial for IoT security, since traditional methods often rely on delayed post-event analysis (Rani et al., 2024). This study focuses on proactive threat identification in order to achieve real-time threat detection, thereby minimizing system downtime and potential financial losses (Uhm & Pak, 2022). This approach offers benefits to critical sectors like healthcare, since delayed responses to DoS attacks could compromise patient safety. Moreover, using ML-based models for detection can reduce the number of false positives generated by conventional security tools. This will simplify security management, thus improving business continuity (Hulayyil et al., 2023).

From a practical point of view, this research will be beneficial for several stakeholders, including IoT product manufacturers, developers, and end-users (Abdul-Ghani & Konstantas, 2019). IoT product manufacturers can integrate the lightweight framework into IoT devices without costly hardware changes, allowing them to produce secure and market-ready IoT products that address growing consumer security demands. For IoT developers, using pre-optimized ML models allows them to quickly deploy secure IoT applications without the need for writing custom code, thus speeding up the development process. Meanwhile, IoT users in

sectors like healthcare and industrial automation gain improved protection against service disruptions caused by DoS attacks.

This study fills a significant gap in available literature and datasets, as they tend to place DoS and DDoS attacks in the same category, whereas these two types of attacks have different properties. Consequently, it offers targeted knowledge and practical guidance to researchers and practitioners who want to improve security in IoT by using accurate and data-oriented methods. The study also ensures reproducibility by using publicly available datasets and ML models.

1.6 Research Scope and Limitations

This research is conducted within the scope of a 12-week project and is dedicated to the implementation and testing of a lightweight ML-based detection framework, specifically on DoS attacks in IoT environments. The aim is to improve the detection of DoS attacks by experimenting with different ML models on publicly available attack datasets, and not by physical implementation or deployment.

1.6.1 Technical Constraint

In order to be realistic with the time and resources available, the scope of the study is restricted to small-scale IoT network setups like those present in smart homes or smart offices. The study only focuses on DoS attacks and does not cover DDoS attacks or other threats, such as malware or ransomware attacks. The developed detection models are software-based and are trained offline, using the CICIoT2023 dataset. Emphasis is placed on using lightweight and efficient ML models, as they are easier to integrate with IoT systems, which are usually rather limited in computing resources.

1.6.2 Methodological Limitations

The project will not require any physical deployment and testing on real IoT devices due to time, cost, and resource limitations. Consequently, the model performance has been tested with the pre-gathered traffic data provided in the CICIoT2023 dataset that includes benign and attack scenarios with real IoT devices. Moreover, no simulation tools like Mininet-WiFi were used because the objectives of the study were achieved by performing analysis and conducting experiments in a controlled data science setting. The evaluation is based on the model performance, inference time and memory consumption, without testing for real-time deployability or scalability.

1.6.3 Deliverables

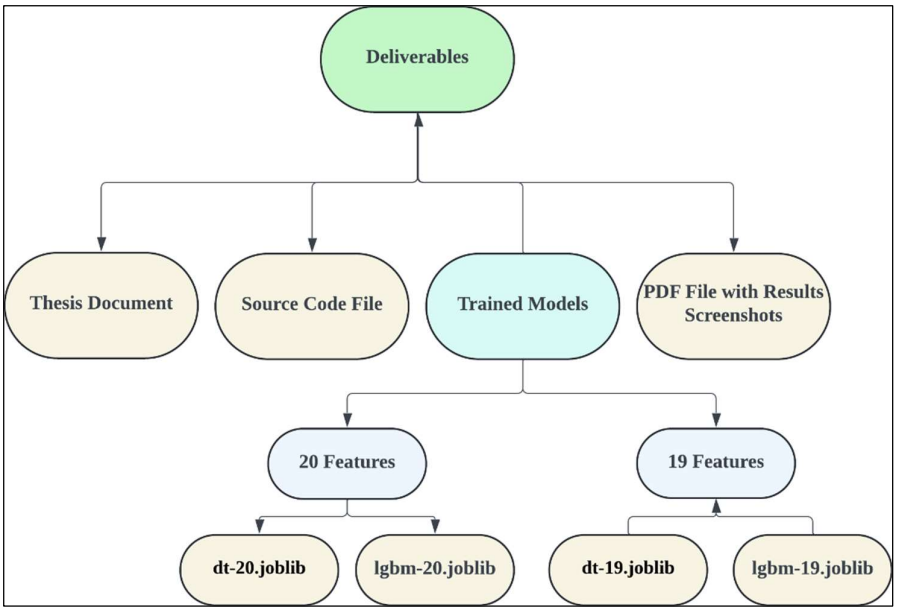

Figure 2: Project deliverables

This project produces the following deliverables:

- A thesis document, which is a comprehensive record of the project's background, methodology, results and discussion.
- A source code file that contains the complete code for the complete ML pipeline (dissertation-project-code.ipynb).
- Two saved model object files trained on all 20 selected features (dt-20.joblib, lgbm-20.joblib).
- Two saved model object files trained on 19 selected features, excluding the top-ranked feature (dt-19.joblib, lgbm-19.joblib).
- A PDF file containing screenshots of key evaluation results and performance metrics (Collected Results.pdf).

These deliverables offer not only practical deliverables that can be replicated but also academic contribution to future research.

1.6.4 Exclusions

In order to keep things clear and focused, this research has a number of exclusions. It does not cover implementations at the hardware level, such as firmware-based defenses or embedded IDS. The developed model is only designed to detect four types of DoS attacks as simulated in

the CICIoT2023 datasets, which includes TCP flood, HTPP flood, SYN flood, and UDP flood. Other types of DoS attacks will not be tested. Additionally, the model only handles binary classification, and it will not be able to give information about the different attack classes. Scalability, deployment, and real-time integration remain outside the scope of this study. These points are suggested as the possible directions of future research.

1.7 Thesis Structure

This thesis consists of five chapters that cover different aspects of the research. The chapters are as follows:.

- **Chapter 1 (Introduction)**: Provides an overview of the study, establishing its foundation by presenting the research background, problem statement, research questions, objectives, significance and scope.
- **Chapter 2 (Literature Review)**: Reviews existing work related to IoT security, DoS attacks, intrusion detection systems (IDS), and machine learning-based approaches to threat detection.
- **Chapter 3 (Research Methodology)**: Outlines the research methodology, with details of the dataset used, data preprocessing techniques, feature selection techniques, model development process, evaluation metrics, and experimental setup for model comparison. The chapter also describes the implementation process of the entire ML pipeline.
- **Chapter 4 (Results and Discussion)**: Reports the model assessment results, evaluates the performance of the models and discusses important findings in relation to the research questions.
- **Chapter 5 (Conclusion and Recommendation)**: Summarises the research outcomes, highlights the study's contributions and limitations, and provides suggestions for future work.

CHAPTER 2

LITERATURE REVIEW

Chapter 2 provides an extended literature review concerning the state of research regarding IoT security, paying special attention to the challenges of Denial-of-Service (DoS) attacks and the role that intrusion detection systems (IDS) play, especially those that are based on machine learning (ML) algorithms. The chapter is fundamental in order to set the scholarly basis of the study, critically analyzing the previous research, revealing gaps in the existing methods, and exposing the areas that require extended research. Its value is that it helps formulate the theoretical part of the project, set the direction to select adequate models, and explain the reasoning behind the methodology of creating a lightweight ML-based detection framework.

The chapter starts with providing an overview of IoT systems and the security landscape associated with them (Section 2.1), which forms the basis of understanding why such environments are especially susceptible to attacks. In Section 2.2, the nature and consequences of DoS attacks in IoT environments are discussed, which highlights the relevance of this particular threat. Section 2.3 is dedicated to the role of IDS in IoT environments, where the traditional and modern methods of detection are discussed. Moreover, Section 2.4 discusses the lightweight ML models that are suitable for IoT environments. Section 2.5 discusses some well-known datasets used in the field of IoT intrusion detection research. Lastly, Section 2.6 investigates common evaluation strategies that assess the effectiveness and reliability of detection models.

2.1 Overview of IoT and Its Security Challenges

The Internet of Things (IoT) has transformed the contemporary digital infrastructure and made it possible to interconnect communication between a vast and expanding range of devices, including wearable health trackers, smart home devices, industrial control systems and self-driving cars. This ubiquitous connectivity is useful in increasing the efficiency of operations and convenience to users in many fields, such as healthcare, smart cities, manufacturing, and transportation. Nevertheless, ubiquity, heterogeneity, and decentralization that define the usefulness of IoT systems also expose them to various security threats.

Heterogeneity of IoT environments is one of the main security challenges. The devices in IoT systems have many differences in capabilities, operating systems, communications standards and deployment environments. Such heterogeneity makes it difficult to enforce consistent security policies (Olaniyi et al., 2023). Besides, a large number of IoT devices are resource-constrained by nature, having limited processing capabilities, memory, and energy. This limits their capability to run traditional security controls, including firewalls and endpoint protection systems (Mutambik, 2024). These problems are regularly worsened by the minimal design attention to security at the manufacturing stage, which leaves gadgets vulnerable even to unsophisticated attacks (Dodson et al., 2021).

The absence of standardization in the communication protocols and security frameworks also hampers the development of effective protection measures. The lack of security guidelines that are universal, cross-vendor, and cross-platform, as Xenofontos et al. (2022) claim, is another factor that leads to the fragmentation of defenses and the lack of coordination when responding to a threat. This has become especially problematic when considering threats like DoS attacks, malware, and physical tampering, which are attack vectors that abuse the openness and interoperability of IoT systems (Sahu & Mazumdar, 2024). This interconnectedness which makes IoT possible also expands the attack surface, increasing the possibility of cascading failures across devices and networks.

Furthermore, the increased usage of cloud infrastructure to store and process the data generated by IoT devices pose the issue of data integrity, confidentiality, and controlled access to data. IoT data traffic is extremely large and dynamic, which means that it does not fit into the traditional perimeter-based security model. According to Mutambik (2024), it is important to note that solutions need to be scalable and interoperable to be able to work in the wide range of real-world deployments. Such complexities do not just require technical countermeasures, but also architectural re-designs that can factor in security as a primary consideration, and not as an afterthought.

Ethical issues and data privacy are also at the center of the wider discussion of IoT security. IoT devices are frequently used in personal or critical environments, like patient observation or intelligent surveillance systems, where unauthorized access can cause substantial damage. To solve this problem, it is necessary to go beyond technical solutions, such as encryption and

access control, and come up with elaborate regulatory frameworks and ethical guidelines (Tawalbeh et al., 2020).

Security-by-design has emerged as one of the key principles towards building resilient IoT ecosystems. The given approach proposes the integration of security-related mechanisms across the device and system lifecycle, including initial design and development, deployment, and decommissioning (Al-Otaibi et al., 2022). In order to keep up with the changing threats, proactive risk evaluation, constant monitoring, and secure update mechanisms are a necessity.

Finally, industry, academia, and regulatory groups should work together to deal with these security issues in an integrated manner. Shared threat intelligence and unified standards can promote consistency, transparency, and resilience within the IoT ecosystem (Nzeako et al., 2024). With the increasing popularity of IoT, the significance of coordinated efforts in the area of cybersecurity becomes even greater.

To conclude, the IoT paradigm is potentially transformative, but it can only succeed once the severe security issues are addressed. These include the need for standardized frameworks, lightweight and flexible defense mechanisms, privacy-preserving system architectures and collaborative governance systems. The potential of IoT can be fully achieved only with a multidimensional and proactive approach to security without affecting safety, privacy, and trust.

2.2 DoS Attacks in IoT Environments

One of the most common and disruptive threats to IoT environments is DoS attacks. The goal of such attacks is to consume the resources of the attacked devices or networks so that the services become unavailable to legitimate users. Such attacks are especially easy to carry out against IoT systems due to their inherent vulnerabilities caused by the variety of devices, limited hardware capacity, and inconsistent security measures. With the growing tendency to implement IoT networks in critical infrastructures like smart cities, healthcare, and industrial control systems, the impact of successful DoS attacks may range beyond the temporary disruption of services and result in major operational failures. In the CICIoT2023 dataset, there are four types of single-source DoS attacks, as shown in Figure 3 below.

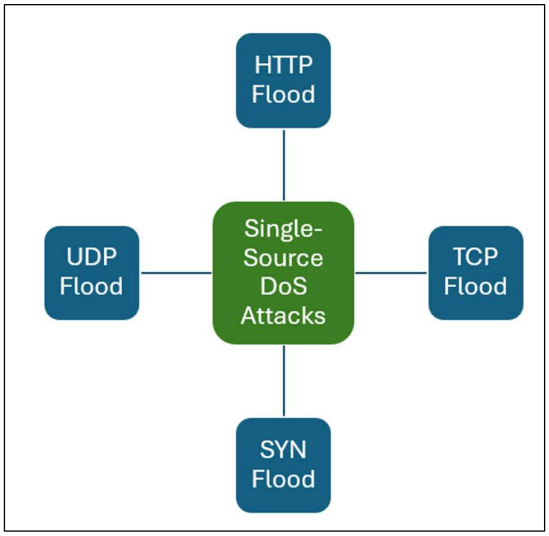

Figure 3: DoS attack types in CICIoT2023 dataset

Additionally, IoT devices usually have significant constraints in terms of their processing power, memory, and energy. According to Kaur et al. (2024), such drawbacks make it challenging to implement traditional security measures, which means that devices are vulnerable to lightweight yet effective attack vectors. DoS attacks normally involve overwhelming the devices or networks with too many requests to take advantage of the incapability of the devices to process large amounts of traffic. Another major problem is IoT devices that have been compromised are often used to build botnets, which increases the strength of attacks and allows launching DDoS attacks at scale (Kim et al., 2023).

DoS attacks have more implications than just the unavailability of services. As Husnain et al. (2022) state, such interruptions may cause data integrity issues, privacy violations, and, in the worst case, harm the safety of the population. For instance, in smart cities, a DoS attack on IoT-powered traffic control or emergency systems could result in real-world consequences, including traffic congestion or delays in emergency response. On the economic front, the downtime caused by the DoS attacks may lead to huge financial losses, especially in industries that rely on real-time IoT data to run their businesses. Ghali et al. (2021) emphasise that long-lasting service outage can destroy user trust and institutional reputation.

Organizations should implement effective mitigation strategies in order to overcome such threats. IDS, particularly the ones combined with ML algorithms, have become a strong solution in detecting traffic anomalies that could represent DoS behavior (Alsulami et al., 2022). ML-

based IDS has the ability to adjust to changing attack behaviors and allow early detection, minimizing the possibility of service interruption. Such systems are especially useful in IoT scenarios, in which signature-based approaches can be insufficient to identify new or stealthy attacks.

Along with IDS, a multi-layered defense approach is necessary. Musthafa et al. (2024) and Kim et al. (2023) suggest hybrid solutions that involve traditional defenses and smart detection systems, so that even when one layer is breached, others would still be operational. Frequent updates of software, safe coding, and encrypted communication protocols are also important. Even lightweight communication protocols like CoAP need to be reinforced with protection against flooding and spoofing (Almeghlef et al., 2023).

Finally, addressing DoS threats in the context of the IoT will need the cooperation of several stakeholders, including developers, manufacturers, regulators, and researchers, to define the common security standards and practices. Although technical solutions play a major role, policy frameworks and awareness campaigns cannot be undermined in establishing a secure IoT ecosystem. Within the scope of this paper, understanding the nature and effects of DoS attacks are of primary importance to the design of a successful detection framework.

2.3 Intrusion Detection Systems in IoT Networks

IDS are necessary to secure IoT networks, as they can constantly monitor the traffic and respond to malicious attempts or policy violations. With the ever-growing popularity of IoT in various different fields, such systems have become fundamental in reducing security risks that are posed by the distributed and heterogeneous nature of IoT devices. Since modern IoT deployments are large-scale and complex, there is a need to ensure the use of scalable and adaptive IDS solutions.

Traditional IDS solutions are normally grouped into signature-based and anomaly-based systems. Signature-based IDS is founded on pre-determined patterns of recognized threats to detect intrusions (Khraisat et al., 2019). These systems are effective against attacks seen in the past, but they are inadequate at identifying zero-day exploits or new forms of known threats, which are common in dynamic IoT environments. In contrast, anomaly-based IDS models identify the deviation in the defined normal behavior and have the ability to detect novel or unknown threats (Mliki et al., 2021). The problem with these systems however is that they tend

to have a high false positive rate, which can cause alert fatigue and erode confidence in the reliability of the system.

In order to address these drawbacks, there has been a growing interest in studying the modern, ML-driven IDS, which are more flexible and intelligent. Such systems have the ability to learn complicated patterns in network data and can generalize to identify new types of attacks they have never seen before, such as DoS attacks that specifically target IoT vulnerabilities. Deep neural networks, decision trees, support vector machines, and ensemble models are some of the techniques that have proved useful in enhancing the accuracy of detection and the response times (Sharipuddin et al., 2021; Ariffin et al., 2022).

New developments are also in the direction of collaborative and privacy-preserving solutions, like federated learning, that enables distributed IoT nodes to learn joint detection models without sharing raw data (Reyaz & Vanitha, 2024). It is especially applicable in IoT scenarios where data locality and the limited bandwidth utilization are major concerns. These decentralized systems make it more scalable and minimize the chances of data breach that can occur in a centralized system.

One of the long-standing issues when it comes to deploying IDS in an IoT setup is related to the limited capabilities of most devices. Edge devices have limited computation power, memory, and battery life, which means that it is not feasible to operate traditional, resource-intensive security tools on those devices. Researchers have reacted by suggesting lightweight IDS models that are tuned for a balanced trade-off between performance and efficiency. There are methods to decrease the computational overhead but still preserve the ability to detect effectively, including distributed detection and feature selection (Alawsi, 2023).

Moreover, blockchain technology has also been considered as a supplementary measure to make IDS more effective. Blockchain can provide auditability, integrity and trust among the IoT nodes through decentralized and tamper-evident logging of intrusion alerts. When used with AI-based IDS, blockchain enables safe information exchange and response coordination of responses between devices (Bediya & Kumar, 2021). Such a combination helps to build a more resilient architecture in terms of resistance to coordinated and large-scale attacks.

To sum up, IDS are still a foundation of IoT security. With the evolving threat landscape, it is critical to shift the focus by moving on from the traditional style of detection to intelligent, scalable, and collaborative models. Continued research and innovation on the subject will play a central role in making IoT infrastructures more resilient to the known and unknown cyber threats. Within the framework of this project, IDS represent the core of the suggested detection approach to DoS attacks in IoT settings.

2.4 Lightweight ML models for IoT Environments

The importance of lightweight ML models as a part of securing IoT environments is becoming actively acknowledged, with CPU, memory, storage, and energy being frequently extremely limited in such settings (Farfoura et al., 2025). Lightweight models are intended to be used on the hardware-restricted IoT devices, whereas conventional ML models can require substantial processing power and significant training durations. Their low latency response and minimal resource usage make them especially well-suited to real-time intrusion detection, anomaly classification and malware detection on a wide variety of IoT deployments (Amgbara et al., 2024).

Several lightweight ML models have been found to be useful in IoT settings. These models provide a balance between model simplicity, computational efficiency, and prediction accuracy, which is why they are well-suited to run on-device security tasks, like anomaly detection and intrusion prevention.

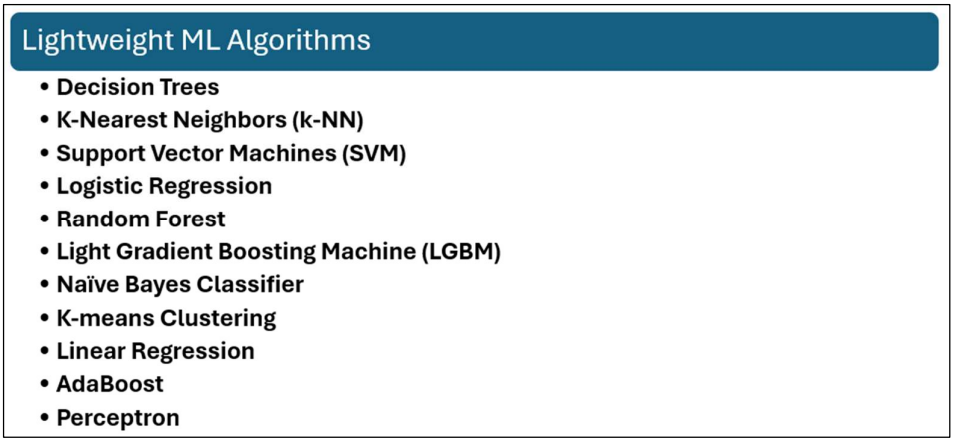

Figure 4: Lightweight ML algorithms for IoT environments

A summary of lightweight ML algorithms commonly used and their applicability to IoT settings are given below:

- **Decision Trees**: Decision trees are simple rule-based classifiers which have minimal memory and computing requirements. They are very efficient and can be used in real time classification in IoT devices due to their logical structure and simplicity of interpretation (Amgbara et al., 2024).
- **K-Nearest Neighbors (k-NN)**: Being a non-parametric algorithm, k-NN assigns data according to their closeness to known examples. It is simple to apply and performs effectively in anomaly detection, but its memory consumption may grow on bigger datasets (Amgbara et al., 2024).
- **Support Vector Machines (SVM)**: In IoT applications, SVMs are applicable when the size of the data is small and moderate. They are efficient and commonly applied in binary classification tasks, but they can be computationally expensive when dealing with a large dataset or when using a complex kernel (Amgbara et al., 2024).
- **Logistic Regression**: Logistic regression is a linear classification model that can be applied on binary classification under restricted conditions. Its minimal computation and fast inference time makes it suitable for simple anomaly or intrusion detection (Amgbara et al., 2024).
- **Random Forest**: Random Forest is an ensemble of decision trees, and it offers better predictive power while using moderate resources. It generalizes better and is more robust than single-tree models, but it might be too heavy on some resource-constrained devices (Amgbara et al., 2024).
- **Light Gradient Boosting Machine (LGBM)**: LGBM is a type of Gradient Boosting Machine that is developed to be fast and efficient. It has an optimized tree construction procedure, which makes it more applicable to be used in a low-resource setting, without losing the gradient boosting advantages (Farfoura et al., 2025).
- **Naïve Bayes Classifier**: Naïve Bayes Classifier is a probabilistic model that is very efficient and simple to implement. It is often applied in IoT to perform malware detection and intrusion classification because of its low memory and processing requirements (Amgbara et al., 2024).
- **K-means Clustering**: K-means is an unsupervised learning method applicable in detecting anomalies without requiring labeled examples. It is computationally lightweight and able to identify new threats in dynamic IoT environments (Amgbara et al., 2024).

- **Linear Regression**: Despite being a regression model, linear regression may be utilized in simplified IoT scenarios on some basic anomaly detection tasks. It is very lightweight and fast yet restricted in its ability to work with complex patterns (Amgbara et al., 2024).
- **AdaBoost**: AdaBoost improves accuracy by concentrating on the instances that have been misclassified. It has a light computational overhead and can be applied to enhance basic classifiers in the IoT setting (Alve et al, 2025).
- **Perceptron**: Perceptron is a simple neural model which can be used in binary classification. Due to its simplicity it can be made to work on very minimal hardware, however it is not as effective in multi-class or complicated tasks (Neto et al., 2023).

Overall, lightweight ML models are needed to apply security solutions directly on IoT devices. Through the selection of algorithms that offer a trade-off between computational efficiency and effectiveness, the threats can be identified and counteracted at the local level without relying on cloud computation or substantial infrastructure.

2.5 Datasets for IoT Intrusion Detection Research

The intensive growth and variety of IoT devices have caused complicated security issues, which require effective IDS that suit the nature of IoT environments. In order to design, train and test these systems, researchers require high quality datasets that are representative in terms of traffic patterns and contain a variety of attack instances. In this section, an overview of well-known datasets actively utilized in the research of intrusion detection in IoT is given.

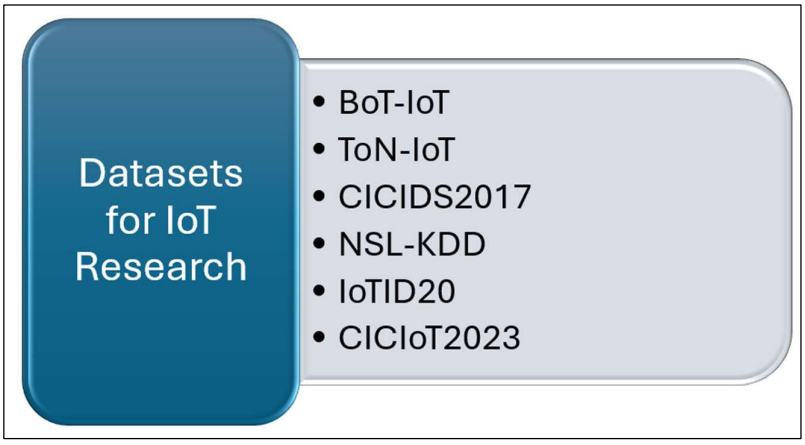

Figure 5: Popular datasets for IoT research

2.5.1 BoT-IoT

The BoT-IoT dataset covers various types of attacks, including DDoS and other network-based intrusions. It is designed to represent realistic IoT network traffic, and it offers a good foundation for training and validating IDS models. The extensive design and simulated IoT setting of the dataset offer substantial value to security studies (Amien et al., 2024).

2.5.2 ToN-IoT

The ToN-IoT dataset focuses on IoT ecosystems in particular, and it includes telemetry, network, and operating system data gathered in benign and malicious situations. It contains a wide range of attack categories such as scanning and DDoS, which allows researchers to test IDS performance on several data modalities. Its focus on recent threat patterns makes it more applicable to the research on ML-based intrusion detection (Alsulami et al., 2023).

2.5.3 CICIDS2017

CICIDS2017 is a popular benchmark dataset in intrusion detection research. It consists of a rich combination of normal and attack traffic, spanning a number of protocols and attack types. It is not IoT-specific, but its extensive nature means it can be applied to an IoT context, especially when comparing the traditional and advanced detection models (Rodriguez et al., 2023).

2.5.4 NSL-KDD

Despite being outdated, the NSL-KDD dataset is still considered a reference in the study of IDS. It addresses some of the shortcomings present in the original KDD99 dataset by reducing redundancy as well as enhancing class balance. Although it does not relate much to present-day IoT settings, it remains a point of baseline comparisons when assessing IDS methods (Tyagi & Kumar, 2021).

2.5.5 IoTID20

The IoTID20 dataset concentrates on the traffic generated by IoT devices under benign and attack conditions. It contains labeled data of many different forms of attacks, and thus it can be used to train supervised learning models. It is designed in such a way that it can be used to conduct focused research on intrusion detection specific to consumer-grade IoT settings (Karamollaoğlu et al., 2022).

2.5.6 CICIoT2023

CICIoT2023 dataset is a new and complete benchmark that aims to overcome the current shortcomings in the research of IoT intrusion detection, especially the absence of real-world activity of IoT devices and attacks launched by compromised IoT devices. It was generated in a smart home laboratory setting consisting of 105 IoT devices, 67 of which were actively engaged in attack scenarios, providing a realistic emulation of heterogeneous IoT traffic. It is worth noting that every single one of the 33 different attacks represented in the dataset was initiated by a malicious IoT device, which is a major difference compared to previous datasets that usually used non-IoT systems to attack (Neto et al., 2023).

The dataset covers seven broad categories of attacks, namely DDoS, DoS, Reconnaissance, Web-based, Brute Force, Spoofing, and Mirai-related threats, with many of them being underrepresented in the older datasets. The network traffic was filtered and saved in pcap format and further processed into CSV files with 47 extracted features including flow duration, protocol type, and packet header length. Aggregation of features by use of fixed packet windows (10 or 100 packets) allows more stable modeling across a wide range of traffic patterns. Overall, CICIoT2023 offers a powerful benchmark and model development base with more than 548 GB of raw data (Neto et al., 2023).

2.6 Methods for ML Model Evaluation

When applied to the context of IoT intrusion detection, ML model evaluation takes a complex form, where not only model performance is taken into account, but also limitations related to practical deployment. A commonly used method to assess a model's effectiveness is by using metrics like accuracy, precision, recall, and F1-score. Figure 6 illustrates these metrics.

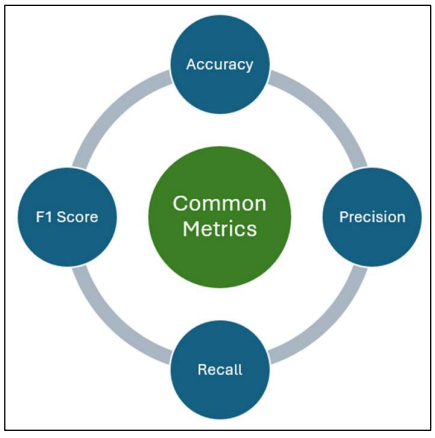

Figure 6: Common metrics for ML model evaluation

Although accuracy gives an overall picture of the performance, it may not be informative in the case of highly imbalanced data, as it is common in IoT traffic where benign traffic vastly outnumbers malicious traffic (Govindaraju et al., 2022). In security applications, precision and recall are of concern especially because high false positive or false negative rates may result in unnecessary resource waste or unnoticed threats (Sayed et al., 2023). The F1-score, a balanced combination of these two measures, is commonly applied to assess the balance of the model, whereas Area Under the Receiver Operating Characteristic Curve (AUC-ROC) allows measuring the tradeoff between detections and false alarms (Alabdulwahab et al., 2024; Maqbool, 2024).

In addition to classification metrics, other aspects of evaluation are model efficiency, particularly with regards to processing time and the use of computational resources, which is essential when deploying the model to resource-limited IoT devices (Abdulla & Jameel, 2023). In order to guarantee stable performance, scholars usually use methods like k-fold cross-validation to alleviate overfitting and enhance generalization (Zhang et al., 2024). Standardized testing and comparison of results across research is commonly done on benchmark datasets. Additionally, feature selection techniques like Recursive Feature Elimination (RFE) can be used to simplify the models and improve the detection performance and computational efficiency (Alasmari & Alhogail, 2024).

CHAPTER 3

RESEARCH METHODOLOGY

The chapter provides the methodological framework followed to conduct this study, which is developing and testing lightweight ML models to detect DoS attacks in IoT contexts. It describes how data preparation, model selection, and evaluation of performance were done systematically to identify security threats in a resource-limited environment.

This chapter is essential to make the research process rigorous, transparent, and reproducible. It gives the basis on which the results in subsequent chapters will be anchored, and it also makes the results not just valid, but also contextually appropriate to the IoT security landscape. This chapter contributes towards the thesis goal by providing a clear description of the experiment workflow to identify practical and efficient ML-based intrusion detection approaches suitable to be used in IoT networks.

The chapter starts with the description of the research design to contextualize the entire approach. Next, there is a clear explanation of the dataset selected and its data characteristics. The following stages, data preprocessing and feature selection and dimensionality reduction, describe the process of optimization and transformation of raw data to make it ready and optimized to be used by ML algorithms. The next stage will concentrate on model selection, in which the lightweight ML models suitable for this study are identified. Lastly, the model training and validation section will detail how the models have been trained and tested, and how validation was done to select a final model.

3.1 Research Design

The study follows a structured experimental design to create and test lightweight ML models that can identify DoS attacks based on IoT network traffic data. Since network traffic is complex to investigate, the study focuses on the efficiency, accuracy and interpretability of the models. A commonly used ML pipeline is illustrated in Figure 7 below. The team behind the CICIoT2023 dataset has already completed the first two stages of the pipeline, i.e., problem definition and data collection. In this study, the focus is on using the CICIoT2023 dataset to implement the remaining steps of the pipeline in a Kaggle Notebook, which includes feature engineering, data preparation, model selection, performance evaluation, and model validation.

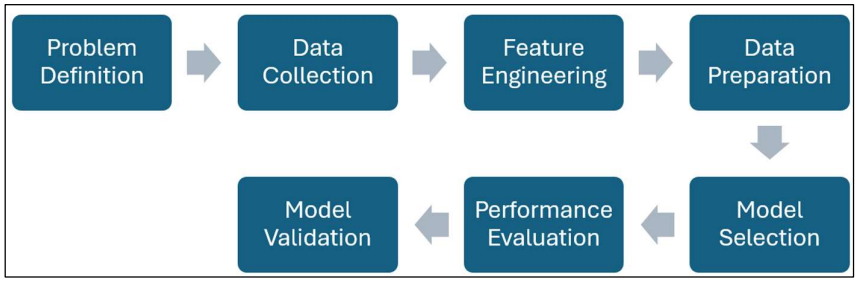

Figure 7: Standard machine learning pipeline

The CICIoT2023 dataset has been chosen due to its relevancy to the IoT network traffic and representation of DoS attacks. Only the first 20 files of the total 169 files available were used because of computational limitations. Such selective approach also reflects the resource limitations that IoT devices often have to operate under, bringing the analysis closer to the actual conditions. Exploratory data analysis was performed on the dataset to get some general ideas about the structure, the statistical properties of the data, and the distribution of labels. In order to reduce the multi-class attack labels into a binary classification problem, all the DoS attack types were classified as DoS, and all the other types (e.g., DDoS, web-based attacks, benign traffic) were classified as non-DoS.

Handling missing values, converting categorical variables, and managing class imbalance were part of the data preprocessing steps. For instance, DoS samples in the training set were upsampled with Synthetic Minority Oversampling Technique (SMOTE), and the number of non-DoS samples was reduced with random undersampling, resulting in a balanced distribution of the training set. In addition, feature selection was performed in several ways: Pearson correlation with the label and between features, Mutual Information (MI) scores, and feature importance ranks obtained with a Random Forest classifier.

During dimensionality reduction, features that had low MI and low importance, and highly collinear features were dropped. It was then followed by scaling of features with Min-Max scaler to normalize the values of inputs prior to training the models. Five lightweight classifiers which are popular in intrusion detection literature were chosen and their performance was compared using 3-fold cross-validation. Accuracy, precision, recall, F1-score, and fit time were calculated as metrics of comparison. Memory usage of the training and testing sets were also monitored to account for computational limitations.

Subsequently, two models with the best performance were chosen and analyzed in detail. The model hyperparameters were refined, and the additional assessment involved confusion matrices, classification reports, training/prediction time, feature importance plots, and statistical measures like average bias, variance, expected loss, and goodness-of-fit. To evaluate the size of the models, each model was saved into a file to determine the file size, which will provide information on how large the model is.

A robustness test was also performed by removing the most significant feature from the training set, and both models were re-trained to observe the performance changes. This was useful in establishing the model that predicts better when important information is missing. The selection of the final model was made by comparing the two best performing models and finding the most sensible trade-off among performance, file size, and computational cost.

A summary of the research workflow is described in Table 1 below:

Table 1: Overview of research workflow

Step	**Description**
Dataset selection	Downloaded CICIoT2023 dataset and used only the first 20 files out of 169 files due to hardware constraints.
Exploratory data analysis (EDA)	Explored the dataset's structure and statistics.
Label engineering	Created binary label: DoS vs non-DoS
Data preprocessing	Handled missing and unique data, split data into training and testing sets, performed class balancing.
Feature selection	Used Pearson correlation, MI scores, Random Forest importance to determine features useful for predictions.
Dimensionality reduction	Removed low-importance and highly correlated features.
Scaling	Scaled the data using Min-Max scaler before model training.
Resource monitoring	Tracked memory usage of training and testing sets.
Model selection	Imported 5 lightweight models to be evaluated.
Performance evaluation	Compared the performance of the models using cross-validation.
Final model evaluation	Selected top 2 models to conduct further assessment.

Robustness test	Removed most important feature and re-trained the top 2 models to assess their robustness.
Final model selection	Selected the final model based on performance drop, speed, and size.

Such a design helps to ensure that all steps within the workflow are made in accordance with the research goal developing efficient and accurate DoS detection models that can be used in constrained IoT settings.

3.2 Dataset Selection and Data Exploration

This section describes the dataset used in the study, and preliminary data exploration steps which have been performed. The selection of an appropriate dataset is one of the crucial steps that determine the relevance, reliability, and applicability of ML-based IDS, especially in the IoT environments.

The CICIoT2023 dataset was chosen as the primary data source for this study. After downloading the dataset, exploratory data analysis (EDA) was conducted to get insights into the structure, distribution, and nature of the data, which guided the subsequent preprocessing step.

3.2.1 CICIoT2023 Dataset Overview and Justification

The CICIoT2023 dataset was selected because it offers extensive coverage of attacks, realistic traffic of IoT devices, and it supports different classification tasks. One of the core objectives of CICIoT2023 is to close the gaps in the previous IoT intrusion detection datasets, namely, the lack of traffic produced by real IoT devices, and the lack of attacks launched within the IoT environment itself. Unlike previous datasets where the simulation of attacks was performed with regular IT systems, CICIoT2023 uses actual malicious IoT nodes (Raspberry Pi) to launch the attacks, which better reflects the threat vectors in smart environments (Neto et al., 2023). Figure 8 shows an overview of the CICIoT2023 research paper.

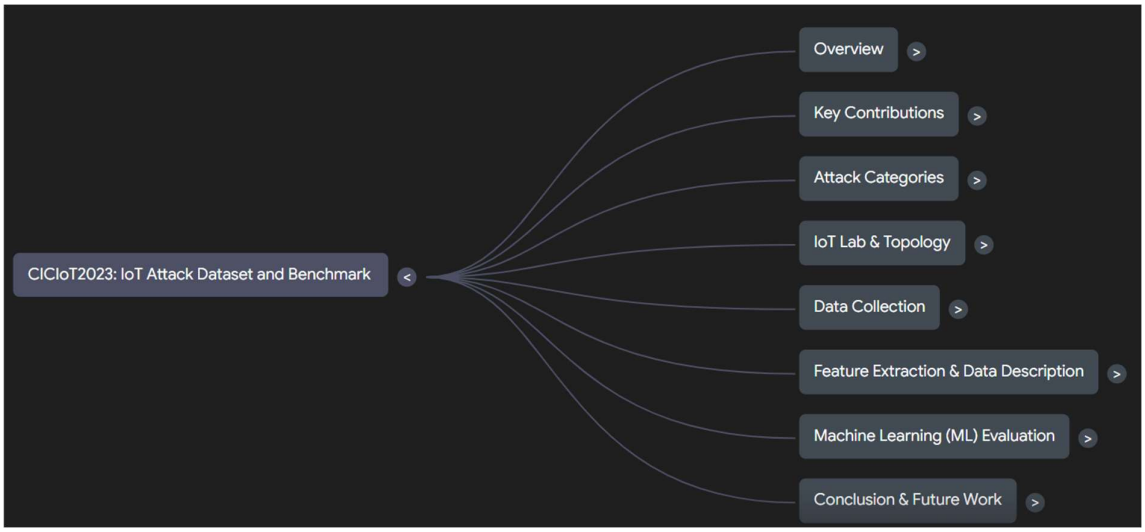

Figure 8: Overview of the CICIoT2023 research paper

This dataset was produced in a controlled laboratory environment that emulates a smart home topology comprising of 105 interconnected devices, 67 of which are used to launch attacks. The attacks were carried out using compromised IoT nodes only, which makes the dataset more relevant to the real-world IoT security studies. It consists of 33 types of attacks that are organized into 7 broad categories, including DoS, DDoS, reconnaissance, spoofing, brute force, web-based, and Mirai-based botnets, many of which are rarely covered in detail in previous datasets (Neto et al., 2023).

The research paper for CICIoT2023 covers the evaluation of three classification tasks:

- Binary classification (malicious vs benign)
- Grouped classification (8 classes, representing benign traffic and each major attack category)
- Multiclass classification (34 classes, covering benign traffic and all attack types).

The dataset paper also reports performance benchmarks that are very reliable, particularly for Random Forest and Deep Neural Networks with an accuracy of over 98% and an F1-score of over 94% in binary classification. The accuracy of the ML models in grouped and multiclass cases was also reasonable. However, the models struggled with minority classes identification, particularly in the case of less common attacks such as web-based or brute-force attacks.

The combination of practical relevance and technical robustness makes CICIoT2023 a fine choice towards constructing and evaluating lightweight, ML-based IDS in modern IoT networks.

3.2.2 Data Exploration of the Dataset

First, the CICIoT2023 dataset was downloaded into Kaggle Notebook. Due to the computational constraints of the development environment, only the first 20 files out of the total 169 CSV files were loaded and combined into a single pandas DataFrame. There is a total of 4,723,822 rows and 48 columns. This subset proved adequate in terms of the variety of attack behaviors it was able to capture, while maintaining system responsiveness during training and testing.

The initial structural analysis of the dataset was performed by inspecting the data types, completeness, and distribution of numeric features. The dataset contains both numeric and textual columns. All textual (non-numeric) features were identified, and numeric features were separated to be analyzed and preprocessed further. Summary statistics helped in understanding the distributions, range, and possible anomalies in all features.

In order to get an idea of the class distribution and evaluate the imbalance, the most common categories of attacks were explored with the help of the value_counts() function used on the label column. All samples with the DoS labels were then grouped together to determine the quantity of each DoS attack type, which were presented in a bar chart. The chart is illustrated in Figure 9.

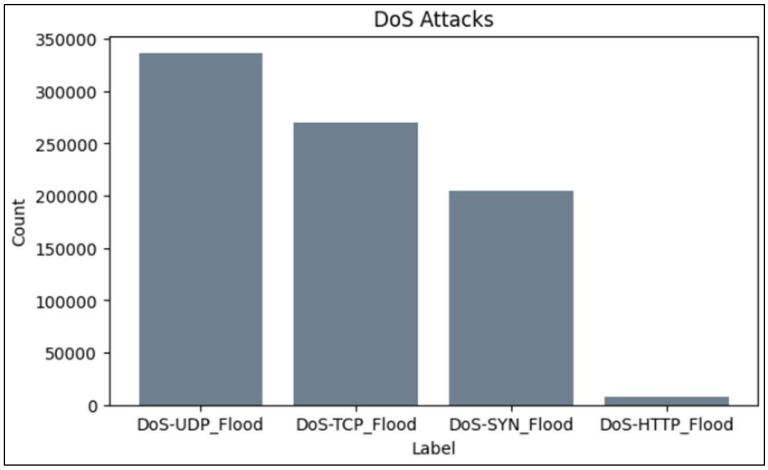

Figure 9: Sample distribution of different DoS attacks

Three out of four types of DoS attacks were generally well represented in the dataset, except the DoS-HTTP_Flood type which was severely underrepresented. Thus, it will be upsampled in a subsequent step.

3.3 Data Preprocessing

3.3.1 Binary Label Creation

In order to specifically target the detection of DoS attacks, a binary label was constructed whereby all entries that involved DoS were assigned 1 (DoS), and the rest of the entries including DDoS, web-based attacks, and normal traffic were assigned 0 (non-DoS). To determine if there is class imbalance, a visualisation on this binary label was created, as shown in Figure 10.

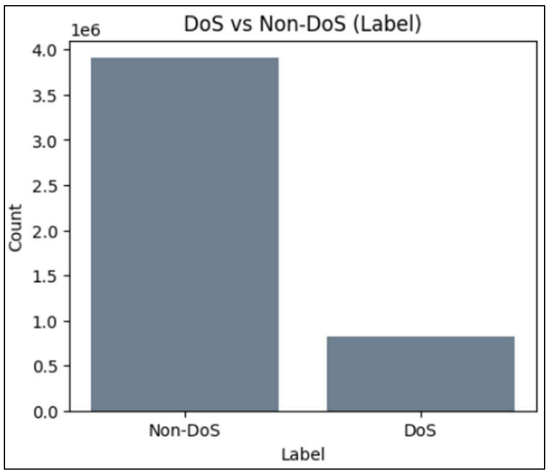

Figure 10: Distribution of DoS and non-DoS samples

As expected, the number of non-DoS samples was much higher than the number of DoS samples. Non-DoS consists of almost 4 million samples, while DoS only has 817,568 samples. This highlights the need for class balancing.

3.3.2 Handling Missing and Unique Data

A critical part of dataset preparation was the identification and removal of data anomalies in the form of missing values and non-informative features. The dataset had 47 columns initially. A column was dropped at the start since the feature was determined to be redundant and provided no useful information through manually inspecting the values and finding information about this particular feature in the dataset paper. Next, a thorough examination showed that no columns contained any missing values, and every record had a valid label, which meant that no rows had to be dropped at this point.

In order to clean the data further, columns that had very minimal variability were examined. In particular, three of the columns have been identified to have only one distinct value over all the records, making them non-informative to learn patterns. These columns were then dropped in order to minimize redundancy and to make the models more efficient.

Additionally, a check for missing numerical entries indicated that there were no missing values that needed imputation. This validation ensured that the dataset was clean and could be used in downstream processing without the need for filling or interpolation. After this stage, the dataset was left with 42 features.

3.3.3 Train-Test Split

To prepare the dataset for feature engineering, the data was split into training and testing subsets at the ratio of 80:20. This will enable the models to be trained on most of the data and still have another set to test without bias.

This split produced a training set that has approximately 3.78 million records and a testing set of 944765 records, each with 42 features. The distribution of labels was also maintained in both subsets in order to be representative. To be more precise, in the training set, 17.31% of the records were of type DoS, whereas the remaining 82.69% were non-DoS. The testing set was also of the same distribution, and this provided consistency and fairness when evaluating the model. This stratified split made sure that each subset contained proportionate number of DoS and non-DoS classes, which facilitated confidence in learning and generalization of the learned models.

3.3.4 Class Balancing

Initially, there was a significant imbalance between the classes in the training data, with non-DoS samples exceeding DoS samples by a large amount. Among the DoS attack types, the HTTP Flood type was severely underrepresented compared to the other three types.

To deal with this, the minority DoS-HTTP_Flood category was synthetically upsampled using Synthetic Minority Oversampling Technique (SMOTE) to decrease the disparity between DoS classes. Its sample size was boosted from around 5,800 to over 81,000, which is more in line with the other DoS types. This was done so that every DoS subtype is more evenly represented in the training data.

The next step was to balance the binary labels of the data by downsampling the non-DoS samples. The non-DoS samples were downsampled randomly to have the same count as the DoS samples. This created a balanced dataset with the same amount of DoS and non-DoS records, with each class having 729,967 samples.

After balancing was done, features that had no variability were then removed to enhance training and to decrease redundancy. This is done by checking whether a feature has a variance of zero.

3.4 Feature Selection and Dimensionality Reduction

In order to improve model performance and to decrease computational complexity, feature selection and dimensionality reduction methods were used. These steps are useful to remove irrelevant, redundant, or less-informative features, and make the model concentrate on the most impactful features. Various statistical and model-based methods were employed to estimate the relevance of features, and the results obtained were used to select a more refined set of features to train on.

3.4.1 Pearson Correlation

Pearson correlation coefficient measures the linear relationship between two continuous variables, and the result is a value ranging between -1 and 1. A coefficient close to 1 or -1 would imply a strong positive or negative relationship, whereas a coefficient close to 0 would imply a weak or no linear relationship. In that regard, it can assist in determining features highly correlated with the target variable, and in detecting multicollinearity between features, which may influence the stability and interpretation of the model (Sriram, 2006).

In order to analyze the linear relationships among features, correlation values were calculated between all pairs of numerical features. Having an absolute value of 0.9 and above for the coefficient was considered as a very strong correlation. Based on this threshold, 16 features were identified as having very strong correlations with at least one other feature, and thus they may be redundant. The highly correlated pairs of features were then visualized in a heatmap, as illustrated in Figure 11, to help narrow down on the possible candidates that could be dropped.

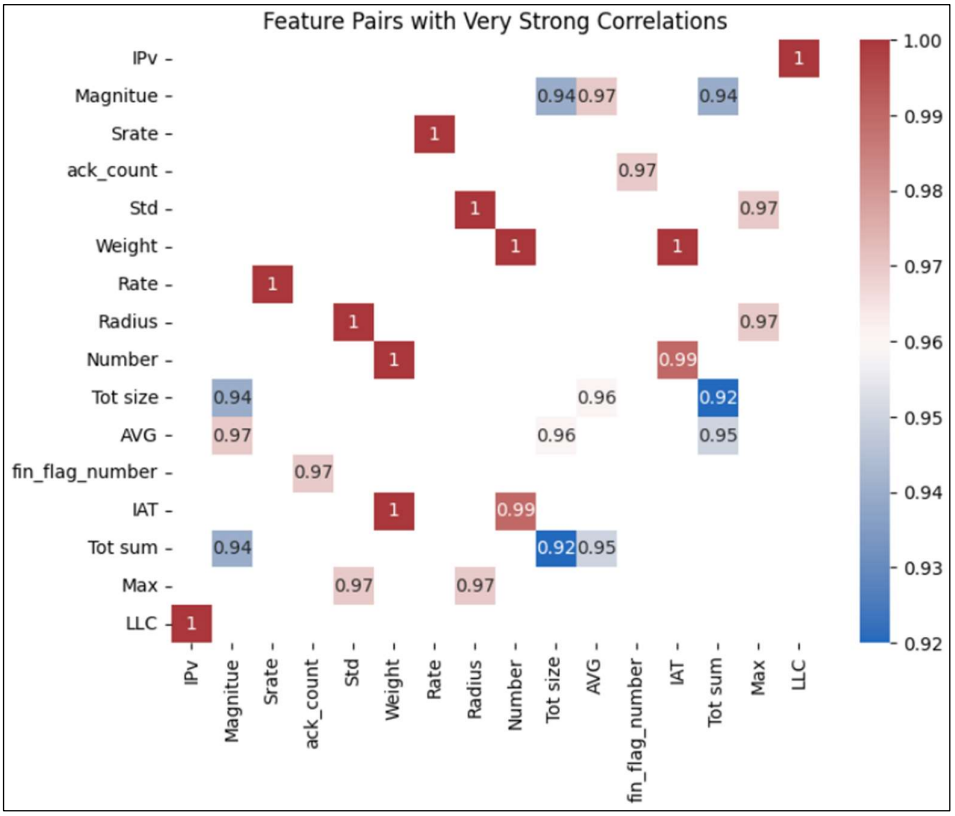

Figure 11: Feature pairs with very strong correlations

Additionally, Pearson correlation was also used to determine the linear relationship between each feature and the binary label. No feature showed a moderate correlation (absolute value 0.4 or above) with the label, indicating that the predictive ability of individual features might be weak on their own. Also, there were 32 features that have weak correlations (absolute value 0.1 or below) with the label, which confirms that it is necessary to combine several features to achieve better model results.

3.4.2 Mutual Information (MI)

MI is a statistical measure that computes the quantity of shared information between two random variables. It is very helpful in the analysis of high-dimensional data. MI can be used to gain information about the relationship and dependency between the variables by quantifying how much uncertainty regarding one variable can be reduced when knowing another variable, thus making it especially useful in feature selection and data interpretation tasks. Also, it is able to capture both linear and non-linear relationships (Falcão, 2024).

In the source code, the dependency of each feature with the label was scored and ranked. The findings showed that some features carried much more predictive information as compared to

others. These scores were visualized as a bar plot, as shown in Figure 12, making it easier to see the most informative features.

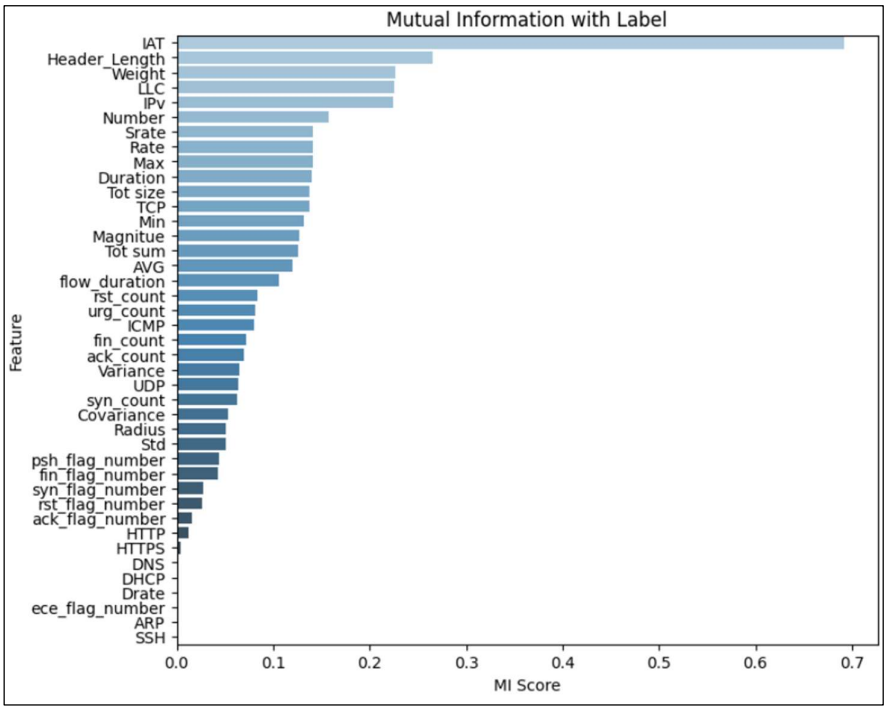

Figure 12: MI score of each feature with the label

Subsequently, the top 25 features with the highest MI scores were chosen as a concentrated set of features to be used in the dimensionality reduction step.

3.4.3 Feature Importance

Feature importance was used to rank input variables based on their contribution to predicting the target label. A tree-based method involving Random Forest classifier was used to measure the extent to which each feature decreased impurity at decision nodes. The features that contributed more to the enhancement of the decision-making process of the model were given higher importance scores.

In order to reduce the number of features, only the top 50% of features with the highest scores (greater than the median value of importance) were included. As the result, there were a total of 21 features that were added to the list of features with high importance. The chosen features were then visualized in a bar plot, and their significance was well illustrated, as shown in Figure 13. This approach assists in selecting significant features to be considered in the dimensionality reduction step.

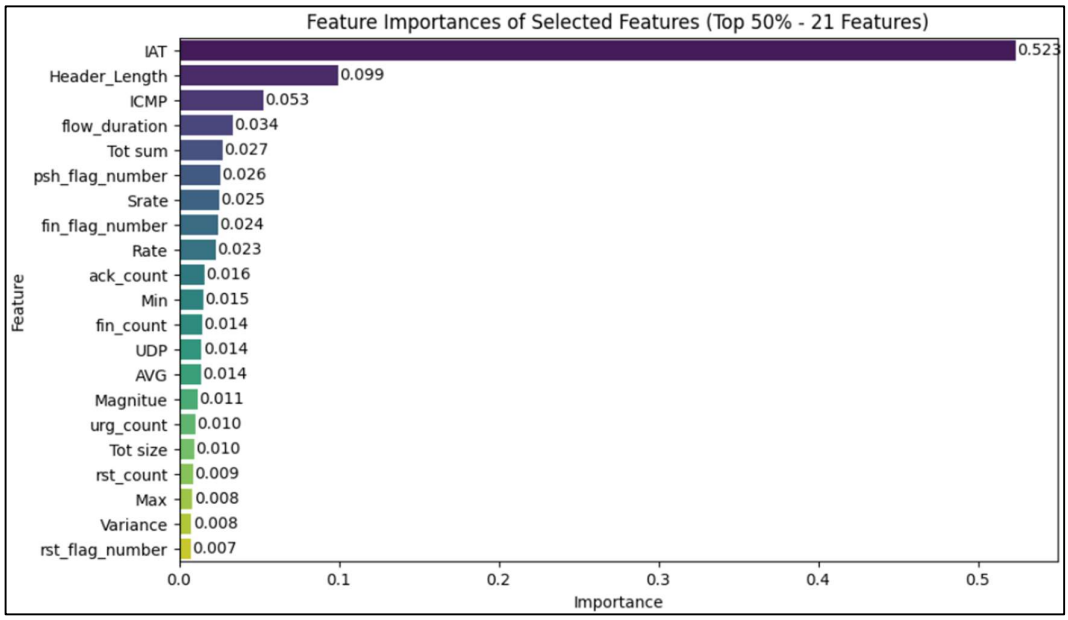

Figure 13: Features with the top 50% importance

3.4.4. Dimensionality Reduction

In order to make the models more efficient and less complex, dimensionality reduction was conducted to drop less informative and redundant features. The reduction process was done based on two major criteria:

1. Removing the features that were not among the top performers in both MI and feature importance.
2. Removing the features that were highly correlated with another feature, keeping only the more informative one.

This process led to the removal of more than half of the initial features, decreasing the number of features in the data from 41 to 20. In addition, a simple Logistic Regression model was implemented to measure performance before and after feature removal in order to determine the effect of this reduction. The results collected are provided in Table 2 below.

Table 2: Performance of Logistic Regression before and after dimensionality reduction

Period	Time taken (s)	Accuracy	Precision	Recall	F1-Score
Before reduction	193.27	0.761	0.705	0.897	0.79
After reduction	102.26	0.753	0.671	0.992	0.801

Despite a slight decrease in accuracy and precision, the F1-score slightly increased, recall had a huge increase, and the total time of computation dropped significantly. This indicates that the feature set with fewer features was not only as powerful in terms of prediction, but it is also more efficient to be used in subsequent modeling steps.

3.5 Model Selection

The selection of ML models in this paper was guided by literature review, and the aim was to balance performance and resource efficiency in the context of IoT environments. Since the majority of IoT devices often have limited resources in terms of computation and memory, lightweight models were prioritized. These models can work well in real-time anomaly detection, as well as other security-related activities without causing high resource consumption. Five models were chosen, which include Decision Tree, Naïve Bayes, Logistic Regression, Light Gradient Boosting Machine (LGBM), and Linear Discriminant Analysis (LDA). The justification for selecting each model is given in Table 3 below.

Table 3: Justification of model selection

Model	Justification
Decision Tree	Preferable due to its simple implementation and decision-logic which can be handy in making real-time decisions on IoT devices. It has a structure that can be applied to edge computing situations in which interpretability and rapid response are significant (Amgbara et al., 2024).
Naïve Bayes	Commonly used in IoT applications where fast classification is required and at a low computational expense. It is probabilistic which makes it suitable in simple security monitoring especially when it comes to anomaly detection using feature distributions (Amgbara et al., 2024).
Logistic Regression	IoT systems that perform binary classification often make use of it. It can be easily incorporated and implemented in devices with limited resources, particularly when the model has to be updated regularly (Amgbara et al., 2024).
LGBM	Although more complicated than the other models, it is included because of its flexibility in real-time systems and the potential to work with large scale data in an efficient manner. It is especially useful in a layered IoT

	security system where some components are able to use more resources (Farfoura et al., 2025).
LDA	Appropriate in cases where minimal memory and CPU usage are vital. It is used frequently in lightweight classification pipelines of embedded devices or very low-capability sensors (Farfoura et al., 2025).

These models offer diverse implementation options that conform to numerous IoT limitations and deployment approaches, allowing this study to investigate both low-resource and moderately demanding alternatives to detect anomalies in network traffic.

3.6 Model Training and Validation

3.6.1 Pre-Training

After the preprocessing stage, the training set had 1,459,934 samples and the testing set had 944,765 samples, which gave an approximate split ratio of 60:40, as illustrated in Figure 14. Both sets contained 20 features.

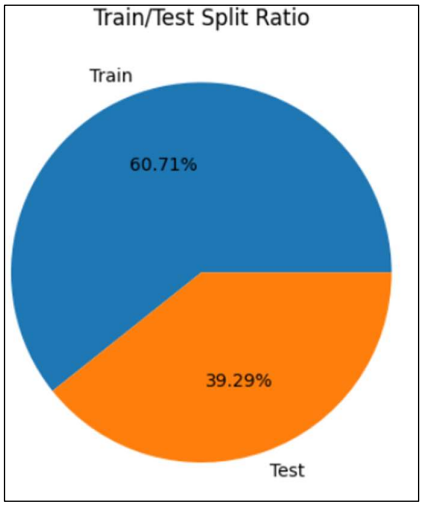

Figure 14: Ratio of train-test split

Additionally, the training labels were well-balanced with the number of non-DoS (0) and DoS (1) instances being equal, both having 50% of the training samples. Such a balance is important in training models that are not biased. In contrast, the testing set was more realistic, having a distribution of about 82.7% non-DoS traffic and 17.3% DoS traffic. This skewed test set is useful as it assists in assessing the model performance in real-life situations where the instances of attacks are comparatively low.

3.6.2 Initial Comparison of All Selected Models

To determine the relative effectiveness of the chosen lightweight machine learning models, a baseline comparison was done using the training dataset. The models were tested with 3-fold cross-validation and were measured with four widely used classification measures, including accuracy, precision, recall, and F1-score. These measures provide a good overview of the effectiveness of each model in identifying both DoS and non-DoS cases.

Besides performance metrics, the average training time (fit time) was also captured to take into consideration the computational efficiency, which is critical when it comes to the deployment in resource-limited IoT settings. The main goal of such a comparison was to determine the two best-performing models, which will be further tested, validated, and analysed.

3.6.3 Further Assessment of Two Best Models

After the initial comparison of all the selected models, the top two models that have the best performance were selected to be compared in more detail. The emphasis was placed on evaluating their runtime properties, model behavior, and feature dependencies, particularly in situations that are indicative of constrained environments like the IoT systems.

Single-threaded execution was imposed in both models as a way to simulate a more limited operating environment, like the ones commonly found in IoT systems. The next step was to train each model on the entire training data and evaluate its predictive performance on the testing set. In order to ensure consistency and fair comparison, both models were trained with the same configuration and on the same scaled features.

The key evaluation steps are as follows:

1. The total training time and prediction time were measured to evaluate the model efficiency and responsiveness. The prediction time included the time taken to predict the entire test set and a smaller batch of 1000 samples.
2. Produced classification reports to summarize standard performance metrics, including accuracy, precision, recall, and F1-score.
3. Generated confusion matrices to provide a graphical representation of the distribution of prediction among actual classes.
4. Performed bias-variance decomposition to compute bias, variance and expected loss. This allows a better understanding of the generalization behavior of each model.

5. Identified the most significant feature according to the feature importance score, and visualised the top 10 most impactful features of each model.

Lastly, all the trained models were serialized and stored in a file to be deployed or used again. The size of the saved file was considered as an indicator of the model's memory footprint, which is a major consideration when deploying it in resource constrained environments.

3.6.4 Final Model Validation

In order to validate the stability and feasibility of the two best models, a robustness test was performed. This was done by identifying and dropping the most significant feature in the training set and retraining both models on the altered training set. The aim was to assess how each model responds to the lack of key input, and whether it is able to maintain its performance in such situations. This assisted in measuring the dependency of each model to particular features and their ability to generalize when presented with incomplete or poor input data.

After the robustness test, the final model was chosen based on a combination of factors. Instead of basing the evaluation purely on predictive performance, practical deployment factors common to real-world conditions were taken into account, especially in a low-resource system like IoT devices. Three important criteria were considered:

- **Resilience to feature removal**: The extent of performance drop when the most significant feature is removed.
- **Computational efficiency**: Reflected in the training and prediction times.
- **Model footprint**: Estimated based on the size of the serialised model file.

Through the comparison of these factors, the final model was chosen as the model that provided the most balanced tradeoff between accuracy, robustness, and cost of operation.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter reports and discusses the results of the experiment in measuring the performance of the selected lightweight ML models in detecting DoS attacks on IoT network environments. The main goal is to find models that will not only be able to effectively distinguish between DoS and non-DoS traffic, but will also be able to work within the capabilities of resource-constrained IoT devices and can therefore be used in real-world anomaly detection tasks.

The chapter is an essential part of the thesis because it validates the modelling approach suggested in previous sections. Through the analysis of model behaviour on a balanced and realistic dataset, the chapter supports the central thesis objective, which is to suggest an effective and feasible detection system of DoS attacks in IoT networks. The findings give an indication of the generalisation ability of each of the models, their resource requirements, and flexibility in the case of missing a key feature, which are all critical in IoT cybersecurity implementations.

The chapter starts with the description of metrics that were used to evaluate the model performance. It then continues with the process of comparing all the chosen models in order to narrow down to the two most promising ones. The two shortlisted models are then assessed in detail, with an analysis of the effectiveness of each model in detecting DoS attacks. Then, a feature removal test is carried out to investigate the effect of omitting powerful attributes. Next, the memory usage and saved model size are also assessed to decide on the feasibility of each model to be deployed in a low-resource environment. The chapter ends with a selection of a final model that provides the optimal trade-off between detection accuracy, computation efficiency and practicality for real-time DoS detection in IoT environments.

4.1 Evaluation Metrics

In order to evaluate the ML models in terms of their ability to detect DoS attacks in IoT networks, four primary evaluation metrics were used, including accuracy, precision, recall, and the F1-score. These measures have been selected because they give a balanced perspective of classification performance, especially in cases where misclassification like false positive or false negative can have a severe impact on the reliability of the system (Farfoura et al., 2025).

Other statistical measures such as bias, variance, expected loss and goodness-of-fit were also determined. Having both sets of metrics will not only provide an evaluation of the predictive capability of the model, but also provide information about the model generalization capability and learning behavior.

Table 4 below gives a summary of the metrics used, including their descriptions and relevancy to the task of DoS detection.

Table 4: Evaluation metrics used

Metric	**Description**	**Relevance to DoS Detection in IoT**
Accuracy	Proportion of total correct predictions over all predictions made.	Gives a general view of the model effectiveness in predicting both DoS and non-DoS classes.
Precision	Proportion of predicted positives that are actually positive.	Useful in minimizing false alerts, which may overwhelm security monitoring systems.
Recall	Proportion of actual positives that were correctly identified.	Important in identification of every attack case to minimize threat evasion.
F1-score	Harmonic mean of precision and recall.	Balances the performance of detection, which is generally effective when dealing with an imbalanced dataset in which positive cases (DoS) are underrepresented.
Bias	Measures the error due to the model's simplifying assumption, where high bias indicates underfitting.	Low bias is preferred so that the model could capture the patterns of the DoS attack behaviors.
Variance	Measures the model's sensitivity to fluctuations in the training set, where high variance indicates overfitting.	Stability among training samples is critical to ensuring consistency of detection performance in IoT dynamic environments.

Expected loss	The average classification error over multiple rounds.	Represents the general model reliability which is a combination of bias and variance.
Goodness-of-fit	Reflects how well the model fits the data overall, its calculated as 1 − Expected Loss	The greater the values, the better the model generalizes to unseen IoT traffic, which is important in real-world deployment.

This evaluation framework allows choosing models that not only detect DoS attacks but also remain reliable in practice because it integrates both predictive and generalization measures. This is necessary because of the limited resources and the dynamism of IoT networks.

4.2 Initial Model Comparison

4.2.1 Model Implementation

This section shows the initial comparison between five lightweight ML models that have been chosen to detect DoS attacks. The models selected included Light Gradient Boosting Machine (LGBM), Naïve Bayes, Decision Tree, Linear Discriminant Analysis (LDA), and Logistic Regression. These models were selected because of their applicability in resource-limited environments, as explained in the methodology.

Table 5 below summarizes the hyperparameters used for each model during this evaluation. Lightweight settings were specifically chosen to maintain efficiency on memory and processing, while allowing effective detection of the attacks. In addition, the 'random_state' was fixed for non-deterministic models to ensure reproducibility of results.

Table 5: Initial models and their hyperparameters

Model	Hyperparameters	Justification
Decision Tree	max_depth=5, random_state=30	Shallow depth ensures low memory usage and reduces overfitting.
LGBM	max_depth=5, num_leaves=32, n_jobs=1, verbose=-1, random_state=30	Runs on a single thread for minimal memory usage, and limits model complexity for faster runtime and better generalization,
Logistic Regression	C=0.1, max_iter=1000, random_state=30	Lower C value indicates stronger regularization that reduces overfitting, and

		increased number of iterations to ensure convergence.
LDA	Default	Simple model, no tuning needed.
Naïve Bayes	Default	Lightweight by nature, no tuning needed.

4.2.2 Model Evaluation

The same training set with 1,459,934 samples was used to train each model and 3-fold cross-validation was used to evaluate each model with accuracy, precision, recall, and F1-score as the evaluation measures. This comparison is aimed at determining the two best models in terms of their predictive performance to be used in further analysis and optimization. The results of the cross-validation are summarized in Table 6.

Table 6: Performance comparison of initial models

Model	Accuracy	Precision	Recall	F1-Score	Fit Time (s)
Decision Tree	0.99971	0.99981	0.99960	0.99971	2.690
LGBM	0.99967	0.99953	0.99980	0.99967	11.357
Logistic Regression	0.75305	0.67117	0.99223	0.80071	33.463
LDA	0.74835	0.66691	0.99230	0.79770	2.035
Naïve Bayes	0.74139	0.66359	0.97941	0.79113	0.603

Out of the five models that were tested, Decision Tree and LGBM were the most superior in terms of all performance measures, making them excellent in classifying DoS and non-DoS traffic.

Decision Tree classifier demonstrated the best overall accuracy (0.99971) and F1-score (0.99971), which means that the balance between precision and recall was very stable. It also had a reasonable training time of 2.69 seconds, which means it can be considered efficient and suitable for low-resource settings common in IoT deployments. Its low computational overhead and very high interpretability makes it an attractive candidate as a practical DoS detection model.

Right behind it was the LGBM classifier, which did not perform as well in accuracy (0.99967), but higher in recall (0.99980), which indicates better sensitivity to detecting DoS attack cases.

Even though it took longer to train (11.36 seconds), this is reasonable in light of its better predictive performance and capability of working with large-scale data using gradient boosting. The strength and scalability of LGBM allow it to be especially applicable in the real-time intrusion detection frameworks, where accuracy is vital (Farfoura et al., 2025).

On the contrary, the other models, namely Logistic Regression, LDA and Naive Bayes, had significantly lower accuracy (below 0.76), although their recall was high. This means that even though these models could identify the majority of attack cases, they still had a high number of false positives, which are indicated in their relatively low precision values. Moreover, Logistic Regression used more than 33 seconds to train, which is not quite appealing in terms of computational efficiency.

4.2.3 Two Best Models Selected

Since DoS attack detection should not only be accurate but also efficient in IoT settings, Decision Tree and LGBM were selected as the most appropriate models that will be further evaluated. These two classifiers did not only show almost perfect detection performance in all major evaluation metrics, but they also kept the training times reasonable, which makes them viable in real-time or near real-time applications.

In addition, they are scalable and flexible and thus can be used on edge devices with limited memory capacity and processing capability. Decision Tree is simple, easy to interpret and has low overhead, whereas LGBM has a better generalization and robustness in working with large and imbalanced data (Amgbara et al., 2024; Farfoura et al., 2025). Therefore, these models offer an all-round balance between effectiveness, efficiency and deployment readiness, which are worth examining in further sections.

4.3 Detailed Evaluation of Selected Models

After the preliminary analysis, Decision Tree and LGBM were chosen to be evaluated in detail due to their good performance and applicability to IoT-based DoS attacks detection. In this section, the practical feasibility of each model will be explored by in-depth analysis of the run-time efficiency, classification performance, dependency on features, and resource usage. Major considerations are training and prediction times, decomposition of bias and variance, feature importance analysis, and the size of the model file, which gives a complete picture of their advantages and disadvantages in limited scenarios.

Both models used the same sets of training and testing data, where the training set contained 1,459,934 samples, and the testing set contained 944,765 samples. Subsections 4.3.1 and 4.3.2 present the findings for Decision Tree and LGBM respectively. Both models were evaluated with a same structure, which includes calculating training and prediction time, generating classification performance and statistical measures, and identifying top important features.

4.3.1 Decision Tree

4.3.1.1 Training and Prediction Time

Table 7: Training and prediction times of Decision Tree

Task	Time Taken (s)
Training (all samples)	5.1516
Prediction (all samples)	0.1630
Prediction (1000 samples)	0.0035

The Decision Tree model proved to be very efficient in terms of the runtime. The model was trained on the whole dataset of more than 1.4 million samples in about 5.15 seconds, which is indicative of the lightweight nature of the model and its applicability to fast retraining if there is a need. The whole test set (944,765 samples) was predicted in 0.16 seconds, and the inference of a smaller batch of 1,000 samples took only 0.0035 seconds.

4.3.1.2 Classification Performance

Table 8: Classification performance of Decision Tree

Metric	Score (%)
Accuracy	99.98
Precision	99.96
Recall	99.98
F1-score	99.97

Based on the classification performance, the model was performing exceptionally well with all the values of Accuracy, Precision, Recall and F1-score being above 99.9%. This shows that the model is very capable of differentiating DoS and non-DoS traffic accurately with few false positives and false negatives. These high scores on all metrics indicate that the model

generalizes well and has balanced detection ability, which is a critical parameter of reliable intrusion detection in critical systems.

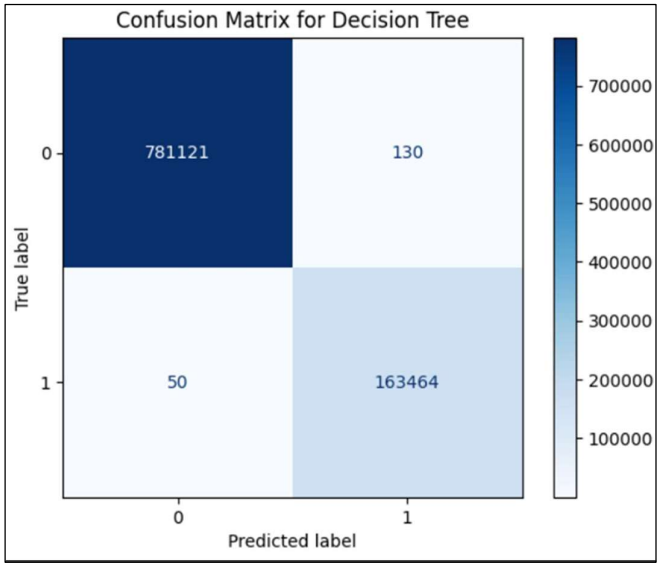

Figure 15: Confusion matrix for Decision Tree

The Decision Tree model has a high classification ability which is evident in its confusion matrix. The model correctly classified 781,121 non-DoS instances (True Negatives) and 163, 464 DoS instances (True Positives) out of the 944,765 test samples. It had 130 false positive (FP) predictions, in which non-DoS traffic was wrongly detected as DoS, and 50 false negatives (FN) that missed some actual attacks. Such low values of FP and FN indicate that the model is not only precise but also sensitive, meaning that it will not generate many false alarms and will detect the threats well, which is particularly important in the case of real-time DoS detection.

4.3.1.3 Statistical Measures

Table 9: Statistical measures of Decision Tree

Metric	Score (%)
Bias	0.022
Variance	0.008
Expected loss	0.024
Goodness-of-fit	99.98

The statistical indicators of Decision Tree model show a balanced and robust performance. The low bias (0.022%) implies that the model grasps the patterns in the data very well, and the low variance (0.008%) implies that the model works similarly across various data sets, which decreases the chances of overfitting. The expected loss (2.4%) indicates that there is a low degree of misclassification. Also, the goodness-of-fit of 99.98% confirms that the model fits the real distribution of the data very well, so it can be a plausible choice when it comes to identifying DoS attacks in a real-life application.

4.3.1.4 Important Features

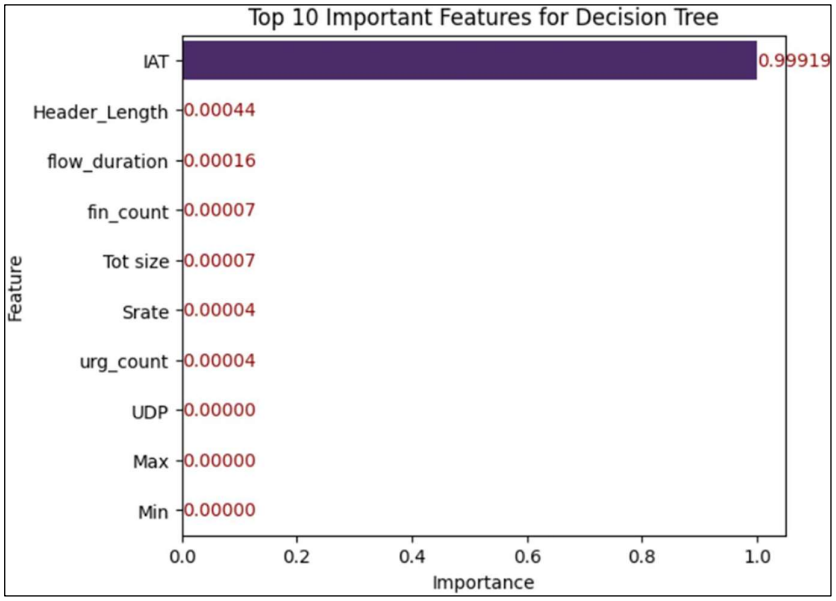

Figure 16: Top important features for Decision Tree

The feature importance scores of the Decision Tree model indicate that it is heavily reliant on one feature, which is inter-arrival time (IAT), since this feature has an extreme importance value of 0.99919. This implies that this feature is almost entirely responsible of the decision-making done by the model, whereas the rest of the features have little to no significance, as the second most significant feature only has a score of 0.00044. There were only 7 features with non-zero importance values, which implies that the model is very selective. Although this simplicity is useful in terms of efficiency and interpretability, it also causes some concern over the possibility of overfitting to one particular feature, and thus lowering generalizability in more varied or noisy environments.

4.3.2 LGBM

4.3.2.1 Training and Prediction Time

Table 10: Training and prediction times of LGBM

Task	Time Taken (s)
Training (all samples)	16.2695
Prediction (all samples)	2.9422
Prediction (1000 samples)	0.0061

The LGBM model took around 16.27 seconds to train on the training set, which is significant compared to Decision Tree. It also had a longer prediction time of approximately 2.94 seconds to run the entire test set and 0.0061 seconds to run 1,000 samples. Although LGBM is computationally more demanding, the increased time can be explained by its more complicated ensemble architecture that usually leads to improved generalization and robustness. These findings suggest that LGBM can be still deployed, particularly in cases where a little longer processing time is acceptable in favor of better learning ability.

4.3.2.2 Classification Performance

Table 11: Classification performance of LGBM

Metric	Score (%)
Accuracy	99.96
Precision	99.90
Recall	99.97
F1-score	99.93

LGBM model demonstrated high classification scores in all the main measures, including accuracy (99.96%) and recall (99.97%), showing that it is very effective in detecting DoS attacks. The precision of 99.90% implies that the false positive rate is very low, and most of the flagged attacks were actual DoS instances. The F1-score of 99.93% indicates a strong balance between the precision and recall, which proves the general effectiveness of the model to classify both DoS and non-DoS traffic.

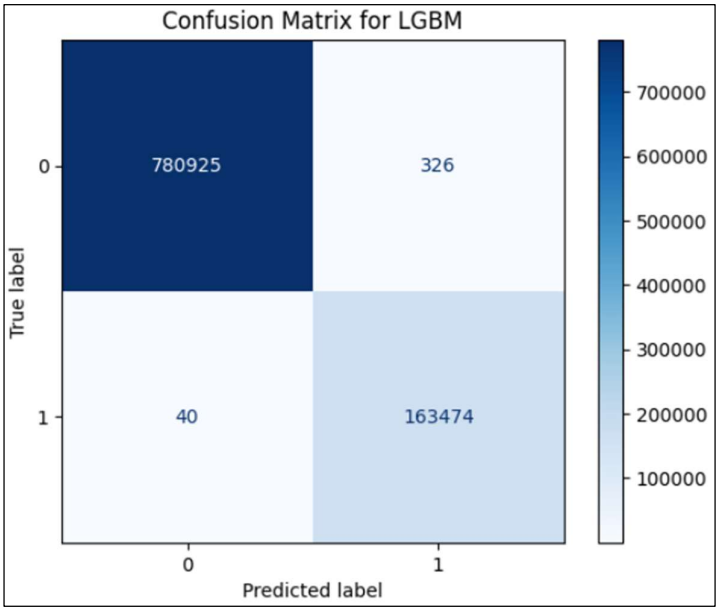

Figure 17: Confusion matrix for LGBM

The confusion matrix for LGBM model shows a robust classification result, as 163,474 true positives and 780,925 true negatives were observed, which means that the model is very accurate in classifying both DoS and non-DoS cases. The number of false negatives is only 40, which implies that very few real attacks were left undetected, which is a crucial quality of DoS detection. The number of false positives (326) is higher than the Decision Tree model, though the value is still relatively low in comparison to the size of the data set. This trade-off could be acceptable in numerous security-oriented applications where the primary goal is to reduce the number of missed attacks.

4.3.2.3 Statistical Measures

Table 12: Statistical measures of LGBM

Metric	Score (%)
Bias	0.037
Variance	0.008
Expected loss	0.039
Goodness-of-fit	99.96

The statistical evaluation of the LGBM model indicates that it has a low bias (0.037%) and variance (0.008%), implying that the model will not overfit or underfit training data. Its high predictive accuracy is also confirmed by the expected loss of 0.039%. Moreover, the goodness-

of-fit score of 99.96% also demonstrates that the model is closely matched with the real distribution of data, which further confirms its effectiveness in terms of real-life DoS detection tasks.

4.3.2.4 Important Features

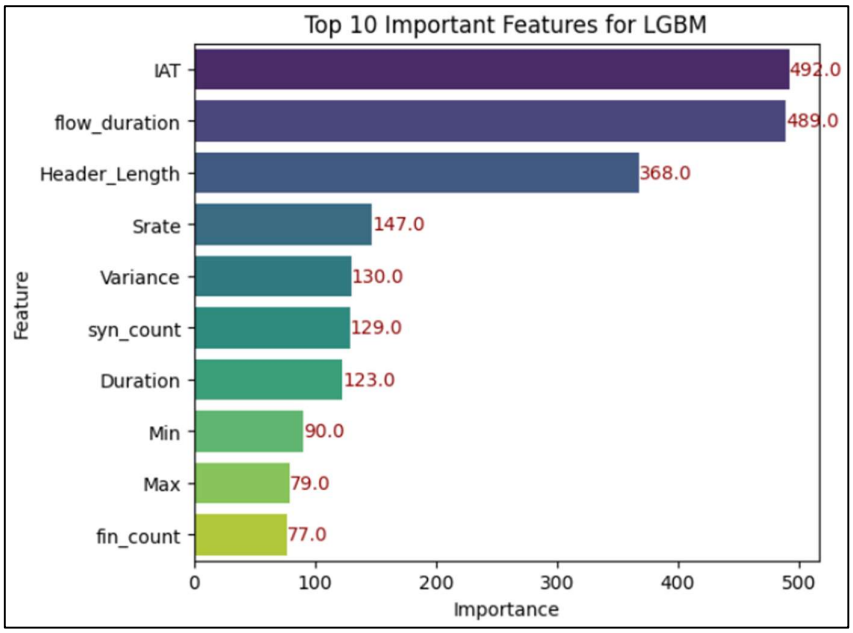

Figure 18: Top important features for LGBM

The feature importance graph of the LGBM model shows a more even distribution of influence among several features as opposed to the Decision Tree. Although IAT and flow_duration are the most prominent features by having importance values of around 490, some other features such as Header_Length, Srate, Variance, and syn_count also have a significant contribution. This implies that LGBM uses a wider range of features to make predictions, which can improve its resilience and flexibility to the changes in DoS traffic patterns.

4.4 Feature Removal Test

In order to further evaluate model robustness and dependence on major input features, a feature removal test was performed by removing the most important feature from the dataset, which is the inter-arrival time (IAT) feature. Both the Decision Tree and LGBM models were re-trained on the modified features set. The performance changes before and after the feature removal were then evaluated.

To determine a baseline performance, each of the two models was tested with all features before the top feature (IAT) was removed. This comparison was done in three primary aspects, which include runtime efficiency, classification performance and statistical robustness.

4.4.1 Runtime Efficiency

4.4.1.1 Before Feature Removal

Table 13: Training and testing times of both models before feature removal

Model	Time Taken for Training (all samples)	Time Taken for Prediction (all samples)	Time Taken for Prediction (1000 samples)
Decision Tree	5.1516	0.1630	0.0035
LGBM	16.2695	2.9422	0.0061

The Decision Tree was faster in training, which took about 5.15 seconds, and the prediction on the whole test set took only 0.16 seconds. It also provided quick prediction responses on 1,000 samples, using only 0.0035 seconds. In contrast, LGBM took a longer training time of 16.27 seconds, test set prediction of 2.94 seconds, and prediction on 1,000 samples of 0.0061 seconds. Although LGBM was slower, its performance was still acceptable considering that it has a more intricate ensemble structure.

4.4.1.2 After Feature Removal

Table 14: Training and testing times of both models after feature removal

Model	Time Taken for Training (all samples)	Time Taken for Prediction (all samples)	Time Taken for Prediction (1000 samples)
Decision Tree	6.2396	0.1313	0.0022
LGBM	18.3119	4.3638	0.0082

After removing the best feature, the training times of the two models increased, where the time taken by Decision Tree increased from 5.15 seconds to 6.24 seconds, and time taken by LGBM increased from 16.27 seconds to 18.31 seconds. This is because calculating splits or boosting without the most influential feature requires more effort. LGBM took a longer time to execute

full-set prediction (2.94s to 4.36s) and Decision Tree took slightly less time (0.16s to 0.13s). Decision Tree spent less time predicting 1000 samples, while LGBM took more time to predict.

4.4.2 Classification Performance

4.4.2.1 Before Feature Removal

Table 15: Classification performance of both models before feature removal

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree	99.98	99.96	99.98	99.97
LGBM	99.96	99.90	99.97	99.93

When it comes to classification performance, both models have a high score of accuracy, with the Decision Tree having 99.98% and LGBM coming in closely at 99.96%. Decision Tree was also a little bit better in terms of precision, recall, and F1-score than LGBM, which means that it could generalize slightly better without losing predictive power.

4.4.2.2 After Feature Removal

Table 16: Classification performance of both models after feature removal

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree	57.68	64.50	74.39	55.30
LGBM	84.81	74.98	84.10	77.84

The performance of the two models fell sharply after the removal of the top feature. The accuracy of Decision Tree decreased significantly to 57.68% with a significant drop in F1-score to 55.30%, which is a very poor result with regards to its predictive powers. This confirms that the model was overly reliant on IAT and did not have the ability to offset with other features.

By comparison, LGBM was more robust and had an accuracy of 84.81% and an F1-score of 77.84%. Despite the decreased performance, LGBM had a more balanced set of feature importances, thus it could still have relatively high recall and overall detection effectiveness without the dominant feature. This shows that LGBM is more flexible when dealing with feature loss.

4.4.3 Statistical Measures

4.4.3.1 Before Feature Removal

Table 17: Statistical measures of both models before feature removal

Model	Bias (%)	Variance (%)	Expected Loss (%)	Goodness-of-Fit (%)
Decision Tree	0.022	0.008	0.024	99.98
LGBM	0.037	0.008	0.039	99.96

Decision Tree had a lower bias (0.022%), expected loss (0.024%), and a greater goodness-of-fit (99.98%) than LGBM, which had a bias of 0.037% and expected loss of 0.039%. The variance was the same (0.008%) in both models, which means that the performance was similar in various data splits.

4.4.3.2 After Feature Removal

Table 18: Statistical measures of both models after feature removal

Model	Bias (%)	Variance (%)	Expected Loss (%)	Goodness-of-Fit (%)
Decision Tree	42.32	0.038	42.34	57.66
LGBM	15.01	1.15	15.19	84.81

When the top feature was eliminated, the Decision Tree model was dramatically affected with bias rising to 42.32% and expected loss to 42.34%, and a sharp decline in goodness-of-fit to the 57.66% mark. This proves that the model was extremely dependent on a single feature and was not able to generalize without it.

Conversely, LGBM showed a slower increase in both bias to 15.01% and expected loss to 15.19%, and still decent goodness-of-fit of 84.81%. Even though its variance grew to 1.15%, the overall effect on generalization was significantly lower than that of Decision Tree, which shows that LGBM is more robust to the loss of key features.

4.4.4 New Distribution of Feature Importances

To demonstrate the new distribution of feature importances following the removal of IAT, two bar charts were created to represent each model, as shown in Figure 19 and 20.

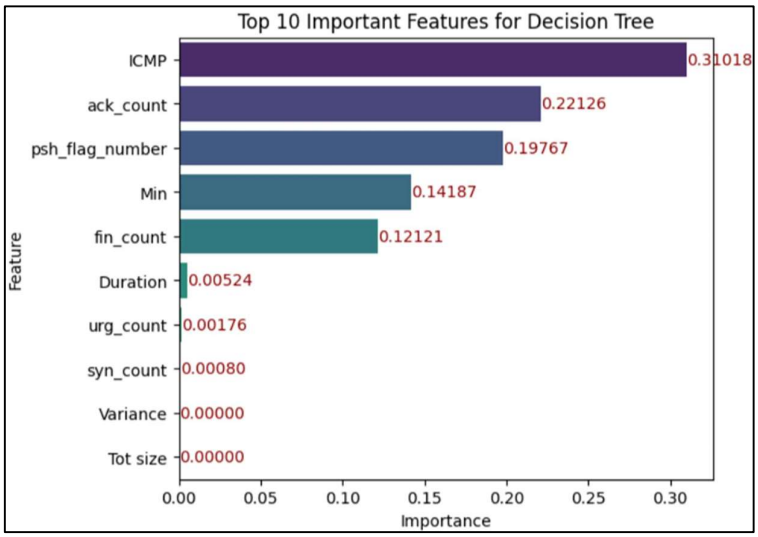

Figure 19: Top important features for Decision Tree after top feature removal

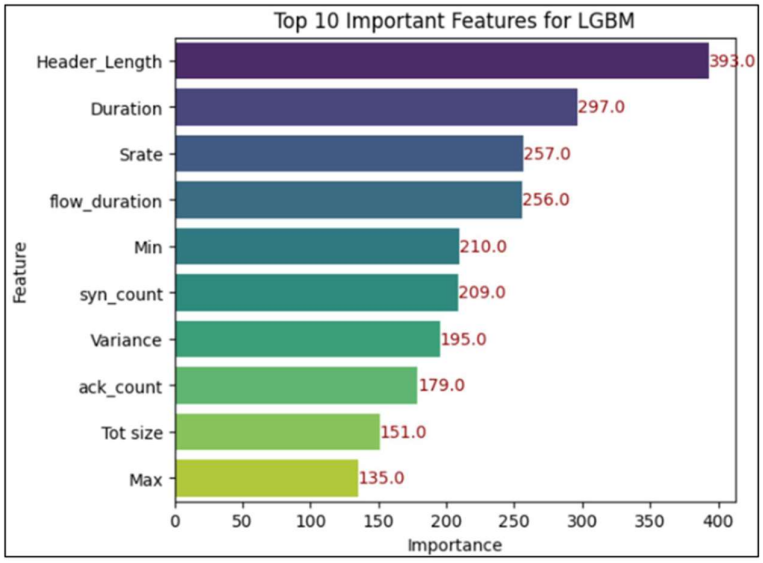

Figure 20: Top important features for LGBM after top feature removal

In the case of the Decision Tree, where initially the model was almost entirely dependent on IAT, the feature removal compelled the model to rely on other features. After removal, there were only six features with non-zero importance values. This result indicates that the model was extremely reliant on a single feature, and although it could be used without IAT, it could not make use of the wide variety of features as effectively as more versatile models like LGBM.

For the LGBM model, the findings reveal a relatively balanced dependence on a broader set of features. The most valuable features after the removal are Header_Length, Duration, Srate, and flow_duration, and then followed by Min, syn_count, Variance, and ack_count. This means that

LGBM can still use a wide variety of features to retain predictive performance, despite the lack of the IAT feature that was dominant. This kind of redundancy is useful to the model stability and flexibility, especially in dynamic or noisy environments.

4.5 Data and Model Size

In order to evaluate the training and testing data in terms of memory efficiency, the amount of memory used by both sets was recorded before and after removing the top feature. This is an important step when taking into account deployment in resource-constrained systems like IoT devices. Also, the comparison can be used to show how much memory footprint was reduced by removing a feature.

Table 19 below gives the summary of memory usage (in megabytes) of both training and testing sets in two scenarios: with all 20 features and with 19 features.

Table 19: Memory usage of training and testing sets with 20 features and 19 features

Dataset Component	**Memory Usage (MB) – 20 Features**	**Memory Usage (MB) – 19 Features**
Training Features	222.77	211.63
Testing Features	144.16	136.95
Training Labels	11.14	11.14
Testing Labels	7.21	7.21

This comparison shows that there is a clear decrease in memory consumption of feature sets when one feature is dropped, which further supports the importance of dimensionality reduction not only in terms of model performance, but also in computational efficiency.

Additionally, the file sizes of the saved model objects are provided in Table 20 below.

Table 20: File sizes of saved model objects

Model	**File Size (KB)**
Decision Tree – trained on 20 features	4
Decision Tree – trained on 19 features	5
LGBM – trained on 20 features	296
LGBM – trained on 19 features	328

The sizes of the saved model files indicate the internal complexity and the structure of each ML algorithm. The Decision Tree models are very lightweight, with both files having 4 KB and 5 KB respectively. This implies that the model is very simple with a few nodes and parameters. On the other hand, the LGBM models are much bigger with file sizes of 296 KB and 328 KB respectively. That is understandable because LGBM uses multi-boosted trees, which adds more parameters and increases the complexity of the model structure (Farfoura et al., 2025).

The fact that file sizes of both models increased when the IAT feature was removed is an indication that the model depth was slightly increased or the models needed other features to sustain performance. In general, Decision Tree models are well suited to be used in storage-constrained environments, whereas LGBM models provide a greater level of complexity but with a reasonable memory trade-off.

4.6 Selection of a Final Model

When choosing the most suitable model for real-time DoS detection in IoT environments, it is necessary to find the optimal balance between accuracy of detection, computational power, and feasibility, especially in resource-poor environments.

Based on the experiments, LGBM showed higher resilience and generalization capacity, particularly in situations where an important feature was absent. Although Decision Tree performed slightly better in the first full-feature evaluation, its overdependence on one feature resulted in a catastrophic performance loss after removing the feature. Its accuracy declined by 42.3%, and its F1-score dropped 44.67% . On the contrary, LGBM preserved a more solid performance, with 84.81% accuracy and 77.84% F1-score after the loss of the best feature, thanks to its more distributed dependency on input variables.

Although LGBM is slower to train (16.27s vs. 5.15s for Decision Tree) and uses more memory (model file sizes of 296, 328 KB vs. 4, 5 KB for Decision Tree) compared to Decision Tree, this trade-off is justified as LGBM is much more robust in detection and less biased, as well as having a higher goodness-of-fit when input is degraded. Besides, its expected loss was also low (15.19%) after removing the top feature, as opposed to 42.34% loss of the Decision Tree.

In real-time IoT applications, where the detection of malicious activity is essential, but the device may not have much processing power and memory, LGBM provides a fair compromise.

It is not as lightweight as Decision Tree, but its performance during runtime, especially using 0.0061 seconds to process 1,000 samples on the full feature set, is well within the realms of acceptable values when considering real-time application. On top of that, the slightly higher memory consumption is compensated by the robustness, flexibility, and detection reliability of LGBM, which is essential in dynamic or adversarial settings.

Thus, taking into account all the aspects, such as detection accuracy, run time efficiency, memory consumption, and robustness in case of feature loss, LGBM is chosen as a final model to be used in DoS detection within IoT networks. It offers the best compromise between predictive performance and feasibility, which guarantees effective protection without affecting responsiveness.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

The project introduced a lightweight ML framework, which was developed and tested to detect DoS attacks in IoT network traffic. The study was conducted using the CICIoT2023 dataset that represents realistic IoT traffic and attack scenarios, and was able to demonstrate a complete ML pipeline in a low-resource environment to overcome the shortcomings of classic IDS.

An extensive literature review was done to present the basis of the study. It discussed the weaknesses of IoT networks, the nature of DoS attacks and shortcomings of current IDS. Several lightweight ML models and frequently-used IoT security datasets were also discussed in the review. This served as the theoretical basis of model selection, preprocessing techniques and evaluation techniques, so that the framework was practical and in line with the current research trends.

The experiment started with extensive data preprocessing steps that consisted of generating binary labels, balancing classes with SMOTE and undersampling, and discarding redundant or low-variance features. Correlation analysis, mutual information scores, and feature importance based on a Random Forest model helped to select features and eventually narrowed the dataset down to a set of 20 features that still demonstrated good predictive power.

Cross-validation was used to train and compare 5 lightweight models: Decision Tree, Logistic Regression, Naïve Bayes, LGBM and LDA. Decision Tree and LGBM were chosen to be evaluated further based on their performance measures and computational efficiency. The two models showed good performance at the beginning with accuracy levels of more than 99%. However, as soon as the most significant feature was removed, the performance of the Decision Tree fell dramatically, having only 57.68% accuracy, whereas LGBM was still stable with more than 84% accuracy. Although LGBM had a bigger file size, it was not affected by feature loss as much and it still had good predictive performance.

Overall, LGBM was identified as the most appropriate model to be used in real-time DoS detection in IoT environments, since it offers the best trade-off between accuracy, robustness, and resource consumption.

5.1 Alignment with Research Objectives

The research was informed by four research objectives (ROs) aimed at assessing lightweight ML models to detect DoS attacks in IoT networks. Table 21 below summarizes how each objective was successfully addressed.

Table 21: Achievement of research objectives (ROs)

Research Objective (RO)	Achievement Summary
RO1: To preprocess and explore the CICIoT2023 dataset by separating and labelling DoS traffic for binary classification.	The dataset was cleaned, split, and transformed into a binary classification task (DoS vs. non-DoS). Class imbalance was addressed using SMOTE and undersampling.
RO2: To apply and assess feature selection and reduction techniques for improving model performance.	Redundant features were removed using Pearson correlation, mutual information, and Random Forest importance, reducing 41 features to 20 and improving runtime without sacrificing accuracy.
RO3: To compare multiple lightweight machine learning models based on their performance metrics.	Five models were evaluated, with Decision Tree and LGBM selected based on their high performance and efficiency.
RO4: To evaluate the robustness of the top-performing models by analyzing the impact of removing the most important feature on their performance.	LGBM demonstrated higher robustness by maintaining reasonable performance after the top feature was removed, unlike Decision Tree which dropped sharply.

The objectives of the research were achieved completely, and the results confirm the efficiency of optimized lightweight ML models, especially LGBM, as a solution to effective, efficient, and robust DoS detection in IoT environments.

5.2 Recommendations for Future Work

Based on the insights that were obtained in this project, a few recommendations can be made to further extend the practicality and scope of research in lightweight DoS detection in IoT systems. The recommendations include:

- **Real-time integration and deployment**: Future work should aim to deploy the proposed lightweight ML framework onto actual IoT hardware to evaluate its real-time performance, latency, and scalability in live environments. This would confirm its applicability in the real-world intrusion detection applications.
- **Expanded attack coverage**: Although current study only covered four types of DoS attacks, future research should broaden the scope to include more diverse and complex attacks such as DDoS, ransomware, and malware, which are highly relevant in IoT networks.
- **Further model optimization**: Further research on model optimization strategies, such as pruning, quantizing, or knowledge distillation, would potentially reduce the resource consumption. This helps to make models more compatible with low-power or embedded IoT devices.
- **Hardware-level implementation**: Future studies can explore the implementation of lightweight IDS mechanisms at the hardware or firmware level, which would offer on-device protection without requiring external systems or cloud-based infrastructure.

These recommendations aim to enhance the overall resilience, generalization, and practical application of DoS detection mechanisms in various IoT scenarios.

REFERENCES

Abdulla, A. R., & Jameel, N. G. M. (2023). A review on iot intrusion detection systems using supervised machine learning: techniques, datasets, and algorithms. *UHD Journal of Science and Technology, 7*(1), 53-65. https://doi.org/10.21928/uhdjst.v7n1y2023.pp53-65

Abdul-Ghani, H. A., & Konstantas, D. (2019). A comprehensive study of security and privacy guidelines, threats, and countermeasures: an iot perspective. *Journal of Sensor and Actuator Networks, 8*(2), 22. https://doi.org/10.3390/jsan8020022

Agbedanu, P. R., Musabe, R., Rwigema, J., & Gatare, I. (2022). Using incremental ensemble learning techniques to design portable intrusion detection for computationally constraint systems. *International Journal of Advanced Computer Science and Applications, 13*(11). https://doi.org/10.14569/ijacsa.2022.0131104

Alabdulwahab, S., Kim, Y., & Son, Y. (2024). Privacy-preserving synthetic data generation method for iot-sensor network ids using ctgan. *Sensors, 24*(22), 7389. https://doi.org/10.3390/s24227389

Alasmari, R., & Alhogail, A. (2024). Protecting smart-home iot devices from mqtt attacks: an empirical study of ml-based ids. *IEEE Access, 12*, 25993-26004. https://doi.org/10.1109/access.2024.3367113

Alawsi, W. A. (2023). Intrusion detection in iot networks using machine learning techniques. *International Journal of Computers and Informatics, 2*(8), 9-33. https://doi.org/10.59992/ijci.2023.v2n8p1

Almeghlef, S. M., Alghamdi, A., Ramzan, M. S., & Ragab, M. (2023). Machine learning-based dos amplification attack detection against constrained application protocol. *Applied Sciences, 13*(13), 7391. https://doi.org/10.3390/app13137391

Alsulaiman, L., & Al-Ahmadi, S. (2021). Performance evaluation of machine learning

techniques for dos detection in wireless sensor network. *arXiv preprint arXiv:2104.01963.*

Alsulami, R., Alqarni, B., Alshomrani, R., Mashat, F., & Gazdar, T. (2023). Iot protocol-enabled ids based on machine learning. *Engineering, Technology & Applied Science Research, 13*(6), 12373-12380. https://doi.org/10.48084/etasr.6421

Alsulami, A. A., Al-Haija, Q. A., & Tayeb, A. (2022). Anomaly-based intrusion detection system for iot networks with improved data engineering. *Applied Sciences, 10*. https://doi.org/10.20944/preprints202210.0431.v1

Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly detection ids for detecting dos attacks in iot networks based on machine learning algorithms. *Sensors, 24*(2), 713. https://doi.org/10.3390/s24020713

Alve, S. R., Mahmud, M. Z., Islam, S., Chowdhury, M. A., & Islam, J. (2025). Smart iot security: lightweight machine learning techniques for multi-class attack detection in iot networks. *arXiv preprint arXiv:2502.04057*.

Al-Otaibi, S. Z. (2022). Data security challenges with its defence strategies of internet of things: critical review study. *Communications in Mathematics and Applications, 13*(1), 401-415. https://doi.org/10.26713/cma.v13i1.1980

Amgbara, S. I., Akwiwu-Uzoma, C., & David, O. (2024). Exploring lightweight machine learning models for personal internet of things (iot) device security. *World Journal of Advanced Research and Reviews, 24*(2), 1116-1138. https://doi.org/10.30574/wjarr.2024.24.2.3449

Amien, J. A., Ghani, H. A., Saleh, N. I. M., Soni, S., Fatma, Y., & Hayami, R. (2024). A comprehensive evaluation of multiclass imbalance techniques with ensemble models in iot environments. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 22*(3), 690. https://doi.org/10.12928/telkomnika.v22i3.25887

Ariffin, S. H. S., Chong, J. L., Latif, N. M. A., Malik, N. N. N. A., Arsat, R. S. S. b., Baharudin, M. A., … & Yusof, K. M. (2022). Intrusion detection system (ids) accuracy testing for software defined network internet of things (sdn-iot) testbed. *ELEKTRIKA- Journal of Electrical Engineering, 21*(3), 23-27. https://doi.org/10.11113/elektrika.v21n3.361

Bediya, A. K., & Kumar, R. (2021). A novel intrusion detection system for internet of things network security. *Journal of Information Technology Research, 14*(3), 20-37. https://doi.org/10.4018/jitr.2021070102

Chawla, S., & Thamilarasu, G. (2018). Security as a service: real-time intrusion detection in internet of things. In *Proceedings of the Fifth Cybersecurity Symposium* (pp. 1-4).

Cloudflare. (n.d.). *What is a denial-of-service (DoS) attack?*. https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/?utm_source=chatgpt.com

Dodson, D., Montgomery, D. C., Polk, W. T., Ranganathan, M., Souppaya, M., Johnson, S., … & Singh, J. (2021). *Securing small-business and home internet of things (IoT) devices: mitigating network-based attacks using Manufacturer Usage Description (MUD) (NIST Special Publication 1800-15)*. https://doi.org/10.6028/nist.sp.1800-15

Elrawy, M., Awad, A., & Hamed, H. (2018). Intrusion detection systems for iot-based smart environments: a survey. *Journal of Cloud Computing Advances Systems and Applications, 7*(1). https://doi.org/10.1186/s13677-018-0123-6

Falcão, A. O. (2024). Fast Mutual Information Computation for Large Binary Datasets. *arXiv preprint arXiv:2411.19702.*

Farfoura, M. E., Mashal, I., Alkhatib, A., Batyha, R. M., & Rosiyadi, D. (2025). A novel

lightweight machine learning framework for iot malware classification based on matrix block mean downsampling. *Ain Shams Engineering Journal, 16*(1), 103205.

Ghali, A. A., Ahmad, R., & Alhussian, H. (2021). A framework for mitigating ddos and dos attacks in iot environment using hybrid approach. *Electronics, 10*(11), 1282. https://doi.org/10.3390/electronics10111282

Govindaraju, S., Vinisha, W. V. R., Shajin, F. H., & Sivasakthi, D. A. (2022). Intrusion detection framework using auto-metric graph neural network optimized with hybrid woodpecker mating and capuchin search optimization algorithm in iot network. *Concurrency and Computation: Practice and Experience, 34*(24). https://doi.org/10.1002/cpe.7197

Hulayyil, S., Li, S., & Xu, L. (2023). Machine-learning-based vulnerability detection and classification in internet of things device security. *Electronics, 12*(18), 3927. https://doi.org/10.3390/electronics12183927

Husnain, M., Hayat, K., Cambiaso, E., Fayyaz, U. U., Mongelli, M., Akram, H., … & Shah, G. A. (2022). Preventing mqtt vulnerabilities using iot-enabled intrusion detection system. *Sensors, 22*(2), 567. https://doi.org/10.3390/s22020567

Hwang, S., & Kim, J. (2021). A malware distribution simulator for the verification of network threat prevention tools. *Sensors, 21*(21), 6983. https://doi.org/10.3390/s21216983

Karamollaoğlu, H., Yücedağ, İ., & Doğru, İ. A. (2022). *A hybrid pca-mao based lstm model for intrusion detection in iot environments*. https://doi.org/10.21203/rs.3.rs-2357212/v1

Kaur, K., Kaur, A., Gulzar, Y., & Gandhi, V. (2024). Unveiling the core of iot: comprehensive review on data security challenges and mitigation strategies. *Frontiers in Computer Science, 6*. https://doi.org/10.3389/fcomp.2024.1420680

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics, 8*(11), 1210. https://doi.org/10.3390/electronics8111210

Kim, T. (2024). A study on impact of lightweight cryptographic systems on internet of things-based applications. *Asia-Pacific Journal of Convergent Research Interchange, 10*(1), 49-59. https://doi.org/10.47116/apjcri.2024.01.05

Kim, D., Jeon, S., Shin, J., & Seo, J. T. (2023). Design the iot botnet defense process for cybersecurity in smart city. *Intelligent Automation & Soft Computing, 37*(3), 2979-2997. https://doi.org/10.32604/iasc.2023.040019

Maqbool, A. (2024). Intrusion detection using network traffic profiling and machine learning for iot. *Journal of Electrical Systems, 20*(3s), 2140-2149. https://doi.org/10.52783/jes.1813

Mliki, H., Kaceam, A. H., & Fourati, L. C. (2021). A comprehensive survey on intrusion detection based machine learning for iot networks. *ICST Transactions on Security and Safety, 8*(29), 171246. https://doi.org/10.4108/eai.6-10-2021.171246

Mohan, A. (2018). Application and usefulness of internet of things in information technology. *International Journal of Engineering and Management Research (IJEMR), 8*(3), 57-61.

Musthafa, M. B., Huda, S., Kodera, Y., Ali, M. A., Araki, S., Mwaura, J., … & Nogami, Y. (2024). Optimizing iot intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques. *Sensors, 24*(13), 4293. https://doi.org/10.3390/s24134293

Mutambik, I. (2024). An efficient flow-based anomaly detection system for enhanced security in iot networks. *Sensors, 24*(22), 7408. https://doi.org/10.3390/s24227408

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023).

CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors, 23*(13), 5941. https://doi.org/10.3390/s23135941

Nzeako, G., Okeke, C. D., Akinsanya, M. O., Popoola, O. A., & Chukwurah, E. G. (2024). Security paradigms for iot in telecom networks: conceptual challenges and solution pathways. *Engineering Science & Technology Journal, 5*(5), 1606-1626. https://doi.org/10.51594/estj.v5i5.1111

Olaniyi, O. O., Okunleye, O. J., Olabanji, S. O., Asonze, C. U., & Ajayi, S. A. (2023). Iot security in the era of ubiquitous computing: a multidisciplinary approach to addressing vulnerabilities and promoting resilience. *Asian Journal of Research in Computer Science, 16*(4), 354-371. https://doi.org/10.9734/ajrcos/2023/v16i4397

Rani, K. S., Parasa, G., Hemanand, D., Devika, S., Balambigai, S., Hussan, M. T., … & Jain, A. (2024). Implementation of a multi-stage intrusion detection systems framework for strengthening security on the internet of things. *MATEC Web of Conferences, 392*, 01106. https://doi.org/10.1051/matecconf/202439201106

Reyaz, M. A. T., & Vanitha, V. (2024). *Advancing healthcare iot security: federated learning and the fedavg approach*. https://doi.org/10.21203/rs.3.rs-4471086/v1

Rodríguez, D. Z., Okey, O. D., Maidin, S. S., Udo, E. U., & Kleinschmidt, J. H. (2023). Attentive transformer deep learning algorithm for intrusion detection on iot systems using automatic xplainable feature selection. *Plos One, 18*(10), e0286652. https://doi.org/10.1371/journal.pone.0286652

Sahu, S. K., & Mazumdar, K. (2024). Exploring security threats and solutions techniques for internet of things (iot): from vulnerabilities to vigilance. *Frontiers in Artificial Intelligence, 7*. https://doi.org/10.3389/frai.2024.1397480

Sayed, N., Shoaib, M., Ahmed, W., Qasem, S. N., Albarrak, A. M., & Saeed, F. (2023).

Augmenting iot intrusion detection system performance using deep neural network. *Computers, Materials & Continua, 74*(1), 1351-1374. https://doi.org/10.32604/cmc.2023.030831

Sharipuddin, S., Purnama, B., Kurniabudi, K., Winanto, E. A., Stiawan, D., Hanapi, D., … & Budiarto, R. (2021). Intrusion detection with deep learning on internet of things heterogeneous network. *IAES International Journal of Artificial Intelligence (IJ-AI), 10*(3), 735. https://doi.org/10.11591/ijai.v10.i3.pp735-742

Singh, S., Singh, D. P., Chandra, K., & Singh, B. (2023). IoT security challenges and emerging solutions: a comprehensive review. *International Journal of Scientific Research in Engineering and Management, 07*(09). https://doi.org/10.55041/ijsrem25662

Sriram, N. (2006). Decomposing the Pearson Correlation. *Social Science Research Network*. https://doi.org/10.2139/SSRN.2213946

Tawalbeh, L., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). Iot privacy and security: challenges and solutions. *Applied Sciences, 10*(12), 4102. https://doi.org/10.3390/app10124102

Tyagi, H., & Kumar, R. (2021). Attack and anomaly detection in iot networks using supervised machine learning approaches. *Revue d'Intelligence Artificielle, 35*(1), 11-21. https://doi.org/10.18280/ria.350102

Uhm, Y., & Pak, W. (2022). Real-time network intrusion prevention system using incremental feature generation. *CMC-Comput. Mater. Contin, 70*, 1631-1648.

Wardana, A. A., Kołaczek, G., & Sukarno, P. (2024). Lightweight, trust-managing, and privacy-preserving collaborative intrusion detection for internet of things. *Applied Sciences, 14*(10), 4109. https://doi.org/10.3390/app14104109

Xenofontos, C., Zografopoulos, I., Konstantinou, C., Jolfaei, A., Khan, M. K., & Choo, K. R.

(2022). Consumer, commercial, and industrial iot (in)security: attack taxonomy and case studies. *IEEE Internet of Things Journal, 9*(1), 199-221. https://doi.org/10.1109/jiot.2021.3079916

Zhang, J., Li, Y., & Zhang, L. (2024). Heterogeneous network intrusion detection via domain adaptation in iot environment. *Internet Technology Letters, 8*(1). https://doi.org/10.1002/itl2.531

RESEARCH ETHICS APPROVAL

Office Record	Receipt – APU Fast-Track Ethical Approval
	Student name:
Date Received:	Student number:
	Received by:
Received by:	Date:

APU/APIIT FAST-TRACK ETHICAL APPROVAL FORM (STUDENTS)

Tick one box (level of study):
- ☑ POSTGRADUATE (PhD / MPhil / Masters)
- ☐ UNDERGRADUATE (Bachelors degree)
- ☐ FOUNDATION / DIPLOMA / Other categories

Tick one box (purpose of approval):
- ☑ Thesis / Dissertation / FYP project
- ☐ Module assignment
- ☐ Other: _____

Title of Programme on which enrolled … Master of Science in Cyber Security …

Tick one box: ☑ Full-Time Study or ☐ Part-Time Study

Title of project / assignment … A Machine Learning Framework for Real-Time Detection of DoS Attacks in IoT Environments …

Name of student researcher … Koo Wai Kit …

Name of supervisor / lecturer… Ts. Dr. Vinesha A/P Selvarajah …

Student Researchers- please note that certain professional organisations have ethical guidelines that you may need to consult when completing this form.

Supervisors/Module Lecturers - please seek guidance from the Chair of the School Research Ethics Committee if you are uncertain about any ethical issue arising from this application.

		YES	NO	N/A
1	Will you describe the main procedures to participants in advance, so that they are informed about what to expect?			✓
2	Will you tell participants that their participation is voluntary?			✓
3	Will you obtain written consent for participation?			✓
4	If the research is observational, will you ask participants for their consent to being observed?			✓
5	Will you tell participants that they may withdraw from the research at any time and for any reason?			✓
6	With questionnaires and interviews will you give participants the option of omitting questions they do not want to answer?			✓
7	Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?			✓
8	Will you give participants the opportunity to be debriefed i.e. to find out more about the study and its results?			✓

If you have ticked No to any of Q1-8, you should complete the full Ethics Approval Form.

		YES	NO	N/A
9	Will your project/assignment deliberately mislead participants in any way?			✓
10	Is there any realistic risk of any participants experiencing either physical or psychological distress or discomfort?			✓
11	Is the nature of the research such that contentious or sensitive issues might be involved? This includes research which could induce psychological stress, anxiety or humiliation, or cause more than minimal pain.		✓	
12	Does your research involve the use of sensitive materials? Eg, records of personal or sensitive confidential information,		✓	

13	Does your research require external agency approval?		✓	
14	Does your research use hazardous or controlled substance?		✓	
15	Does your research require you to visit participants in their home or non-public space?		✓	
16	Does your research use genetically modified organisms?		✓	
17	Does your research investigate illegal activities or behaviours?		✓	
18	Does your research involve discussion or collection of information on potentially sensitive, embarrassing or distressing topics, administrative or secure data? This includes research involving respondents through internet where visual images are used, and where sensitive issues are discussed		✓	
19	Does your research involve invasive or potentially intrusive procedures?		✓	
20	Does your research involve administration of substances?		✓	
21	Will your research be involved in the collection/ processing of human tissue samples		✓	
22	Will your participants be receiving financial compensation for participating in your research?			✓
23	Will your research data be used in the future after the conclusion of your project?		✓	
24	Will your research involve in processing sensitive data belonging to an organisation/persons?		✓	
25	Will your research be collecting photographs, videos, and audio recordings of the participants?			✓
26	Will the participants' personal particulars be known to any third party?			✓
27	Will the participants' data confidentiality be made known to the public?			✓
28	Will the research be conducted where the safety of the researchers maybe in question?		✓	
29	Will be the research be conducted outside of the UK and/or Malaysia?		✓	
30	Will your research involve human participants at premises other than those of the University?		✓	

If you have ticked Yes to any of Q9 – 30, you should complete the full Ethics Approval Form. In relation to question 10 this should include details of what you will tell participants to do if they should experience any problems (e.g. who they can contact for help). You may also need to consider risk assessment issues.

			YES	NO	N/A
31	Does your project/assignment involve work with animals?			✓	
32	Do participants fall into any of the following special groups? **Note that you may also need to obtain satisfactory clearance from the**	Children (under 18 years of age)			✓
		People with communication or learning difficulties			
		Patients			
		People in custody			
		People who could be regarded as vulnerable or lack capacity to make decision for themselves			
		People engaged in illegal activities (eg drug taking)			
		Groups of people whose relationship among each other allow one to have influence over the other such as: Carers and patients with chronic conditions; teachers and their students; prison authorities and prisoners;			

	relevant authorities	employers and employees			
		Deceased person's body parts or other human tissues including bodily fluids (e.g. blood, saliva).			
		groups where permission of a gatekeeper is normally required for initial access to members.			
		Human participants who are off-campus APU staff or students who wish to carry out investigations involving human participants at premises other than those of the University			
33	Does the project/assignment involve external funding or external collaboration where the funding body or external collaborative partner requires the University to provide evidence that the project/assignment had been subject to ethical scrutiny?			✓	

If you have ticked Yes to any Q31-33**,** you should complete the full Ethics Approval Form. There is an obligation on student and supervisor to bring to the attention of the APU School Research Ethics Committee any issues with ethical implications not clearly covered by the above checklist.

STUDENT RESEARCHER

Provide in the boxes below (plus any other appended details) information required in support of your application. THEN SIGN THE FORM.

Please Tick Boxes

I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU School Research Ethics Committee.	✓
I am aware of APU liability policy and will make the necessary arrangement for insurance coverage of all researchers and participants of the project/assignment.	✓
Give a brief description of participants, procedure of recruitment and procedure of data collection (methods, tests used etc) in up to 150 words. This research project does not involve any human participants. Therefore, no recruitment or consent procedures are applicable. Data will be collected through publicly available datasets that contain labelled IoT network traffic data, including normal and attack patterns. No real-world or live network data will be used. Simulation tools will be used to replicate IoT environments, enabling the evaluation of machine learning models. No personal or sensitive data will be involved in this study.	
I also confirm that: i) All key documents e.g. consent form, information sheet, questionnaire/interview, and all material such as emails and posters for the purpose of recruitment of participants are appended to this application.	✓
Or ii) Any key documents e.g. consent form, information sheet, questionnaire/interview schedules which need to be finalised following initial investigations will be submitted for approval by the project/assignment supervisor/module lecturer before they are used in primary data collection.	✓

Signed… … … … … … … … … … … … Print Name… Koo Wai Kit … … … … … … … … … … Date… 16/5/2025 … … …
(Student Researcher)

Within this document, any variation to the items considered which affects ethical issues of the stated research will require submission of a revised research plan and research methodology details; as a consequence, new ethical consent may need to be sought.

The completed form (and any attachments) should be submitted for consideration by your Supervisor/Module Lecturer

SUPERVISOR/MODULE LECTURER
PLEASE CONFIRM THE FOLLOWING:

	Please Tick Box
I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU School Research Ethics Committee	✓
I have checked and approved the key documents required for this proposal (e.g. consent form, information sheet, questionnaire, interview schedule)	✓

SUPERVISOR AND SECOND ACADEMIC SIGNATORY

STATEMENT OF ETHICAL APPROVAL (please delete as appropriate)

1) **THIS PROJECT/ASSIGNMENT HAS BEEN CONSIDERED USING AGREED APU PROCEDURES AND IS NOW APPROVED**

2) **THIS PROJECT/ASSIGNMENT HAS BEEN APPROVED IN PRINCIPLE AS INVOLVING NO SIGNIFICANT ETHICAL IMPLICATIONS, BUT FINAL APPROVAL FOR DATA COLLECTION IS SUBJECT TO THE SUBMISSION OF KEY DOCUMENTS FOR APPROVAL BY SUPERVISOR (see Appendix A)**

Signed… VIN … … … … … … … … … … … Print Name… Ts. Dr. Vinesha A/P Selvarajah … … … … … … … … … … Date… 16/5/2025 … … …
(Supervisor/Lecturer)

Office Record	Receipt – Appendix A (APU Fast-Track Ethics Form)
Date Received:	Student name: Student number: Received by:
Received by:	Date:

APPENDIX A
AUTHORISATION FOR USE OF KEY DOCUMENTS

Completion of Appendix A is required when for good reasons key documents are not available when a fast track application is approved by the supervisor/module lecturer and second academic signatory.

I have now checked and approved all the key documents associated with this proposal e.g. consent form, information sheet, questionnaire, interview schedule

Title of project/assignment… A Machine Learning Framework for Real-Time Detection of DoS Attacks in IoT Environments …

Name of student researcher … Koo Wai Kit …

Student ID: TP081761 …… Intake: … APDMF2406CYS …

Signed… VIN … … … … … … … … … … … Print Name… Ts. Dr. Vinesha A/P Selvarajah … … … … … … … … … … Date… 16/5/2025 …
(Supervisor/Lecturer)