جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

Kingdom of Saudi Arabia
Ministry of Higher Education
Imam Abdulrahman Bin Faisal University
College of Computer Sciences & Information Technology

“Detecting DDoS Attacks Using Machine Learning Techniques”

ML
DETECT

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Science in {Program Name}

by

Nasser A. Alhajri (2210006797)

Rayyan B. Al Nahwi (2210002103)

Ibrahim A. Al Husaini (2210002314)

Aseel A. Alrudayni (2210001894)

Noor A. Hadari (2210002310)

Supervised by

Dr.Dhiaa A. Musleh

Committee Member Names

Dr. Yasir Alguwaifli

Ms. Dalal Aldowaihi

8/2024

DECLARATION

We hereby declare that this project report entitled “ML Detect” is based on our original work except for citations and quotations, which have been duly acknowledged. We also declare that it has not been previously or concurrently submitted for any other degree or award at any university or any other institution. This project work is submitted in the partial fulfillment of the requirements for the degree of “Bachelor of Science in Computer Science” at Computer Science Department, College of Computer Sciences and Information Technology, Imam Abdulrahman Bin Faisal University.

Project Team:

Group Number: 9MS5		
Team Member’s Name	Signature	Date
Nasser Alhajri		
Rayyan Al Nahwi		
Ibrahim Al Husaini		
Aseel Al rudayni		
Noor Hadari		

Project Supervisor:

Name	Signature	Date
Dr.Dhiaa A. Musleh	Dhiaa	

Table of Contents

Table of Figure: 5
Table of Tables 6
List of Abbreviations: 7
Acknowledgement 8
Abstract 9
Chapter 1: Introduction 10
1.1 Introduction 10
1.2 Problem Statement 10
1.3 Background 10
1.4 Motivation 10
1.5 Justification 11
1.6 Aims & Objectives 11
1.7 Scope / Limitation of the Study 11
1.8 Social, Professional, legal, and ethical Implications 11
1.9 Project Organization 12
1.10 Business impact 12
1.11 Chapter summary 13
Chapter 2: Background and Review of Literature 14
2.1. Introduction 14
2.2. Background 14
2.2.1 Overview 14
2.2.2 Types of DDoS attacks 14
2.2.3 Impact of DDoS Attacks 15
2.2.4 Challenges in DDoS Detection and Mitigation 15
2.2.5 Current solutions and limitations 16
2.3. Literature Review 16
2.3.1 Evolution of DDoS Attacks and Traditional Detection Methods 17
2.3.2 Software-Defined Networking (SDN) Frameworks and Smart City DDoS Detection 17
2.3.3 Anomaly Detection Techniques for DDoS Defense 18
2.3.4 IoT Vulnerabilities and ML Solutions 18
2.3.5 DL for Complex DDoS Detection 19
2.3.7 Collaborative Detection and Cross-Network Defense 20

2.3.8 AI and Ensemble Learning for DDoS Detection 20
2.3.9 Efficient Detection in IoT with ML 21
2.3.10 Transfer Learning and Cross-Context Adaptation 21
2.3.11 The Future of ML in DDoS Detection 21
2.4 Knowledge Gap. 22
2.5 Chapter summary 22
Chapter 3: SPMP and Requirement Specification 26
3.1 Project Overview 26
3.1.1 Purpose, Scope, and Objectives 26
3.1.2 Assumptions, Constraints, and Risks 27
3.1.3 Project Deliverables 27
3.1.4 Schedule and Budget Summary 28
3.1.5 Evolution of the Plan 28
3.1.6 References 28
3.2 Project Organization 28
3.2.1 External Interfaces 28
3.2.2 Internal Structure 29
3.2.3 Organizational Structure 30
3.3 Managerial Process Plans 31
3.3.1 Startup Plan 31
3.3.2 Work plan 32
3.3.3 Project Tracking plan: 34
3.3.4 Risk Management Plan 36
3.3.5 Project Closeout Plan 36
3.4 Technical Process Plan 37
3.4.1 Process Model 37
3.4.2 Methods, Tools, and Techniques 38
3.4.3 Infrastructure 38
3.4.4 Product Acceptance 38
3.5 Supporting Process Plans 38
3.5.1 Documentation 39
3.6 Overall Description 39
3.6.1 Product Perspective 39

3.6.2 Product Functions 39

3.7 Specific Requirements 40

3.7.1 User Interfaces 40

3.7.2 Functional Requirements 40

3.7.3 Non-Functional Requirements 41

Chapter 4: Methodology & Design 42

4.1 Methodology 42

4.1.1 Methodology Framework 42

4.1.2 Definition of Proposed Classification Techniques 47

Chapter 5: CONCLUSION .... **Error! Bookmark not defined.**

5.3 Issues Faced 79

5.4.1 Technical Skills 80

5.4.2 Managerial Skills 81

REFRENCESS 82

Table of Figure:

Figure 1:External Interfaces 28

Figure 2:Project Organization with Internal Structures. 29

Figure 3: Phases of developing a ML model 44

Figure 4: The Random Forest Algorithm [36] 48

Figure 5: Homepage 57

Figure 6:Problem/Solution 57

Figure 7: Login 58
Figure 8: Sign Up........ 59
Figure 9: Privacy and policy 59
Figure 10: Solution Interface 60
Figure 11: Sequence Diagram Login 66
Figure 12: Sequence Diagram Detection function........ 67

Table of Tables

Table 1. Summary of Literature Review........ 25
Table 2: Assumptions, constraints, and risks table 27
Table 3 Organizational Structure 30
Table 4 Staffing plan........ 31
Table 5 Project staff training plan........ 32
Table 6 Work activities........ 32
Table 7 Schedule allocation........ 33

Table 8 Resources allocation 34
Table 9 Project Metrics 35
Table 10 Risk Management Plan 36
Table 11 Infrastructure 38
Table 12: Homepage Interface 64
Table 13: Login Page 64
Table 14:Sign Up Page 65
Table 15:Solutions Page 65
Table 16:Login Diagram 67
Table 17:Detection Diagram 67

List of Abbreviations:

Abbreviations	Definition
AI	Artificial intelligence
DDoS	Distributed Denial of Service
DOS	Denial of Service
ML	Machine learning
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
SYN	Synchronize
ICMP	Internet Control Message Protocol
OSI	Open Systems Interconnection
IPS	Intrusion Prevention Systems
SDN	Software-Defined Networking
DL	Deep Learning
TMR	Triple Modular Redundancy
IoT	Internet of Things

TD	True Positive Detection
FP	False Positive
IXP	Internet Exchange Point
SVM	Support Vector Machine
k-NN	k-Nearest Neighbors
RNN	Recurrent Neural Network
SDN	Software-Defined Networking
CNN	Convolutional Neural Network
DFS	Decision Feedback System
NFs	Network Functions
DNS	Domain Name System
AIMM	AI-based Intelligent Mitigation Model
IDS	Intrusion Detection System
LR	Logistic Regression
RF	Random Forest
DT	Decision Tree
NB	Naive Bayes
IDPS	Intrusion Detection and Prevention System
IPFIX	Internet Protocol Flow Information Export
XAI	Explainable AI
LSTM	Long Short-Term Memory
TCP	Transmission Control Protocol
CIC	Canadian Institute for Cybersecurity
VSC	Visual Studio code
PCA	Principal Component Analysis
RFE	Recursive Feature Elimination

Acknowledgement

Our team members would like to express their absolute thanks to Allah for everything, especially for this opportunity. We would like to also Thank our graduation project supervisor Dr. Dhiaa Musleh for all his contributions in excelling this project to its upmost potential, his support gave us the drive and motivation to strive better and to overcome all obstacles together. Additionally, we would like to thank our families for their support, which effectively demonstrated our motivation and made us the best version of ourselves.

Abstract

This project aims to improve the latest techniques used in Distributed Denial of Service attacks (DDoS) detection using Artificial Intelligence (AI) and Machine Learning (ML). Conventional methods of DDoS detection have thus been proved to be inadequate given the complexity of the attacks. Consequently, the project looks to use various machine learning techniques with a view to improve the functionality of DDoS detection systems. Furthermore, it assesses the use of these techniques in cloud and Internet of Things (IoT) computing paradigms to overcome different issues arising from these frameworks. The idea is to build upon current systems that lack detection and enhance the model interpretability such that cybersecurity experts can address threats. This study will help to build effective, flexible and optimal anti- DDoS approaches that will help to strengthen the digital systems against new forms of cyber threats.

Chapter 1: Introduction

1.1 Introduction

With the expansion of the Internet and increased reliance on it, cyber threats and their complexity have increased. One of these threats is Distributed Denial of Service (DDoS) which aims to flood available software with a large amount of traffic. We will focus on using Artificial Intelligence (AI) to ease the process of detecting DDoS. In this chapter, we will start talking about the problem, then the background for DDoS, then the motivation and proof, then the limits of this project.

1.2 Problem Statement

Imagine working in a reputable company such as Google, then suddenly, your computer freezes making you unable to continue the work intended. While the employee struggles with the computer, it turns out that many employees are facing the same issue. Symptoms such as a frozen computer or a sluggish responding computer are related to a widely known attack called a DDOS attack, this attack has caused devastating incidents which altered the way people handle the organization's security. Moreover, as we progress in this evolution in techno-related subjects, we need to provide better security to our organizations to ensure that many of the attacks such as the DDOS attacks can be minimized and captured before any major issues arise. AI and ML are making a noticeable evolution in modern days [1], so why don't we use AI and ML to our advantage? How can we achieve this goal?

1.3 Background

In today's fast and enormous rate of growth in both AI and Cybersecurity, we face many challenges safeguarding the most important assets from the increasing cyber threats that are arising with this growth rate. Having AI by your side can make cover huge amounts of data in seconds, which can detect any distinguish data or manipulated data that could be associated with cyber threats. In this modern era, we need to process huge amounts of data as it enters the network of a certain company. Additionally, we must have a fast-learning model that can keep up with this fast growth and ensure the safety of companies.[2]

1.4 Motivation

DDoS threats are not stoppable, and it increases risk to company assets and infrastructure. We could mitigate that risk using ML especially with its rapid improvement. The project has potential for significant real-world impact. By successfully identifying DDOS attacks, this helps protect organizations from losses or damage associated with services disruptions.

1.5 Justification

With the increasing reliance on digitalization cloud and services, the ability to detect DDoS attacks in datasets is crucial for mitigating damage risk from happening. By focusing on ML, it will be high safety to infrastructure services and resource being on demand, leading to resolve critical gaps in current cybersecurity.

1.6 Aims & Objectives

- Understanding DDoS Attack Dynamics and Mechanisms.
- Collect and Preprocess Data.
- Use machine learning algorithms to detect DDoS attacks.
- Evaluate System Effectiveness, by extracting data and testing our detection rate.

1.7 Scope / Limitation of the Study

The project's main goal is to utilize ML to detect DDoS attacks. Application-layer attacks are not included in our project because it is complex and requires specific datasets. The project utilizes publicly accessible datasets that may not accurately reflect actual traffic or attack methods.

1.8 Social, Professional, legal, and ethical Implications

The use of AI and ML in cybersecurity, most especially in the detection of DDoS attacks has enormous social, professional, legal, and ethical concerns.

- **Social Implications:** Applying AI to strengthen cybersecurity can increase the public's trust in the digital media and various services. It also raises concerns with emphasis on privacy as well as moral ways of applying AI technology on network traffic monitoring.

- **Implications for Professionals:** Security experts are going to be forced to adapt in the manner that conventional security work is progressively being integrated with AI. As will be appreciated by a reader; to maximize usage of these technologies, one must work with AI experts and learn continuously.

- **Legal Implications:** it means that data protection laws should be complied with when involving AI in cybersecurity. To avoid any legal consequences, it is possible to stress the need to ensure the anonymity and safety of data for training AI models.

- **Ethical Implications:** It is therefore clear that, when used to detect cyberattacks, AI needs to be impartial, and its usage has to be openly declared. Avoidance of prejudice and bias against certain people and constitutional responsibility for actions performed via the use of AI system are ethical concerns.

1.9 Project Organization

Chapter1: Introduction

This chapter describes our project, which aims to detect DDoS attacks using AI and ML. The issue statement, background, motivation, goals, and the project's restrictions and scope are all covered. There is also discussion of ethical, legal, professional, and social ramifications.

Chapter 2: Background and Literature Review

This chapter offers a thorough analysis of the state-of-the-art in terms of DDoS attack avoidance, AI, and ML applications in cybersecurity.

Chapter 3: SPMP and Requirement Specification

Here, we go over the steps involved in creating the AI model, such as gathering data, preparing it, and using ML techniques.

Chapter 4: Methodology & Design

This section covers how the AI model will be incorporated into the cybersecurity framework and outlines the system's design and architecture, including the hardware and software requirements.

Chapter 5: Implementation

This chapter goes over the testing protocols and assessment metrics that are used to determine how well the system detects DDoS attacks.

Chapter 6: Conclusion and Future Work:

The project's results are summed up in the last chapter, which also addresses possible improvements and future study avenues.

1.10 Business impact

Modern businesses have greatly relied on the internet, hence exposing them to DDoS attacks. Therefore, this paper concludes that it is critical to have an effective DDoS detection policy that can be applied in an organization to enhance financial security, reputation and trust from customers. There are large scales of damage that can be caused by such attacks including a lot of money loss due to time wastage. In terms of cost savings, from this perspective, it is far more beneficial for businesses to invest in the detection of disruptions as this means that the service a company provides will be running as fluid as possible and, consequently, generate as much revenue as possible. It also makes future preparation easier, builds up a good image by assuring customers of reliability and continuity of services, which means customer loyalty. Furthermore, good DDoS detection helps customers to trust the services they employ and ensures the continuity of such relations. In conclusion, securing an enterprise not only protects operations, but also paves the way on trust which is crucial and creates a long-term competitive advantage.[3]

1.11 Chapter summary

To conclude, our goal is to use AI and ML to detect DDoS attacks. We demonstrated the issue by showing how these attacks affect organizations. Through the use of a real-time threat identification and mitigation paradigm, the project aims to improve network security. Understanding DDoS dynamics, working with specialists, and assessing system efficacy are important goals. We also talked about the ethical, legal, professional, and social ramifications of using AI in cybersecurity. All things considered, this chapter offers a thorough overview of the project's objectives and the plan for reaching them.

Chapter 2: Background and Review of Literature

2.1. Introduction

This chapter will provide an overview of existing software and services to counteract DDoS attacks. A review of the current methods and techniques in practice, with emphasis on ML techniques in improving the detection systems. Also, the current literature review will indicate the existing gaps in knowledge, the tables will present the current software features comparison, with some of the marked absence functionalities.[4]

2.2. Background

As we progress in today's vast evolving digital development, as is the case for any cyber-attacks, hence the need for a powerful tool to detect such attacks. One of the most devastating attacks is the DDoS attack, which targets a server by flooding it with a huge amount of traffic through the network, this caused many challenges to all private and public sectors across the world, even global companies such as Microsoft, Google and others faced these challenges at some point. Therefore, ML is one of the most promising solutions to detect all sophisticated DDoS attacks, by using many ML algorithms such as: Random Forest, Extreme Gradient Boosting (XGBoost) and many others will aid many companies to be safe and cautious from DDoS attacks. Additionally, even with the evolvement of DDoS attacks. Hackers are advancing with their improvements; therefore, we should be prepared with solutions to mitigate these attacks in any shape or form.[5]

2.2.1 Overview

A DDoS attack is an intentional attempt to orchestrate the stoppage of a specified server, service, or network through floods of External Traffic. While a simple DDoS comes from a single location, DDoS consists of many contaminated machines, usually affiliated with a botnet, that overwhelm the intentioned site. These compromised devices can be situated in various parts of the world; thus, the attack is of a distributed type. The mechanism of a DDoS attack is aimed at completely consuming the bandwidth or resources of the targeted system so that genuine clients cannot access the Service. Traditional large-scale attack types such as DDoS are problematic to manage as they bring in a lot of traffic and it is hard to determine which traffic is bad.[6]

2.2.2 Types of DDoS attacks

DDoS attacks can be broadly classified into three main types based on the method of attack [1][7]:

- **Volumetric Attacks:** These are some of the most familiar kinds of DDoS attacks and the goal of those kinds of attacks is to use up the bandwidth of the target. Currently, the attackers congest the network with a hard and overwhelming bandwidth utilization. Some of the examples are User Datagram Protocol (UDP) flood or Internet Control Message Protocol (ICMP) flood.

- **Protocol Attacks:** These attacks take advantage of unrepaired weak points in network protocols to flood the network devices, including firewalls or load balancers, with excessive traffic. One classic example is the Synchronize (SYN) flood; this involves a suspect that sends SYN requests to begin a connection but does not proceed to the third stage of connection.

- **Application Layer Attacks:** These attacks flood the application with apparently valid requests improving its capability to pass through the system defenses. They are aimed at flooding the application layer (Layer 7 on the OSI model) that has to do with Web traffic. An example with the same intent is the Hypertext Transfer Protocol (HTTP) flood in which the attacker floods the web server with multiple requests with an intention of overloading it.

2.2.3 Impact of DDoS Attacks

DDoS attacks can have far-reaching consequences on organizations, including [3]:

- **Financial Loss:** Through DDoS attacks, the risks are that service availability decreases significantly and is unavailable for long periods, which is especially disastrous for firms whose operation depends on such services for revenue generation, such as e-commerce sites. Failure to use the technology for a single minute could reduce sales and cost more to bring the system back online.
- **Operational Disruption:** The worst type of cyber assault can shut down entire networks or services so that bona fide customers are unable to gain access. This will disrupt its working, affect customer relations and in extreme cases, the organization's credibility will be at risk.
- **Data Breaches and Exploits:** it is, however, not the primary intention of DDoS attacks, though they can be used to distract the attention of the owner while the attacker conducts other sinister activities such as data theft, or malware installation.

2.2.4 Challenges in DDoS Detection and Mitigation

- **High Volume of Traffic:** DDoS attacks produce high volumes of traffic that are difficult to handle and a regular mechanism of filtering out the malicious request compromises legitimate users of the services.

- **Legitimate vs. Malicious Traffic:** Most of today's DDoS attacks are byzantine in nature and bear the hallmarks of normal, benign traffic. They can send request which looks legitimate to many basic stages of common detection systems.

- **Evolving Attack Strategies:** Cyber attackers are not passive; they are constantly changing their form of attack by developing new methods and fusing two or more attack modes collectively referred to as multi-vector attacks. This makes the traditional detection systems less efficient the longer the time.

- **Global Distribution:** DDoS attacks in most cases originate from a group of affected devices spread across the physical geography. That is why they are very difficult to prevent since it may be very difficult to identify and filter the source of the attack.

2.2.5 Current solutions and limitations

Existing DDoS detection solutions typically include:

- **Signature-Based Detection:** This approach is based on known attack patterns and signatures causing a heavy reliance on signatures. Still, it often fails at detecting new or emerging or complex threats that may differ from the known ones.

- **Anomaly-Based Detection:** Anomaly detection systems help in the identification of network traffic that is most unusual and/or apart from the normal traffic. As for the techniques described above, this approach can successfully identify new attack types but at the same time leads to high false-positive rates where regular traffic is mistaken for attacks.

- **Rate Limiting and Traffic Filtering:** They include techniques where specific traffic is blocked, or rather the rate at which requests are made from an IP address is slowed down. Rate limiting is an effective countermeasure to some forms of attack, but it can create difficulties for those users who are completely innocent, including when the input is heavy.

Nevertheless, most of them are still traditional and hence, reactive and can barely cope with the current day and age DDoS attack extensions. This has fostered the desire to use AI and ML technologies in real-time detection and much improved defenses.[8][9]

2.3. Literature Review

The expansion of online technologies and the increasing use of centralized computing and the use of Internet of Things (IoT) products have dramatically opened up new markets and industries for buyers and sellers. However, this has also created a fertile environment for an increase in cybercrimes, and one of the most dramatic types of attacks is a DDoS attack. The purpose of such attacks is to saturate a target network with traffic in order to prevent all of its rightful users from utilizing the service. This section will illustrate that DDoS attacks have changed in different ways over the years, and that the attackers are getting smarter. This literature review discusses the body of work on the current research on ML and DL for the detection of DDoS attack, which has attracted much attention due to the increasing scale and sophistication of the attack.[10]

2.3.1 Evolution of DDoS Attacks and Traditional Detection Methods

Originally, DDoS attacks were rather unsophisticated: large amounts of traffic got to the network with the intention of overloading it. These volumetric attacks are still present today though with the near impenetrable barriers being developed by various organizations, new approach has been adopted including application layer attacks, reflection-based attacks, and integrated multiple vector attacks that get to various layers of hosts in the network. The traditional approach of detecting DDoS attacks Prior was the capacity of ineffective and painting-only threshold-based detection mechanisms Depending on the basic attacks, such as rate limiting or deep packet inspection [11].

With these new sophisticated forms of DDoS reflected in low volume and stealth attacks, traditional static methods of detecting attacks can only do so much thus requiring new fresh means. This appears to provide a unique solution for the use of ML in detecting abnormality within sets of traffic data found within a network. While training on history data, ML models are able to recognize signs with the deviation of which may indicate an attack. For instance, [11] proposed a ML classification of DDoS attacks based on Decision trees and Random forest. Their approach enables the system to predict attacks on the basis of traffic flow patterns and as a result provides an opportunity for a preventive mechanism of the system compared to a reactive measure when an attack is already ongoing.

2.3.2 Software-Defined Networking (SDN) Frameworks and Smart City DDoS Detection

The use of networks has grown complex, more so in smart cities where the network connectivity is more extensive leading to opportunities for a DDoS attack to trigger massive disturbance. Smart cities have interrelated systems of services that include transport, health, and utilities. In their work, [12], proposed an SDN framework for securing those important structures against DDoS attacks. SDN optimizes rationalization control over the network because via it administrators freely can reprogram the network in real-time. effectively explained how SDN controllers can work continuously for smart city networks, through incorporating AI-based anomaly detection, for the rejection of even highly developed DDoS attacks. The use of SDN to mitigate DDoS threats has gained considerable traction due to its ability to centralize control and offer greater flexibility in managing network traffic. [13] This paper discussed the use of SDN in cloud architecture to show that the flexible model of SDN can help monitor and counteract DDoS attacks in real-time. Their work integrated ML modelling in the SDN environment to upgrade the dexterity in identifying the attacks at the transport as well as application layers

2.3.3 Anomaly Detection Techniques for DDoS Defense

Another benefit of using ML in DDoS detection is the possibility to undertake the implementation of the anomaly-based filtering. Also, instead of detecting certain signs that are typical of certain types of attack, as in the case of signature-based computer systems, the anomaly-based computer systems can detect new attacks since they are able to compare the observed network behavior to the normal behavior.[14] oriented their work on anomaly-based filtering applied to Domain Name System (DNS) authoritative servers where the authors described the problem of differentiating between normal variations in traffic and actual intrusion. While anomaly detection systems are quite effective, they pose a major challenge due to high false positive rates and have to decide when to block traffic that might interfere with legitimate traffic during other busy traffic times.

extended the theoretical application of anomaly detection in mitigating DDoS attacks in a different way and within the context of IoT networks. Regarding their research findings, they were more concerned with a factor which is common among IoT devices, that is, they are usually characterized by limited computational capabilities a fact which makes them prone to attackers; Therefore, the objective of their work was aimed at determining how IoT devices stand to benefit from easier methods of anomaly detection. The study also proposed a machine learn approach using Network Functions (NFs) as features to capture early malicious traffic behavior before they produce a voluminous traffic that traditional means cannot capture due to resource constraints as observed in real-world IoT environment. [15]

extended the anomaly-based filtering to the IoT and cloud computing, especially for real time detection. Their work showed that with ML it is possible detect Slowloris attack in high traffic network as their attacks belong to the application layer attacks which are normally difficult to detect because of their inherent nature. [16]

2.3.4 IoT Vulnerabilities and ML Solutions

Since more services are being shifted online and the Internet of Things is expanding, DDoS attacks are a growing problem. The IoT devices are more vulnerable since they have restricted security measures to begin with; hence, those are the perfect targets for hackers who wish to form botnets.[17] work on detecting DDoS attacks in IoT settings also highlights the issues of feature selection in ML models. His study also implemented a new method of feature selection incorporating XGBoost with detection accuracy of 99.993%. The effectiveness of this strategy in a high density IoT network clearly indicates a need for an adaptive detection system that can quickly scale up to keep with the increasing IoT demand.

The above-discussed idea has further been expanded by **Chakraborty et al.** [18] who compared various ML algorithms in the detection of DDoS attacks in IoT networks. Their work focused on comparing logistic regression (LR) with decision trees, random forest techniques and they indicated that even though random forest model gave the best result, LR served the same purpose with improved computational results that fit even the most constrained IoT devices. This balance between detection performance and computational efficiency is paramount for enhancing the applicability of ML based DDoS detection methods for large scale IoT ecosystems.

2.3.5 DL for Complex DDoS Detection

In the recent past however due to advancements in ML methodologies, there has been discussion on the need to develop other models such as the DL model. Decision trees and Support Vector Machine (SVM) have been used earlier to detect obvious DDoS attacks, but new trends in attacks are more sophisticated and less noticeable and hence more complex to analyze. The ability of using convolutional neural networks (CNN) and recurrent neural networks (RNN) in identifying application-layer DDoS attacks was also studied by **Kumar et al.** [19]. These attacks are a bit challenging to halt because they seem to emulate normal traffic at the same time flood the target's resources. Due to opportunities to work with a large number of records and find concealed relationships, CNNs and RNNs can become a suitable solution for searching for such a more sophisticated type of attack.

subsequently extended the work on DL in SDN based network, where the performance of DL models in identifying transport-layer and application-layer DDoS attacks was highlighted. Their study demonstrated that by integrating both ML and DL, SDN controllers greatly enhance their capacity to address a significant number of attacks while avoiding the problem of false positives.[20]

offered one more viewpoint of using DL to detect DDoS attacks on the IoT networks. They investigated Rapid Miner and used Random Forest and Decision Trees models that yielded high detection accuracy but argued that the problem was the significant numbers of data that IoT devices communicate. It was found in their research that it is crucial to use efficient methods of further data processing within the real time attack detection environment. [21]

In this direction, **Hameed et al.** [22] extended this context analyzing the integration between ML models and Anomaly Detection Systems, pointing out that response time is a major issue in IoT and SDN networks. In their work, they showed that high detection rates of these anomalies could help minimize the impact of DDoS attacks, in terms of downtime.

2.3.6 Real-Time Detection and Application-Layer Attacks

Application layer attacks like Slowloris thus pose a threat now than ever before because such attacks take a lot of minimal resources to execute but affect a particular service. In the real-time detection of Slowloris attacks, the ML models were analyzed by **Wibowo et al.** [16]. In their research, they used ML techniques like Gradient Boosting and Random Forest algorithms to detect and differentiate between the permitted and prohibited Traffic and showed that the use of this approach led to few chances of attack going unnoticed. Predicting limited yet devastating attacks in real time is of paramount importance for preserving the availability of important Web sites and services.

But, in the development of ML models for real-time detection system there is a problem of computational training and deployment of the models. It implies that the detection of DDoS attacks should progress in effectiveness and flexibility to meet the increasing prospects of the sophistication of these attacks. **Anley et al.** [23] have dealt with this challenge under their proposed transfer learning-based approach to DDoS detection. Here, transfer learning helps models carry forward what they had learnt from the other environments in order to learn faster than before to adapt to the new threats. That is especially valuable in the context of dynamic and fluid attack surfaces that are constantly in flux as the method of integrating the fresh data does not require reinventing the whole detection models.

2.3.7 Collaborative Detection and Cross-Network Defense

Because DDoS attacks are continually evolving and becoming more massive and sophisticated, there is more emphasis on coordination within networks and organizations. A collaborative detection system was proposed by **Wagner et al.** [24] that intends to use Internet Exchange Points (IXPs) as sensors. IXP members can exchange traffic data to determine when significant DDoS attacks are taking place as individual IP networks would not be able to do themselves. In particular, this approach comes handy when it comes to the amplification attacks, during which attackers use publicly available servers to boost traffic load.

2.3.8 AI and Ensemble Learning for DDoS Detection

The incorporation of AI technologies in DDoS detection systems has provided outstanding enhancements in the identification systems and the time taken. **Jaszcz, A.** [25] proposed an ensemble learning system which integrates decision tree, SVM, and k-Nearest Neighbors (k-NN) algorithms. Ensemble learning offers an improvement in the general performance of the detection systems, especially in real-time interims where both precision and pace are crucial. Due to the adaptive nature of the parameters of the model, it achieves a high level of detection accuracy with a minimum false alarm rate when the network conditions in question are constantly changing.

Moreover, it is also proved that DL models also have a promising impact in improving the detection of DDoS attacks in cloud environments. **Nadeem et al.** [26] proposed a cloud-based distributed intrusion detection system that incorporates anomaly detection as well as signature-based techniques for various cyber threats. Because when measuring to handle huge traffic common in cloud environments it is one of the best tools to take to counter both brute force and DDoS attacks. The advantage of having several kinds of systems working in parallel is that it in sum offers a much better way to counteract multiple kinds of attacks.

2.3.9 Efficient Detection in IoT with ML

Najafimehr, M. [27] work on a DL-based approach for IoT DDoS detection employs the use of a number of hybrid feature selection techniques, taking into consideration the limited and constrained environment that IoT can present. High detection accuracy was, therefore, complemented with low computation which was important in the study. This is particularly important to guarantee that DDoS detection solutions can be implemented where IoT devices are limited in resources but exposed to attacks.

Likewise, **Kalutharage, C.S** [28] studied the application of decision trees and ensembling for DDoS detection in IoT; the goal was to achieve low False Positive (FP) while high True Positive Detection (TD). They also established that ensemble learning methods that integrate several classifiers are very efficient in the task of differentiating regular traffic from a DDoS attack traffic in the Internet of Things environments.

2.3.10 Transfer Learning and Cross-Context Adaptation

Alshahrani, M.M [29] employed transfer learning in a more general sense of that, related to detecting DDoS attacks in the constantly changing environment. They also examined how the transfer learning can be used in the cross-context settings, that is, how the models learned in one environment can perform when transferred to the new network conditions; this makes their performance in the identification of the new attack types more effective.

2.3.11 The Future of ML in DDoS Detection

The explored ML and DL approaches in DDoS detection remain limited, but, on the whole, the outcomes are encouraging. In [30], **Maksimović et al** proposed the new method of enhancing the DDoS detection systems by implementing the Triple Modular Redundancy (TMR) with ML algorithms. This method minimizes chances of false positives as it only targets legitimate traffic anomalies as possible attacks. This method advances the practical fundamentals of combating DDoS by using ML supplemented by conventional redundancy methodologies.

It is now possible to confirm that ML and DL are playing a crucial role in fighting DDoS attacks. Since the emergence of these attacks is becoming more complex and frequent, detection systems should develop as well. Real time detection and AI based techniques along with the collaborative defense strategies seem to provide ideal solution. Nevertheless, a number of important problems persist, most significantly concerning the computational aspect and the issue of flexibility. The future work on this matter will be directed to the development of more efficient methods for the improvement of the existing detection systems concerning the issue of scalability and adaptability.

2.4 Knowledge Gap.

That being the case, fundamental gaps persist when it comes to applying ML in DDoS detection. It should be recognized that the majority of works concern current, labelled datasets, while real traffic is much richer and more diverse. Little has been done to investigate how predictive models continue to learn usable patterns from emergent, unorganized, or noisy data types in real-time scenarios [1][5].

Furthermore, we will be testing multiple ML algorithms to search for the best algorithm which will aid in the detection of DDoS attacks. Additionally, there will be criterion for the algorithms which are, firstly, accuracy of detecting DDoS attacks. Secondly, the precision of the detection of the algorithms. Thirdly, recall in which the algorithm can identify a DDoS attack from other normal traffic. Fourthly, F1–Score. Finally, false positive rates, which can be reduced using the ML techniques and improve the detection rates.

Our project will make the algorithm learn from datasets to differentiate between normal traffic spikes and false positive from DDoS attacks. This will change the perception of choosing criteria of the accuracy, which will consider other factors such as: identifying DDoS attacks from other normal traffic, additionally, this will benign traffic that could potentially be considered as a false positive in a different system.

2.5 Chapter summary

This chapter discusses the current protection strategies against DDoS attacks, with a focus on the use of ML in improving the detection systems. This section classifies DDoS attacks according to their sources before elaborating on the effects of such attacks and the difficulties with distinguishing between genuine and malicious traffic. In this paper, the traditional approaches like signature and anomaly-based detection are analyzed alongside their weakness. The literature review also explores the developing attacks and the incorporation of ML approaches, DL and SDN frameworks.

Ref	Authors (Year)	Aim	Classifiers	Measurement	Dataset	Preprocessing	Feature Extraction	Result	Strength	Weakness
[30]	Maksimović et al., (2024)	To improve threshold determination in DDoS IDS	Stacking ensemble, TMR	Accuracy: Optimised_TMR 99.6%, KNN 99.3% F1 Score: Optimised_TMR 97.0%	Kaggle dataset: Military network intrusions	Data cleaning	41 quantitative features	Improved accuracy in DDoS detection using TMR; avoided bad over-voting problem.	Combines top algorithms (Decision trees, Naive Bayes, SVM, k-NN, Logistic Regression) to avoid bad over-voting in TMR; improved threshold determination.	Potential slow performance due to multiple algorithms; execution time and scalability could be an issue when using larger datasets.
[16]	Mohamad Hegar Sukmana Wibowo, (2024)	Real-time detection and prevention of Slowloris DDoS attacks	Support Vector Machine (SVM), Neural Networks, Random Forest, Gradient Boosting	N/A	Custom dataset (Slowloris attack traffic)	Data cleaning and normalization	Connection duration, request rate, packet size, number of open connections	Real-time detection and prevention of Slowloris DDoS attacks using machine learning. High accuracy achieved, particularly with Gradient Boosting.	High precision, recall, and F1 score; models are scalable and can be deployed on standard server hardware	Limited to Slowloris attacks; requires continuous updates with new training data to maintain effectiveness.
[23]	Mulualem Bitew, Angelo Genovese, Davide Agostinello and Vincenzo Piuri (2024)	robust defense mechanisms to safeguard network infrastructure availability and integrity.	Deep Learning models (CNN, VGG19,DNN,CONV18)	Accuracy: CNN 99.87% VGG19 99.99% (Their Model)	CIC-DDoS2019 CSE-CIC-IDS2018 UNSW-NB15 KDDCup’99	Data normalization, augmentation	Automated feature learning through CNN	employed custom CNN models (Conv4, Conv8, and Conv18), along with pretrained models (VGG16, VGG19, and ResNet50), trained on cybersecurity benchmark datasets, including KDDCup’99, UNSW-NB15, CSE-CIC-IDS2018, and CIC-DDoS2019.	Using Different datasets to see the result on different data types.	Local detection thresholds may miss attacks; complexity in cross-site coordination
[29]	Alshahrani, M.M. (2023)	Develop an SDN framework for smart cities	XGBoost	XGBoost achieved an accuracy of 99.99%, precision of 97%, recall of 99%, an F1 score of 98%, and a false-positive rate of 0.05.	BoT-IoT dataset	Cleans and formats traffic data using the pcap utility, preparing it for machine learning analysis.	Key network traffic features like packet sequence number, timeframe stats, and connection counts are extracted for anomaly detection.	High accuracy in detecting DDoS attacks	High accuracy, enhanced security	Complexity of implementation
[20]	NOE MARCELO YUNGAICELA-NAULA, CESAR VARGAS-ROSALES, AND JESUS ARTURO PEREZ-DIAZ (2023)	Detect IoT DDoS attacks using a hybrid feature selection model	XGBoost, Random Forest, KNN, SVM	Accuracy: KNN 99.971% SVM 99.774%	CIC IDS 2017, CIC IoT 2023	Data cleaning and filtering applied to remove irrelevant information and normalize datasets.	A hybrid feature selection algorithm identified key network traffic features for DDoS detection.	Achieved high accuracy and recall rates	High accuracy, explainable model, improved feature selection	Limited to datasets used, may not generalize to all scenarios
[11]	Tariqul Islam, Md. Ismail Jabiullah, Dm. Mehedi Hasan Abid (2023)	Develop an effective method for detecting and preventing DDoS attacks using artificial intelligence techniques. They focus on achieving over 96% accuracy in detecting DDoS threats by leveraging machine learning and deep learning methodologies.	Artificial Neural Networks (ANN) & Deep Learning (DL)	N/A	The paper does not specify a particular dataset name.	Data Reduction and Data Cleaning	Target Feature Distribution and Data Analysis	The study identifies seven sub-categories of DDoS attacks and utilizes supervised and unsupervised learning methods. The evaluation showed high precision and recall rates for various attack types, with the SYN flood attack demonstrating the highest accuracy.	- High detection accuracy - Comprehensive analysis of different DDoS attack types and effective classification methods. - Use of advanced techniques such as deep learning and data preprocessing for improved performance.	- The reliance on specific data characteristics may limit applicability to diverse real-world scenarios.
[17]	Pavitra Modi (2023)	Develop an efficient machine learning method for detecting DDoS attacks in IoT devices. It proposes a hybrid feature Selection algorithm that selects the most relevant features and utilizes an XGBoost model, achieving high accuracy and recall on various datasets.	Hybrid feature selection algorithm combined with XGBoost	XGBoost model, the results of which are explained using feature importances. Our model attains an accuracy of 99.993% on the CIC IDS 2017 dataset and a recall of 97.64 % on the CIC IoT 2023 dataset.	CIC IDS 2017 & CIC IoT 2023	CIC IoT 2023 Dataset: The “label” column was transformed to represent benign traffic as 0 and any type of DoS or DDoS packet as 1. CIC IDS 2017 Dataset: The dataset was cleaned to remove packets with ‘NaN’ and ‘infinity’ values.	The CIC IoT 2023 dataset used the Standard scaler, and the CIC IDS 2017 dataset's non-categorical features were scaled after removing categorical variables. The hybrid feature selection algorithm calculated Pearson correlations, selected features based on thresholds, evaluated them with Spearman and Kendall coefficients, and ranked features using information gain, selecting those above the mean.	-The proposed model significantly improves the detection of DDoS attacks in IoT devices with high accuracy. -The model provides interpretable results through feature importance analysis.	-Achieves very high accuracy, indicating effectiveness in detecting DDoS attacks. -Offers explainability, allowing users to understand which features contribute most to detection.	The dependence on specific datasets may limit generalizability to other types of DDoS attacks or IoT environments.

Ref	Authors (Year)	Aim	Classifiers	Measurement	Dataset	Preprocessing	Feature Extraction	Result	Strength	Weakness
[21]	Mahmood A. Al-Shareeda, Selvakumar Manickam, Murtaja Ali Saare (2023)	Analyze and compare various machine learning (ML) and deep learning (DL) techniques for detecting Distributed Denial of Service (DDoS) attacks, highlighting when to use each approach in Intrusion Detection Systems (IDS).	Machine Learning (ML) Deep Learning (DL)	Accuracy: SVM 95.24% Decision Tree 99.93% K-mean clustering 99.69%	The article does not specify a particular dataset	Data Cleaning and Normalization	The paper does not provide a Feature Extraction	The paper indicates that both ML and DL techniques show promising results in DDoS attack detection.	-The paper provides a detailed comparison of ML and DL techniques for DDoS detection. -It emphasizes the effectiveness of anomaly-based IDS, which can detect new, previously unknown attacks.	Absence of a concrete dataset makes it harder to replicate results or validate findings.
[19]	Deepak Kumar, R.K. Pateriya, Rajeev Kumar Gupta, Vasudev Dehalwar, Ashutosh Sharma (2023)	Develop an LSTM model for detecting DDoS attacks in network traffic, targeting higher accuracy than traditional machine learning methods using the CICDDoS2019 dataset.	Long Short-Term Memory (LSTM) NDAE	Accuracy: NDAE 99.6%	CICDDoS2019	Cleans the raw network traffic data by removing irrelevant information and normalizing features to ensure consistency, which enhances the LSTM model's performance during training.	Selecting relevant features that enhance DDoS attack detection efficiency. While the LSTM model can automate feature selection, this initial step is beneficial. The dataset is then divided into training and testing sets for effective model validation. Share	The proposed LSTM model demonstrated high accuracy in identifying DDoS threats from network traffic packets. The model outperformed traditional machine learning techniques on the CICDDoS2019 dataset.	-Achieving 98% accuracy indicates strong performance in detecting DDoS attacks. -The use of LSTM allows for effective feature selection and extraction automatically, which is beneficial over traditional methods.	-The model's performance is reliant on the quality and representativeness of the CICDDoS2019 dataset.
[28]	Kalutharage, C.S.; Liu, X.; Chrysoulas, C.; Pitropakis, N.; Papadopoulos, P. (2023)	DDoS attack identification in IoT networks	Explainable AI (XAI)	Accuracy: DT 97% RF 98%	USB-IDS dataset (University of Sannio,	CICFlowMeter for bidirectional traffic	Application-layer, volumetric, TCP state exhaustion features	Superior detection performance compared to state-of-the-art methods	High detection accuracy and explainability	Limited to specific attack types; dataset reliance
[27]	Mohammad Najafimehr, Sajjad Zarifzadeh, Seyedakbar Mostafavi (2023)	Detection of DDoS attacks	Random Forest, SVM, CNN, LSTM	Accuracy: RF 99.9% SVM 93%, CNN 99%	CAIDA, ISCX, CICDDoS2019	Normalization, Sampling, Feature Engineering	Packet size, Interval, Bandwidth	Effective DDoS detection on IoT and cloud	High accuracy in specific datasets, well-documented models	Lacks real-world deployment, limited datasets used
[14]	Jonas Bushart, Christian Rossow (2022)	Assess resilience of DNS against application-layer DDoS	Anomaly-based Filtering	IXPs can deploy our scheme and drop attack traffic to reasonable query loads at or below 100k queries per second a false positive rate of 1.2 % to 5.7 % (median 2.4 %).	Real-world dataset from a large ccTLD operator	NetFlow data collection, learning past resolver behavior	Behavioral profiling based on query loads	Defense against application-layer DDoS on DNS	Effectively filters attack traffic before it reaches servers	Relies on prior behavioral training; subject to concept drift
[13]	Muhammad Waqas Nadeem, Hock Guan Goh*, Vasaki Ponnusamy and Yichiet Aun (2022)	Securing the Software defined Network (SDN) from DDoS attacks.	Random Forest (subset: Recursive feature elimination)	Accuracy: RF 99.97%	NSL-KDD	Data Cleaning, Normalization/Scaling, Feature Selection, Encoding Categorical Data, Splitting Dataset, Balancing the Dataset, Cross-Validation.	Source and Destination IP Addresses, Average Packet Size, Idle Time. Flow Duration, Number of Bytes per Flow: The total number of bytes transmitted during a flow.	Rf is considered has the greatest accuracy	Achieves higher accuracy than any other ML technique	Using only one machine learning technique.
[18]	Biswajit Mondal, Chandan Koner, Monalisa Chakraborty, Subir Gupta (2022)	Eliminating DDoS attacks from internet of things (IoT).	Logistic Regression (LR), Naive Bayes (NB), SVM, K-NN, Decision Tree (DT), and Random Forest (RF)	Accuracy: LR 97% NB 44% SVM 85% K-NN 97% DT 98% RF 98%	NSL KDD	Data Cleaning, Normalization/Scaling, Feature Selection, Encoding Categorical Data, Splitting Dataset, Balancing the Dataset,.	Source and Destination IP Addresses, Average Packet Size, Idle Time. Flow Duration, Number of Bytes per Flow: The total number of bytes transmitted during a flow.	Many algorithms work but Random Forest is the most accurate	High percentage of accuracy.	Many protocols, computer limitations.
[26]	Jaszcz & Połap (2022)	Develop a framework (AIMM) for detecting DDoS attacks using AI methods.	Neural Network, k-nearest neighbors (k-nn)	Accuracy: 99.5% (Algorithm1 from the research no name for it)	BOUN DDoS Dataset	Time-based data aggregation (0.5-second windows) reduced data size while retaining critical information.	Four features were extracted: unique destination IPs, max IP occurrences, packet size sum, and number of packets.	High accuracy in detecting DDoS attacks using a hybrid AI approach.	Flexible framework, high accuracy, combines multiple AI methods, can work with small datasets	Not specified in detail, potential issues with handling imbalanced datasets

Ref	Authors (Year)	Aim	Classifiers	Measurement	Dataset	Preprocessing	Feature Extraction	Result	Strength	Weakness
[22]	Muhammad Ismail, Hameed Hussain, Ayaz Ali Khan, Ubaid Ullah, Muhammad Zakarya, Aftab Ahmed, Mushtaq Raza, Izaz Ur Rahman, Muhammad Haleem (2022)	Develop an effective framework for classifying and predicting Distributed Denial of Service (DDoS) attacks using machine learning techniques.	Random Forest & XGBoost	Accuracy: Random Forest: 89% XGBoost: 90%	UNSW-nb15 dataset	-Cleaning Irrelevant Data -Handling Missing Values	transforming raw data into a format suitable for machine learning algorithms.	The proposed models showed significant improvement over existing research, which reported accuracies of approximately 85% and 79%.	- Utilization of modern datasets (UNSW-nb15) instead of outdated ones. - High accuracy rates for both models. -Comprehensive framework for DDoS attack classification and prediction.	- The study relies on specific datasets, which may not cover all types of DDoS attacks. - Potential limitations in real-world applicability without further validation.
[12]	Lee et al. (2022)	Propose an autonomous defense system for IoT	Convolutional Neural Network (CNN)	Accuracy: CNN 99.5%	Packet traffic and features	Cleans packet-level data and standardizes it for analysis by a CNN.	Extracts features such as packet size, rate, source/destination IP, and protocol type to distinguish between normal and attack traffic.	Effectively distinguishes DDoS attacks	High accuracy, effective detection	Potential computational overhead
[15]	Rami J. Alzahrani, Ahmad Alzahrani (2021)	implement different Machine Learning (ML) algorithms in WEKA tools to analyze the detection performance for DDoS attacks	(K-NN), (SVM), (NB), (DT), (RF) And (LR)	Accuracy: Decision Tree (DT) 99% Random Forest (RF) 99% Time Decision Tree (DT) 4.53s Random Forest (RF) 84.2s	CICDDoS2019	Converted dataset attributes	Various features depending on model	(DT) And (RF). Both get 99% Accuracy But (DT) better because it takes 4.53s and RF takes 84.2s	Using six different algorithms to see which one is the best	Some models may not generalize well across datasets
[24]	Daniel Wagner, Daniel Kopp, Mathhias Wichthuber, Christoph Dietzel, Oliver Hohlfeld, Georgios Smaragdakis and Anja Feldmann (2021)	design and evaluate a collaborative architecture that allows participant mitigation platforms to exchange information about ongoing amplification attacks	eleven IXPs (Internet exchange point) that operate in three different regions.	N/A	RADb	Flow-level aggregation (IPFIX), sampling (1:10k)	1,106 features derived from flow-level traces (e.g., traffic volume, duration)	Coordination in detecting and mitigating attacks benefits smaller network infrastructures, potentially detecting over 80% more attacks. Internet infrastructures like IXPs can drop attack traffic close to reflector locations, reducing potential harm.	Using a lot of IXPs in different regions	Dependence on quality of training data, complexity in architecture design
[25]	Muhammad Nadeem et al. (2021)	Develop a framework using IDPS for cloud security	Machine learning methods, DL	Accuracy: Random Forest 99.99% CatBoost 99.99%	Not specified	Filters incoming traffic using signature-based and anomaly-based IDS systems to categorize traffic.	Extracts features such as packet flow, source/destination IPs, and port numbers to detect attack patterns.	Proposes network topology and prevention methods	Overview of IDS types, detailed architecture	Lacks specific data and metrics

Table 1. Summary of Literature Review

Chapter 3: SPMP and Requirement Specification

3.1 Project Overview

This section is about Software Project Management Plan (SPMP), and we will discuss the six major sections of the project. Section one is the project's Overview which includes the Purpose of our applied research system (ML Detect), scope, objectives, assumptions, constraints, risks, project deliverables schedule, budget summary, evaluation of the plan, and references. Section two, Project Organization which is including external interfaces, internal interfaces, roles, and responsibilities. Section three is about Managerial Process Plans, and this section consists of the startup plan, work plan, project tracking plan, and project closeout plan. Section four is about Technical Process Plans, and the plans are about process models, methods, tools, techniques, infrastructure, and product acceptance. Section five and it's about Supporting Process Plans, and it includes the documentation that is necessary for this chapter. It includes any additional plans that are a must to keep the project on the right track and will lead us to achieve all the requirements.

3.1.1 Purpose, Scope, and Objectives

The goal of the ML Detect is to detect DDoS attacks in datasets of network traffic using simple ML algorithms. The objective is to quantify, to what extent ML can identify DDoS patterns from past network information. While the goal of ML Detect is to be able to detect adversarial attacks in various scenarios, it is only applicable to predict network traffic datasets that are obtained from the public domain within a research laboratory environment. In the case of ML Detect, a simple set of classification models including the DT will be employed to identify moving traffic as either normal or potentially carrying DDoS. In the next sections, we will concentrate on the models' training and their evaluation with small dataset sizes to compare their DDoS detection performance. The outcome is a report of the results gotten from the models and how the accuracy and performance was done hence advancing the knowledge of DDoS using ML.

3.1.2 Assumptions, Constraints, and Risks

The table below will discuss the assumption, constraints, and potential risks that may arise in the project.

ASSUMPTIONS	• The team utilizes publicly available datasets for DDoS detection. • Sufficient knowledge of Python and ML libraries (e.g., Scikit-learn, Pandas). • The project will focus only on analyzing dataset accuracy, not real-time detection.
CONSTRAINTS	• Limited availability of real-world DDoS datasets. • Limited access to real-time network traffic data. • Limited variety in datasets could reduce model generalization.
RISKS	• Dataset quality may impact model accuracy.

Table 2: Assumptions, constraints, and risks table

3.1.3 Project Deliverables

By the end of the project, we will deliver to our supervisor and evaluators the following,

- Chapter 1: Introduction.
- Chapter 2: Background and literature review.
- Chapter 3: Software Project Management Plan (SPMP) and Requirement Specification.
- Chapter 4: Methodology and Design.
- Chapter 5: Implementation.
- Chapter 6: Conclusion.

3.1.4 Schedule and Budget Summary

For our project, the budget will be 200 SAR for google colab which helped us run the training and testing stages of the ML algorithms, which led to the enhancement of the project overall. for the other aspects of the project, no budget was used to accommodate the project.

3.1.5 Evolution of the Plan

IEEE Std 1058-1998 standard will be used in our project, and we may change it if there is any need for another standard.

3.1.6 References

1058-1998 - IEEE Standard for Software Project Management Plans - IEEE Standard. (2019). Retrieved 19 October 2019, from https://ieeexplore.ieee.org/document/741937.

3.2 Project Organization

3.2.1 External Interfaces

Supervised by Dr. Dhiaa Musleh, this work is mainly concerned with the detection of DDoS using a ML algorithm with real–time traffic analytical systems.

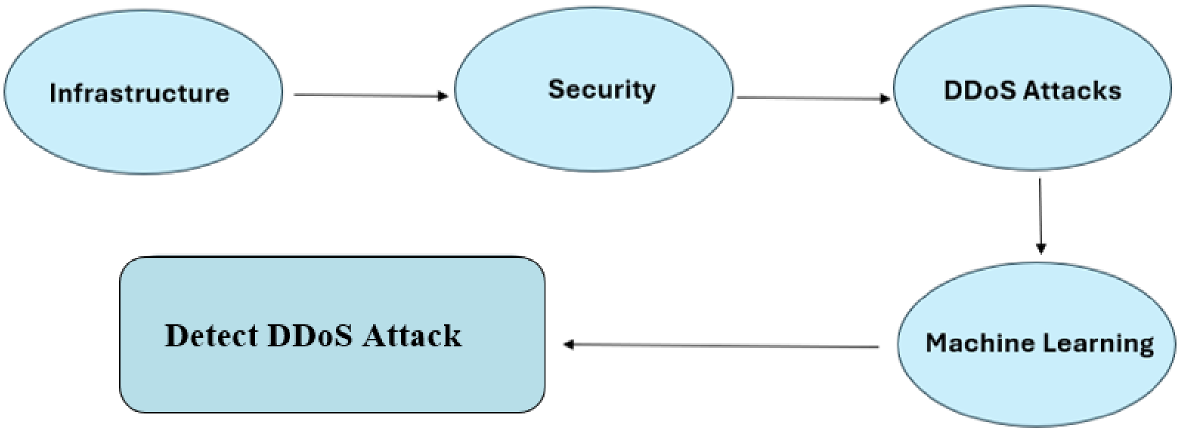


Figure 1:External Interfaces

3.2.2 Internal Structure

In the distributed roles structure presented below, Nasser is the overall project manager, and Rayyan, Aseel, Ibrahim, and Noor are responsible for various tasks committed to their individual and team assignments.

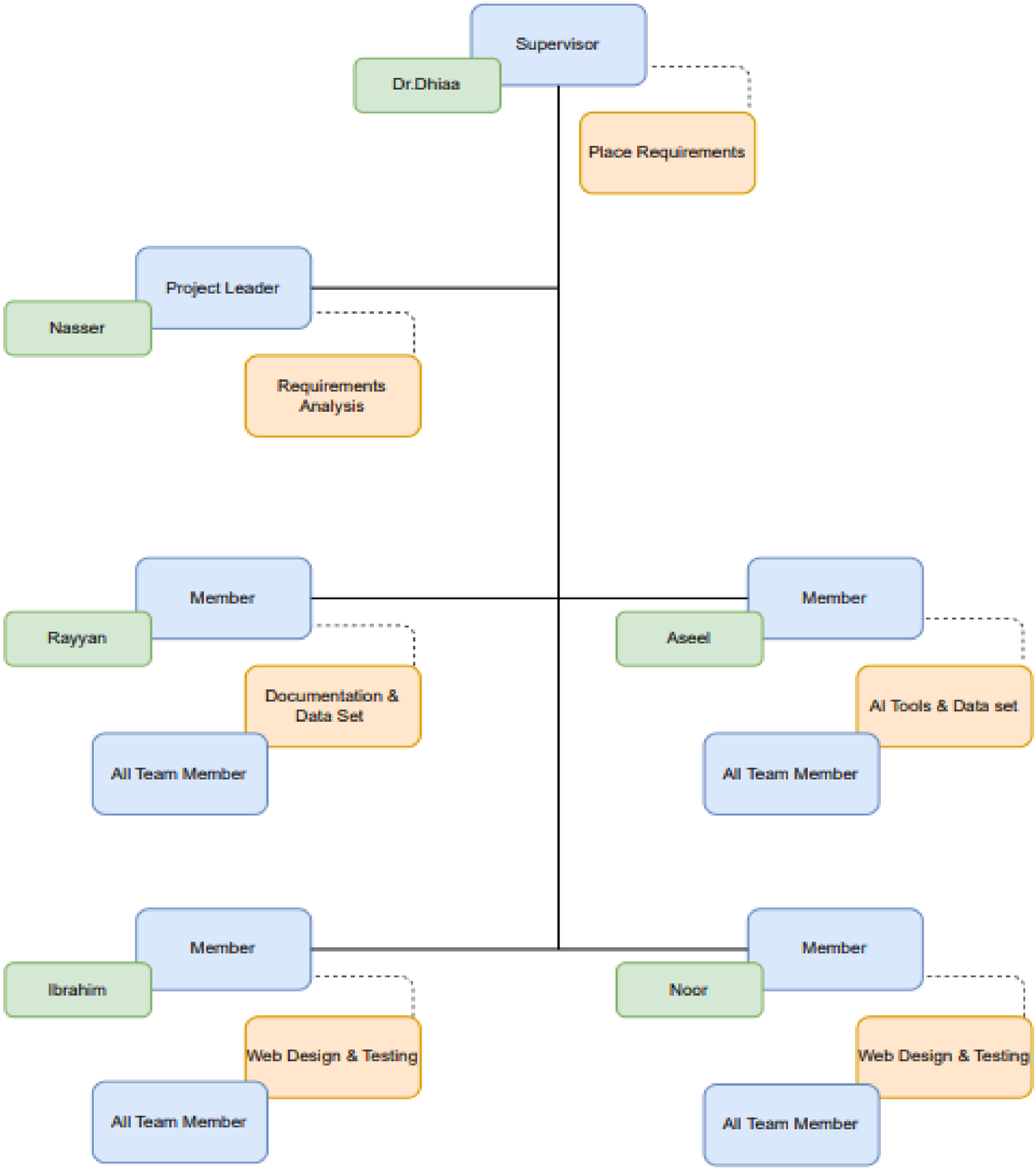


Figure 2:Project Organization with Internal Structures.

3.2.3 Organizational Structure

The table below helps us understand the structure of the project as it shows distribution of roles and responsibilities among the team members. Ever since the leader's function is to observe and coordinate the outcome of the work and regulate the interactions between the team members, he would be the most important in this axis.

Leader Name	Participated Members	Role	Responsibilities
Nasser A. Alhajri	**All**	**Project Leader**	• **Planning the project flow.** • **Managing the timeline.** • **Distributing tasks to team members.** • **Contacting and coordinating with the supervisor.**
Aseel A. Alrudayni	**All**	**AI Tools & Data Set**	• **Selecting and configuring ML tools.** • **Handling dataset preprocessing and model training.**
Rayyan B. Al Nahwi	**All**	**Documentation & Data Set**	• **Collecting data for training models.** • **Managing project documentation.**
Noor A. Hadari	**All**	**Web Design & Testing**	• **Assisting in system testing.** • **Conducting system-level tests to ensure performance and reliability.**
Ibrahim A. AlHusaini	**All**	**Web Design & Testing**	• **Designing the project webpage.** • **Conduct testing and ensuring the system functions correctly.**

Table 3 Organizational Structure

3.3 Managerial Process Plans

The plans for project start-up, risk management, project work, project tracking and project closeout are defined in this section.

3.3.1 Startup Plan

3.3.1.1 Estimation Plan

The cost estimation for the project is 200 SR since the project requires paid software or special hardware and the fact that each of five group members owns a personal computer/laptop, this will eliminate hardware cost estimation Regarding the timeliness, the estimation is 32 weeks, 5 working days with weekly/biweekly meetings with a supervisor, and 1 up to 4 sessions a week for members via Discord; In addition, the textual communication will be conducted through the WhatsApp application. Moreover, the resources needed for the project will be PyCharm (Python) Programming language or Google colab and ML tools, also we need a dataset to perform various ML algorithms on. The estimations were based on team experience, certifications and previous projects.

3.3.1.2 Staffing plan

Our team consists of five senior students specializing in computer science passionate about completing the project by the end of this academic year. All the team members study at Imam Abdulrahman bin Faisal University (IAU). All members should have experience and skills in each of AI, Cybersecurity, data mining, programming skills, writing skills, decision making, and communication skills, etc. The human resources, time, and skill level of the team members required to complete each step are shown in the table below.

Project Phase	Duration	Human Resources	Skill Level
Project Proposal	Three weeks	All Team Members	Low
DATA COLLECTION (DATASETS, ARTICLES, ETC)	Four weeks		Medium
PROJECT MID REPORT	Three Weeks		Medium
CH3 (SPMP)	Two Weeks		Medium
CH4 (Methodology and Design)	Eight Weeks		High
CH5 (Implementation & Testing)	Six Weeks		High
Final Report	Six Weeks		High

Table 4 Staffing plan

3.3.1.3 Project staff training plan

Every member of our team must be properly educated in ML, Python Programming language, AI, and cybersecurity. The training will be completed through self-study using online resources, and our team attending a data mining course taught by Dr. Irfan Ullah that includes ML lessons. The type of training and the method used to get it are shown in the table below.

Type of Training	Personnel to be Trained	Method Used
Principles of AI	All Team Members	AI course by Dr. Imran and online resources
Data mining technics		Data mining course taught by Dr. Irfan Ullah and online resources
knowledge about Cybersecurity		Online resources and previous college courses
Selected Topics		University course taught by Dr. Irfan Ullah
Python Programming language (ML)		Online resources and previous college courses

Table 5 Project staff training plan

3.3.2 Work plan

3.3.2.1 Work activities

The project activity name, expected duration, deliverables, and status are shown in the table below.

Task No.	Activity Name	Expected Duration	Deliverables	Status
1	Project Proposal	One week	Document + Website	Completed
2	DATA COLLECTION (DATASETS, ARTICLES, ETC)	One week		Completed
3	PROJECT MID REPORT (CH1, CH2)	Three Weeks		Completed
4	CH3 (SPMP)	Two Weeks		Completed
5	CH4 (Methodology and Design)	Three Weeks		Completed
6	CH5 (Implementation & Testing)	Three Weeks		Completed
7	Final Report	Four Weeks		Completed

Table 6 Work activities

3.3.2.2 Schedule allocation

Table below is a Gantt chart demonstrating the estimated duration for each task

Tasks	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Task 1	▬															
Task 2		▬														
Task 3			▬	▬	▬											
Task 4						▬	▬									
Task 5								▬	▬	▬						
Task 6											▬	▬	▬			
Task 7														▬	▬	▬

Table 7 Schedule allocation

3.3.2.3 Resources allocation

Table below shows the skills and resources needed for each task; resources can be human, non-human, and software

Task	Activity	Skills	Human resources	Non-human resources	Software
1	Discussing the proposal	Communication skills Time management	All team members and the supervisor	Computers	Discord, WhatsApp
2	DATA COLLECTION (DATASETS, ARTICLES, ETC)	Technical writing, Communication skills, Time management, Teamwork,		Computers	MS WORD, Google sheets, Discord, Kaggle
3	PROJECT MID REPORT	Writing skills, Analytical skills, Time management, Editing skills,		Computers	Discord, MS WORD, Google sheets
4	CH3 (SPMP)	Technical writing, Project management, Analytical skills, Time management		Computers	MS WORD, Discord, WhatsApp
5	CH4 (Methodology and Design)	Problem-solving, Analytical skills, technical writing, Design skills		Computers	MS WORD, Figma, Discord and HTML
6	CH5 (Implementation & Testing)	Programming, Debugging, Analytical skills, Time management		Computers	GitHub, VSC, Kaggle, Google colab, Excel and PyCharm

7	Final Report	Writing skills, Editing skills, Time management, Attention to detail,		Computers	MS WORD, Discord , Google Sheets

Table 8 Resources allocation

3.3.2.4 Budget allocation

The only hardware needed for the project is a computer. Each team member has one; For our project, the budget will be 200 SAR for google colab which helped us run the training and testing stages of the ML algorithms, which led to the enhancement of the project overall. for the other aspects of the project, no budget was used to accommodate the project.

3.3.3 Project Tracking plan:

3.3.3.1 Requirements management

The Leader must collect all chapters and analyze them. Then, if necessary, team members will discuss what they can and cannot change to the project. If the project receives his approval, he will submit it to the supervisor, who will then instruct changes to the project, team members will change it, and if the supervisor approves the changes. Then a new version of the software requirement specification document will be created.

3.3.3.2 Schedule Control

- All milestones are supplied in the course syllabus.
- Milestones are broken down into tasks that are then distributed amongst members concerning the level of demand for each job.
- Project milestones are uploaded to the supervisor Dr. Dhiaa Musleh via OneDrive for feedback.
- All members check schedules, submit tasks, and track milestones through OneDrive, and functions through Google Sheet.

3.3.3.3 Quality Control

Quality control requires every leader and the team member to analyze and consider different components of the delivery several times before delivering the file to the supervisor in order to ensure that the delivery is correct. And there are no mistakes, when conditions are met, they are met.

- Certain milestones must always be prepared according to templates, and any additional instructions supplied by the project supervisor must be followed.

3.3.3.4 Reporting

- Team communication is facilitated by WhatsApp messenger and the usage of Discord.
- Essentially, Google sheets are used to record all the milestones and tasks, relevant documents and resources
- Milestones are requested by the Supervisor from OneDrive
- Meeting is held on weekly basis with project supervisor Dr.Dhiaa concerning the submitted work and future deliverables

3.3.3.5 Project Metrics

Project metrics are a strategy for measuring project quality and productivity; they also define project performance. These project metrics are shown in Table below.

Metrics	Questions	Comments
Time	On what stage is the work done regarding the schedule?	In section 3.2.1, project progress is checked against the schedule plan
Value	Were all the situation project objectives achieved? And to what extent?	The project will meet the requirement specifications (SRS).
Quality	Were the issues regarding the quality of the project solved?	After testing the system, some defects might occur, all of these defects and malfunctions will be restored, and the system should be working as expected.
Scope	Did the scope of the project change during the course of the project?	The concept may differ from the planning phase due to changes in the requirements.

Table 9 Project Metrics

3.3.4 Risk Management Plan

Many risks needed to be tamed to ensure proper workflow and maintainability, hence detect any kind of risk that can arise. When a risk is present, we need to solve and deal with it to remove stress from the workload and remove unnecessary and unwanted events to occur. Additionally, risks can delay the progress of the project's goal. In order to deal with risks, the group leader has set a well-defined process which covers all shortage from each member. In the table below you can see all risks:

Number	Potential Risk	Probability	Effects	Action	Impact
1	Inaccurate Dataset.	Medium	High	Find a suitable Dataset for the project.	Degrade the quality of the project.
2	Shortage of time.	High	High	Remove unnecessary workloads and divide the work equally.	Miss important deadlines.
3	The concept of the project is not clear to everyone.	High	High	Write a brief description so all members can go back to for understanding.	Work is not consistent and poor.
4	Inaccurate Time	Medium	Low	Diagram that shows all due dates and deadlines.	Missing vital deadlines.
5	Technical risk	Low	Medium	Teamwork to overcome these issues.	Delay work.

Table 10 Risk Management Plan

3.3.5 Project Closeout Plan

The project closeout plan is one of the last things that need to be done while managing a project. As team members put the final touches on all in order and complete all the deliverables that are needed and wanted, all the objectives and goals have been accomplished and the project is finished. In the end, the following phases have been concluded and completed:

1. Introduction
2. Background and Literature Review
3. SPMP and Requirement Specification

4. Methodology & Design
5. Implementation
6. Conclusion and Future Work

3.4 Technical Process Plan

This plan centers on gaining and examining the efficiency of different ML models to have the right one to use in detecting DDoS attacks. Instead, we want it to be as accessible as possible and involve different users.

3.4.1 Process Model

We have decided to use the Iterative Model for development since it can be implemented in several cycles. An iterative model is simply breaking down the tasks into smaller manageable chunks that would be processed in iteration. This phase is broken down into the following:

1. **Data collection and preparation:** we'll take a dataset that is available to all individuals. The data will be cleaned and preprocessed to avoid creating confusion during analysis. This encompasses dealing with missing values, normalizing features and feature engineering (selected features include size of packet and connection rate). The outcome is that it will help to receive a clean ready-made data set ready for training and evaluating the models.

2. **Algorithm Selection and Testing:** Choose multiple algorithms for ML to find out their accuracy in identifying DDoS attacks. We'll try out one of the models which is more traditional analytics tools such as Random Forests and Support Vector Machine (SVM). These will be trained separately using simple ML tools like scikit-learn for simple models. To get a reliable measurement, we shall be using cross validation in the evaluation of models.

3. **Performance Evaluation:** we will take the accuracy and precision and false positive rates to compare models, more evaluation metrics could arise in the testing phase. We can add another kind of performance metric which is the ease of implementation and computation time, both are very important in the field of cybersecurity.

4. **Iterative Improvement:** improving different models to enhance the overall performance metrics and to ensure that all algorithms had the same measurements. Moreover, this will give more accuracy to the algorithms.

5. **Final Model Selection:** choosing the final model will be based on multiple factors not just the accuracy of the algorithm but also the false positive rate, this will add a level of confirmation to the choice to make it unanimous. The final product will be a well-defined and tested model that covers all factors.

3.4.2 Methods, Tools, and Techniques

Supervised learning such as Random Forest and SVM will be implemented so as to test which approach will be best suited for DDoS detection. Both the models will then be trained and tested with the help of open-source tools primarily in Python and some tools from R and will use few libraries such as scikit-learn, TensorFlow, Keras.

3.4.3 Infrastructure

Hardware	For training the models, we will have the luxury of using personal laptops or desktops with minimum RAM of 8 GB.
Software Environment	We will define the environment where we are going to perform data preprocessing with PyCharm (Python) and the libraries we are going to use as well as Jupyter Notebooks to maintain the same working environment for all team members.

Table 11 Infrastructure

3.4.4 Product Acceptance

Since we're focusing on testing models, the product will be the best performing algorithm, chosen based on the following criteria:

- ❖ Accuracy: The final model specification is to obtain at least 95% of detection accuracy.
- ❖ False-Positive Rate: The attrition rate must be low so that the false-positive rate is less than 5%.
- ❖ Ease of Use: The chosen model should be rather easy to implement and explain to the wider audience.
- ❖ Efficiency: It should be efficient enough so that it does not challenge the standard computational parameters of the hardware.

We will demonstrate the output of the final model through graphical reports and write-ups, thus catering for those with a technical background as well as those without. Project acceptance is well marked by approval from the project supervisor.

3.5 Supporting Process Plans

This section includes documentation, configuration management, quality assurance, reviews and training of such plans to make the Supporting Process Plans to be well structured, transparent, accurate, highly documented and properly streamlined throughout the cycle of the project.

3.5.1 Documentation

Across the entire project duration, there shall be well-documented, and a detailed explanation of the work done at every level. It should also assist all the team members to know the progress, outcomes and the end results of the project. Nonetheless, where appropriate, the documentation will use the IEEE format.

3.6 Overall Description

This section provides a general overview of the project, outlining its context, goals, and key features.

3.6.1 Product Perspective

The product is an ML-based system for detecting Distributed Denial of Service (DDoS) attacks. It integrates with network monitoring systems and aims to improve the real-time detection of malicious network traffic. The system will process incoming network data, apply ML models to identify patterns associated with DDoS attacks, and respond by blocking or mitigating the identified malicious traffic.

- **System Integration:** The system can be deployed alongside firewalls and Intrusion Detection Systems (IDS) as an added layer of security.
- **Technology Stack:** The product will be built using Python, ML libraries such as scikit-learn and TensorFlow, and network monitoring tools.

3.6.2 Product Functions

The core functions of the DDoS detection system are:

- **Traffic Monitoring:** Continuously monitor incoming network traffic in real-time.
- **DDoS Detection:** Apply ML models to analyze traffic patterns and detect potential DDoS attacks.
- **Alerting:** Send alerts to administrators when an attack is detected, including details on the traffic and preventive measures taken.
- **Reporting:** Generate reports summarizing detected attacks, actions taken, and system performance.

3.7 Specific Requirements

This section details the specific functional and non-functional requirements that the system must meet to ensure proper operation.

3.7.1 User Interfaces

1. **Home page:** general information about the website and consists of all the menus.
2. **Log in:** to use your credentials to enter the account.
3. **Sign up:** create an account.
4. **About us:** who are we? What are we doing? And why are we doing this?
5. **The Solutions menu:** consists of 3 pages. Advanced DDoS Detection. Second, results and Findings. Finally, Contributors.
6. **Solution dashboard:**
   a) **Advanced DDoS Detection:** upload a csv file and choose the wanted ML algorithm to receive the performance metrics.
   b) **Results and Findings:** share key results and findings in our graduation project that were vital to implement our machine learning algorithms.
   c) **Contributors:** extending our deepest gratitude to our supervisors and the mentors who helped us in this project
7. **Contact us:** sending a message to our team

3.7.2 Functional Requirements

- **Attack Detection:** The system must accurately identify the various categories of DDoS attack (volumetric, protocol, application layer) with a rate of not less than 95%.
- **Reporting and Logs:** The system should also keep detailed logs and reports of all attacks it has found and traffic patterns, source IPs among others actions the system took.
- **Alerting System:** Using the program modules, administrators should be notified the moment an attack is launched, and some of the details that should be relayed may include the nature of attack, and the source of the attack.

3.7.3 Non-Functional Requirements

- **Performance:** The processing of up to 10 Gbps of network traffic should not hamper the functionality of the system.
- **Scalability:** The system will have to be able to accommodate increasing amounts of network traffic and should be friendly to large networks.
- **Accuracy:** False positive should be kept below 5% in order not to disrupt the genuine traffic frequently.
- **Security:** The system should meet the current network security standards with the intent of preventing vulnerability that would compromise traffic data.
- **Usability:** The network administrator, the actual user of the tool, must be able to immediately deliver attack visuals and controls in an easily usable UI.
- **Reliability:** The system should be dependable, with structural availability not less than 99.9% in order to monitor and protect the system in case of a failure.

Chapter 4: Methodology & Design

4.1 Methodology

In this section the methodology framework is presented along with a description of the proposed machine. and assessment methods, as well as a model evaluation procedure.

4.1.1 Methodology Framework

Several proposed ML techniques are used in the ML Detect project to detect DDoS attacks successfully. The algorithms considered are Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Bagging, Light Gradient Boosting Machine (LightGBM), Category Boosting (Catboost), and AdaBoost. A quantitative comparison of each technique is made and based on the results of the technique with respect to accuracy, precision, recall, F1-Score and reduce false positive. The objective is to find the model with the best detection performance, the highest level of accuracy and reduce the false positive alarms during the detection duration, which will form the basis for the functional incorporation of the developed ML Detect system to real-time DDoS attack detection.

For this project, the dataset used is network traffic which is collected from public datasets which is CICDDoS2019 and other emulated environments to achieve all-weather Benign and Attack traffic. Most of the time, data preprocessing is done very orderly in a way to enhance the quality of the data fed into the models. It entails data cleaning which involves deletion of noises and irrelevant attributes, numerical attributes scaling and normalization, transformation if necessary. The obtained data is then randomly divided into training and testing datasets in a 4:1 ratio as a simplified application of the k-fold cross-validation technique to avoid the problem of overtraining.

The initial step in building the model is completed on the prepared data and different Algorithms are tested for their efficiency. Where a model is not able to perform as expected, hyper parametrization is carried out. Grid search or random search methods are used to fine-tune parameters including, learning rate or tree depth or the number of estimators for better detection outcomes. Moreover, methods like Recursive Feature Elimination (RFE) are used occasionally when required for improved model performance and distance of legibility.

After such an impressive campaign has been developed and optimized, it is tested for its efficiency in a validation process. The model is then used in real time traffic analysis in the ML Detect architecture as mentioned above. New data is used to make predictions, and Dynamic Analysis is constantly in feedback and adaptive mode checking on its performance against new interfering signals and or new attack patterns etc. The results of different models are compared with respect to these pre-defined objective criteria to determine which model should be incorporated henceforth. The last one is predetermined by its capacity to retain a high rate of detections while still sliding low through the detection of false positives, thus making a reliable DDoS detection and proper solutions to it.

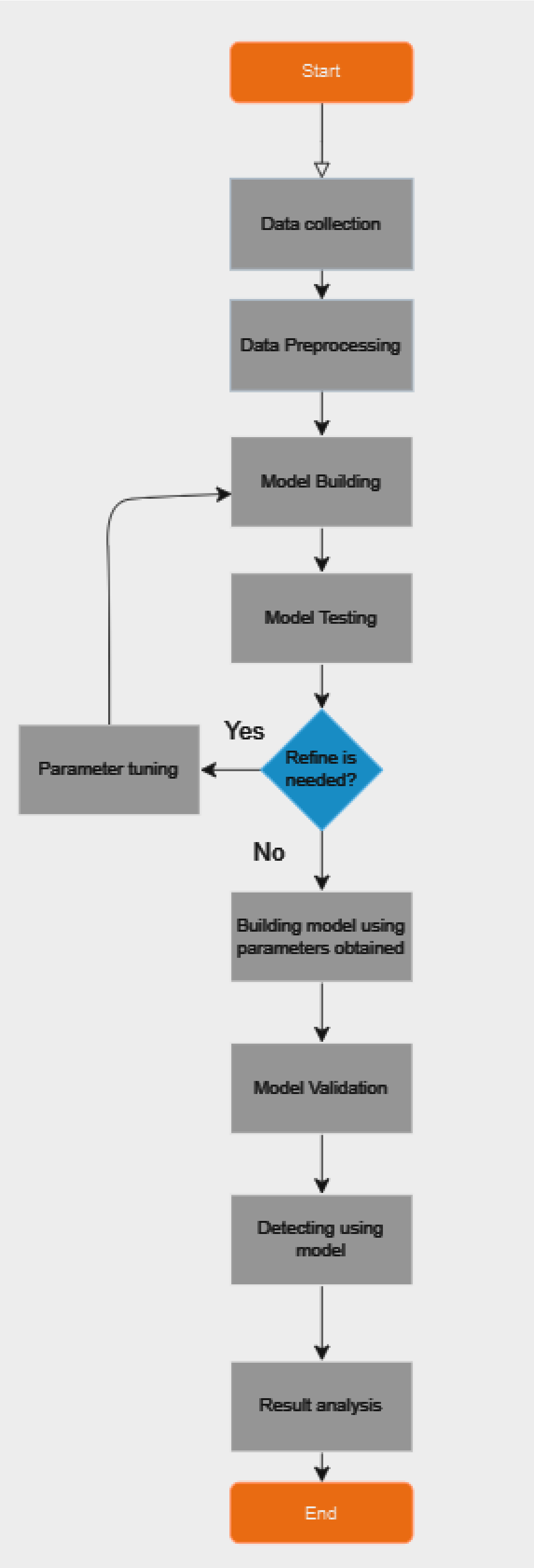


Figure 3: Phases of developing a ML model

4.1.1.1 *Data Collection phase*

The actual data collection process takes place in this stage through the use of publicly available CICDDoS2019 dataset. Attack scenarios based on real-world DDoS scenarios appear in this dataset which includes controlled simulated attack types and labeled benign flow data records. The pre-labeled dataset comprises two distinct groups: these include benign traffic and DDoS attack traffic that provides sufficient diverse data for successful model training and evaluation processes.

4.1.1.2 *Feature Extraction Phase*

There exists a phase where critical attributes are extracted from the CICDDoS2019 dataset for building a robust feature set. A set of important behavioral features such as packet inter-arrival time and flow duration in addition to protocol types and source/destination IP behaviors will serve as indicators for distinguishing normal from attack traffic. Machine learning models require these features to identify correctly the patterns that relate to DDoS attacks.

4.1.1.2.1 Data pre-processing phase

Preprocessing of collected data occurs prior to training in order to enhance model effectiveness. The CICDDoS2019 dataset contains extensive network traffic data and accepts packets with possible duplications and unwanted signals along with unnecessary fields. The data preprocessing stage includes data cleaning followed by the extraction of relevant fields including packet size as well as duration and flow statistics followed by scaling and normalization techniques to standardize the input data. The reduction of noise and the enhancement of both precision and speed in the learning algorithm require these necessary steps [15].

4.1.1.2.2 Noise Removal Phase

To build reliable models the process of noise removal incorporates the use of valid and clean data. The model receives cleaned data from the CICDDoS2019 dataset after removing double packets and incomplete traffic flows together with unneeded metadata. The dataset becomes more specialized through this filtering process which optimizes its effectiveness for DDoS detection pattern recognition.

4.1.1.2.3 Normalization

The CICDDoS2019 dataset requires normalization to balance the measurement scales of packet size and flow duration information. The Python packages NumPy and Pandas enable a process to validate that each feature provides equivalent impact during model development. Bigger feature values will affect prediction results negatively if normalization is not performed. Model accuracy and both training stability and speed improve through proper normalization of features.

4.1.1.2.4 Removing irrelevant features phase

The CICDDoS2019 dataset contains fields which provide minimal benefit during DDoS detection operations. The CICDDoS2019 dataset contains three types of information that slow down attack identification such as repeated headers and zero-filled packets along with fields which provide no attack-related value. Such irrelevant data points are excluded from the feature set so that it becomes more efficient and concentrates on attributes known to enhance detection accuracy and decreases computational requirements.

4.1.1.2.5 Feature Reduction Phase

The most important attributes guide the selection of features when performing dataset simplification. The CICDDoS2019 dataset receives Principal Component Analysis (PCA) treatment to decrease its dimensions while maintaining valuable data through our research. The training process achieves faster speeds and greater efficiency with better accuracy as a result of this technique [21].

4.1.1.2.6 Feature Grouping and Correlation

The variables are clustered so that feature grouping and correlation analysis expose dependencies between network traffic characteristics. This approach is useful in enhancing the predictive nature of the model used. For instance, if source IP is taken in conjunction with the destination IP as feature vectors, then some patterns are notable of a DDoS attack. Moreover, it can be useful in attributing the features that complement each other, for example, the size of packet and the frequency of the transmission.

There are three main purposes for using feature grouping in ML Detect:

1. **Detecting Patterns:** It assists in identifying those rush traffics that might be an indication of an attack.
2. **Reducing Redundancy:** These features are related to similar items, so that there are fewer redundant fields and processes.
3. **Predicting Anomalies:** Proactively helps to forecast potential extraordinary behavior, thus improving the performance of the threat identification process.

4.1.1.2.7 Feature Importance Analysis

Feature Importance Analysis is a method for finding out about the importance of different features in identifying DoS attacks, namely DDoS attacks. Here it can be compared to text classification, where each feature vector is represented by words and TF-IDF weights.

- **Feature Frequency:** Describes the relative frequency of any feature of practical interest, such as the size of the packets or type of protocol. Highly frequently occurring features, which do not result in additional value, on the other hand, receive a lower rank.

- **Inverse Feature Rarity:** Describes how anomalous a feature is in relative to how it appears in all other samples of data. To be able to find attacks features that are less likely to occur are considered more important.

This serves to enhance the means of determining which features have a large impact on the accuracy of DDoS detection and lower false positive while at the same time minimizing impacts of the least informative features.

4.1.2 Definition of Proposed Classification Techniques

On completing the data pre-processing, the collected data will be passed on to different classifiers of ML. Some of the models used are Naïve Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Bagging, Category Boosting (CatBoost), AdaBoost, Artificial Neural Network (ANN), k-Nearest Neighbors (KNN) and Stochastic Gradient Descent (SGD).

This section will explain only in general about each classifier and why using this classifier for DDoS attacks detection. As a result, the aim here is to identify the suitability of these models for real-time attack detection and choose the one that yields maximum accuracy and restricts time.

4.1.2.1 Random Forests (RF)

Random Forest, being a type of ensemble learning, develops several decision trees and then combined results to minimize variance. This method works through predictions for each tree and through a majority vote in classification problems. Overfitting can be eliminated because uncorrelated trees average the result, which would minimize predictive errors.

Advantages of Using Random Forest:

1. **Reduced Overfitting:** To evade the issue of overfitting, Random Forests carry out the average of many trees, so that overall generalization is enhanced.
2. **Flexibility:** Proves to be very useful in both classification and regression analysis problems and can deal with the missing values.
3. **Feature Importance:** They are convenient in terms of feature selection where we can easily identify those with the most influence during the selection of the best model.

Challenges of Using Random Forest:

1. **Resource Intensive:** Random Forests demanding large computational resources and time and space memory in case of large datasets.
2. **Complexity:** To paraph, the model by its very design is less interpretable than a single decision tree.
3. **Time-Consuming:** Training many trees takes a long time, especially during real-time training, and this aspect affects its performance.

This approach can be used to the advantage of ML Detect since it has high accuracy and robustness in the identification of DDoS attacks while it still needs resource optimization.[31]

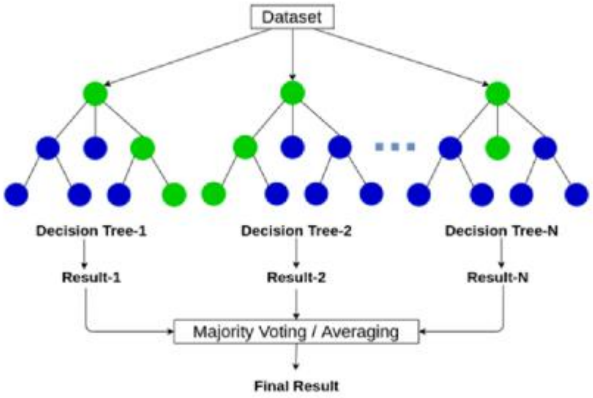


Figure 4: The Random Forest Algorithm [36]

4.1.2.2 Naïve Bayes (NB)

Naïve Bayes is an ML Classifier, a type of supervised classification that is used in ML Detect for identification of DDoS attacks. It is built on Bayes' formula, which estimates the chance of a particular data point to belong to a certain class. When it comes to classifying text data Naïve Bayes has a major disadvantage since it assumes all the features are independent; the presence of a feature in a particular class does not influence the presence of another feature in the same class.

This algorithm defines the likelihood of a data instance belonging to DDoS attack or Benign traffic given some event. The most probable class is chosen as prediction class Hence, The prototype of the class with the highest probability is identified as the Pred class. This method is very easy to implement and works well in problems where classes are very well separated.[15]

4.1.2.3 Support Vector Machine (SVM)

SVM is another type of supervised learning in ML Detect that aims at classifying network traffic as, either that of a legitimate or an attacker. SVM works by loading data into an optimal hyperplane that will divide the data into two or more classes. By the points closest to the hyperplane, known as Support Vectors, the distance of the hyperplane is defined.

Yet, this is not a limitation for SVM since kernel functions allow us to solve also nonlinear problems. It works better in a higher number of dimensions, where class margins are more easily distinguishable. But for making it effective for very large data sets, it has certain limitations. The equation for the hyperplane in SVM is given by:

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = \mathbf{0}$$

Equation 1

where w is the weight vector x is the input vector and *b* is the bias. In this it seeks to optimize the distance between the data points of the different classes and the hyperplane and thus have maximum margin.[32][33]

4.1.2.4 Logistic Regression (LR)

In fact, the by-name Logistic Regression (LR) pertains to the classification technique employed in the ML Detect for differentiating between the benign and malicious traffic. It is especially useful where the problem is of the type dealing with a binary set of outcomes. The algorithm employs a linear equation as the measure of correlation.

Equation 2

$$h\Theta(x) = \beta_0 + \beta_1 X$$

X, and transforms it into a probability using the sigmoid function:

$$h\Theta(x) = 1/(1 + e^{-(\beta_0 + \beta_1 X)})$$

Equation 3

This transformation produces the S-shape curve and is used to map value in between 0 and 1 only. If the probability value is less than 0.50, then the data is labeled as ’0 – benign,’ otherwise, if it is greater than or equal to 0.50, the data is labeled as ‘1–malicious.’ It is Logistic Regression which is easy and quite efficient for network security for binary classification.[34]

4.1.2.5 AdaBoost

AdaBoost, short for Adaptive Boosting, is an ensemble learning method that is used in ML Detect to lever up the performance of weak classifiers. It is based on the principle of combining several weak classifiers to make a strong classifier and more accurate. The first step in the process is to choose at random a subset of the training data and then train the first model. In later round of the learning process Ada gets the samples with high weights that were misclassified and tries to give more attention to these observations.

Further, models with high accuracy are assigned higher weights so that they make a major input to the final prediction. It goes on until the model converges or till the predefined number of iterations is completed. AdaBoost achieves better or equivalent classification accuracy when used to train models, especially those trained on imbalanced or high complexity data.

4.1.2.6 XGBoost

XGBoost is one of the most efficient ML algorithms which are used in our ML Detect for effective DDoS attack detection. It expands from the gradient-boosted decision trees and is trained for the high scalability and high performance.

XGBoost is a supervised learning algorithm which is trained with labeled network traffic data to enable the algorithm to make its decisions. It uses decision trees which are stop-and-go questions that filter traffic as either ‘good’ or ‘ill-intentioned’. The boosting pushes multiple trees in XGBoost system, increases the prediction quality and reduces errors making it suitable in large and complicated data in network security.

4.1.2.7 CatBoost

CatBoost is a gradient boosting algorithm which is integrated into ML Detect to work with categorical variables. This method is based on the creation of MZD, which is an ensemble of T weak learners which are shallow decision trees, and the trees are added gradually in a way to minimize the loss function.

But CatBoost differs from other boosting algorithms by applying such methods that minimize overfitting, as a result, the number of parameters to be tuned can be greatly reduced. The algorithm guarantees the model is not overfitted, making it ideal for use when handling DDoS where such a problem would be counterproductive.

4.1.2.8 Stochastic Gradient Descent (SGD)

ML Detect exploits Stochastic Gradient Descent (SGD) as an optimization algorithm for model training through which error rates during learning reach minimum values. Instead of traditional gradient descent's practice of assessing gradients throughout the whole dataset the model uses one data point at a time during parameter updates. Because of its light incremental nature, it offers optimal performance in real-time DDoS detection systems operating on a large scale.

ML Detect depends on SGD for boosting model training performance and reaching convergence especially in linear models and neural networks. Another advantage of SGD is its random feature which helps the system avoid overfitting because it enables the model to escape local minima thus providing better adaptation to dynamic attack patterns found in network traffic.

4.1.2.9 k-Nearest Neighbors (KNN)

The k-Nearest Neighbors (KNN) algorithm functions as an effective and simple classification tool within ML Detect that uses proximity measurements in feature space to detect DDoS attacks. KNN-based classification places new traffic points against their closest "k" data neighbors to determine their label from the most prevalent class type.

The ML Detect solution relies on KNN processing to evaluate incoming traffic against its historical records for normal behavior identification. ML Detect operates well in dynamic network settings because of its non-parametric structure together with its simple design which speeds up deployment without complex model training needs.

4.1.2.10 Artificial Neural Networks (ANN)

The detection system ML Detect makes use of Artificial Neural Networks (ANNs) as a brain-inspired model which analyzes sophisticated network traffic patterns.

The ML Detect system utilizes ANNs to evaluate multi-dimensional traffic information which enables it to differentiate between genuine and harmful network connections. The nonlinear learning ability together with training adaptation of these models makes them excel at detecting sensitive signs of DDoS attacks which leads to more accurate detection and fewer false positives.

4.1.3 Model Evaluation

The specific measures that will be used for assessing the performance of classifiers in the ML Detect include accuracy, precision, recall or sensitivity, and F1-Score. The primary categories of metrics used by these models are calculated from true positive (TP), true negative (TN), false positive (FP), and false negative (FN) measures to demonstrate the differentiation between benign and malicious traffic. Such evaluation criteria make sure that the models generate the correct result for identifying the DDoS attacks.

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

Equation 4

$$Precision = \frac{TP}{TP+FP}$$

Equation 5

$$Recall = \frac{TP}{TP+FN}$$

Equation 6

$$F1\ score = 2 * \frac{precision * recall}{precision+recall}$$

Equation 7

4.2 System Design

The System Design describes the needed architecture and components to assess benchmark ML algorithms for detecting and mitigating against DDoS attacks. As the work primarily revolves around model evaluation rather than system deployment, the design is considerably more flexible, modular system that relies on data analysis.

4.2.1 System Architecture

The structure of the learning system is developed in a modular fashion where various types of ML models can be jointly coded and tested for performance comparison. It consists of the following components:

a. Data Collection:
   - i. Function: collect and preprocess data from public datasets.
   - ii. Features:
     1. Handle raw data (especially in CSV format).
     2. Automating tasks such as normalization and data cleaning and feature extraction.
   - iii. Input: Raw CSV dataset file
   - iv. Output: Dataset that is ready for model training.

b. Model Evaluation Engine:

i. Function: Fit several supervised ML algorithms onto the preprocessed data Split the data into the training, testing and validation sets.

ii. Features:

1. Here, it supports algorithms such as Random Forest, XGBoost, CNNs, LSTM, and others as well.
2. Compared model performance using factors such as accuracy, precision, recall, F1-score and, false-positive rate.

iii. Input: Preprocessed dataset.

iv. Output: The model performance metrics and the computed evaluation results.

c. Visualizing and reporting module:

i. Function: Make visual representations of the outcome of model assessments and underscore them with detailed reports.

ii. Features:

1. Just reports results as graphics (such as confusion matrices, ROC curves, etc).
2. Produces summary documents needed for the best model for risk measurement.

iii. Input: Results of the performance tests from the Model Evaluation Engine.

iv. Output: Visual aids and computational presentations and documented findings.

4.2.2 Workflow

i) Data Processing:

(1) Read files and perform data cleaning techniques such as handling missing values in the data set, scaling and encoding the nominal variable.

(2) To filter out key characteristics of the connection, including packet size, connection rate, types of traffic.

ii) Model Testing:

(1) Divide the data into three chunks – used for training, for validation and for testing (these ratios may be used, 70-20-10).

(2) Train models on set and separately use the set to validate.

(3) Check the performance of models of given types on the testing set having standardized correspondence to the goal the model was created for.

iii) Comparison and Analysis:

(1) Ability to plot model results alongside Douglas comparison based on performance.

(2) To compare the models, we can plot the results in the following way:

iv) Documentation:

(1) Document observations for each of the selected models on factors that often make a particular model emerge best.

(2) This proposal aims at identifying potential biases and methods of fighting to develop an action plan to implement when the biases are experienced in a project that involves identifying key Stakeholders and applying project management methods to come up with ways of making project deliverables available.

(3) The paper will conclude with a report on the findings that will be presented to the relevant project stakeholders.

4.2.3 Hardware and Software Requirements [23][30]

Hardware Requirements:

- Personal laptops or desktops with:
- Minimum requirement of 8GB RAM and i5 Processor.
- Storage: 256 GB SSD.
- Optional: Permission to utilize cloud computing such as Google Collab, AWS or Azure for building complicated algorithms.

Software Requirements:

- Programming Language: Python.
- Libraries/Frameworks:
  - ♦ Data Processing: Pandas, NumPy.
  - ♦ ML: scikit-learn, TensorFlow, Keras or scikit-learn, PyTorch, and Keras.
  - ♦ Visualization: Matplotlib, Seaborn.
- Development Environment: Jupyter Notebook or PyCharm.
- Data Sources: Public datasets

4.2.4 Evaluation metrics

The evaluation metrics will help us better understand how the algorithm technique is compared to others, hence we will be using the following metrics:

- ❖ **Accuracy:** Total accuracy measure, where the closer to 100% is better.
- ❖ **Precision:** Ability to distinguish between correct and total cases of DDoS attacks.
- ❖ **Recall:** The possibility of identifying all the real DDoS attacks.

- ❖ **False Positive Rate:** It’s the percentage of the user traffic that has been classified as self-SPIT when it is genuine traffic.
- ❖ **F1-Score:** Achieves an equilibrium between precision and recall.

4.2.5 Modular Design Benefits

The system can have many benefits; however, our system has the best and most vital benefits which are as follows:

- **Flexibility:** It is not necessary to redesign the system when constructing new algorithms since the process is easy to accomplish.
- **Reusability:** Preprocessing and evaluation components can be used in other projects in future or can be enhanced for further usage.
- **Scalability:** It remains scalable through support of integration with other larger datasets or more complex models if the requisite resource capacity is available.

4.3 User Interface Design

An overview of user interfaces, including their design guidelines, screen objects and actions, and screen visuals, is given in this section.

4.3.1 Interface Design Rules

Interface Design Rules:

Eight Golden Rules of Interface Design are a set of principles formulated by one of the fathers of the field of Human-Computer Interaction (HCI) – Ben Schneiderman. The rules mentioned are aimed at helping in developing good graphical front ends with special emphasis on usability, ease of use and the satisfaction of the users [35]. Here's a breakdown of each rule:

Strive for Consistency:

Explanation: Speaking of consistency, this is when you avoid using different terminology and layouts, different color schemes, and interaction styles in your application. This assists users to build familiarity and keep the interface rather predictable and easy to use.

Example: Such aspects as color and shape of buttons, location and structure of menus that are to be positioned on any page of the website.

Facilitate the Frequent Users so They Can Benefit from Shortcuts:

Explanation: For expert users, offer power mode (or shortcuts such as hotkeys or gestures or command line options) that would make interactions faster. This rule makes things easier for power users while not leaving the novices in the cold.

Example: Ctrl + C for copy and Ctrl + V for paste in the text editors.

Offering Informative Feedback:

Explanation: The system should provide prompt and informative responses to each move of the user. It was suggested that the feedback should be proportional to the action taken.

Example: When uploading files, using a spinner or a progress bar, or when a form is submitted, using a message of success.

Design dialogues with the intention of attaining closure:

Explanation: When there are several activities involved in a process, they should be sequenced well and should be easily divided into the first part, middle part and the last part. Use confirmation or summary to give users a feeling of completeness after the completion of a task.

Example: When a user is making an order online, show him the order summary of the entire transaction.

Offer Simple Error Handling:

Explanation: It is better to avoid error conditions at all costs and if this cannot be accomplished then the error state must always be recoverable. Provide comprehensible and constructive error messages and do not use terms that are unfamiliar to users.

Example: Overlaying mandatory fields in red when a user left them empty with a message like, “This field is mandatory” rather than a general error number.

Allow Simple Undoing of Operations:

Explanation: Any action that a user performs should be reversible so that the user is able to go back to the previous state if he or she felt that the action was wrong or reckless, or to redo an action in case the user felt that the action was inconsequential.

Example: A “Delete” button on an email, which gives an option to undo the deletion, or a “Back” button in forms which are multiple steps.

The third element for fostering is the promotion of Internal Locus of Control:

Explanation: It is preferable that users should have the impression of commanding the interface rather than the other way round. Design interfaces that have reactions to the actions being made by the users and do not behave in a peculiar way.

Example: Let the users decide when to save a document rather than making the application to save a document without informing the users.

Reduce Short-Term Memory Load:

Explanation: Reduce the amount of information that must be stored in the user's mind to interact with your design. However, make sure that information is well displayed on the interface.

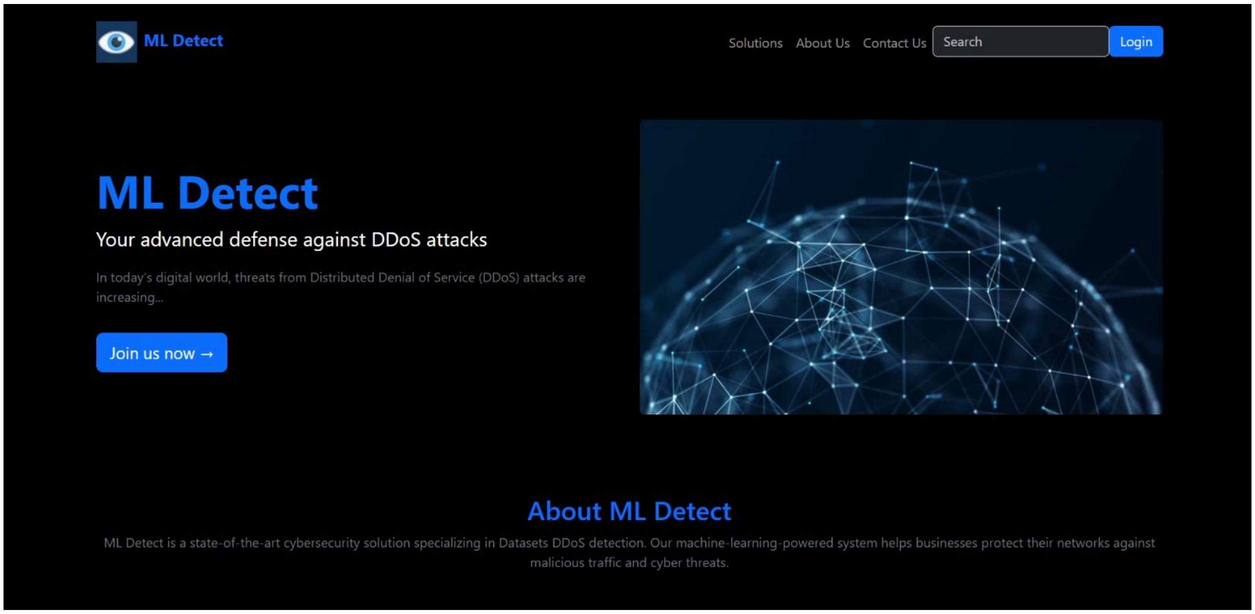
Example: Auto-complete fields which contain similar data as entered before or displaying tooltips or hints for the fields.

4.3.2 Overview of User Interface

The ML Detect homepage provides a user-friendly navigation menu to key sections like "Solutions," "About Us," "Dashboard," and "Contact Us," alongside detailed insights into DDoS threats and ML Detect AI-driven solutions. The login interface features a welcoming security message with an email/password panel, a prominent blue "Login" button linked to sign-up. The sign-up page urges users to safeguard their networks with a clear header and essential input fields, including mandatory privacy consent and an optional subscription checkbox. The DDoS Detection Suite page showcases three cards "Advanced DDoS Detection," "Result & Finding," and "Contributors." with concise descriptions and buttons to enhance user engagement. This cohesive design, marked by a dark background and blue accents, effectively communicates ML Detect commitment to professional cybersecurity services.

4.3.2.1 Homepage Interface

ML Detect offers users a navigable top bar on its homepage which directs traffic toward Solutions, About Us and Contact Us web pages. The left side on ML Detect brings attention to rising DDoS attacks in the Problem segment while introducing users to the Our Solution where machine learning and AI capabilities fight off these dangers. ML Detect invites users through join us button to focus on improving digital security to ensure online safety.



ML Detect

Your advanced defense against DDoS attacks

In today's digital world, threats from Distributed Denial of Service (DDoS) attacks are increasing...

About ML Detect

ML Detect is a state-of-the-art cybersecurity solution specializing in Datasets DDoS detection. Our machine-learning-powered system helps businesses protect their networks against malicious traffic and cyber threats.

Figure 5: Homepage

4.3.2.1.1 About ML Detect

About ML Detect

ML Detect is a state-of-the-art cybersecurity solution specializing in Datasets DDoS detection. Our machine-learning-powered system helps businesses protect their networks against malicious traffic and cyber threats.

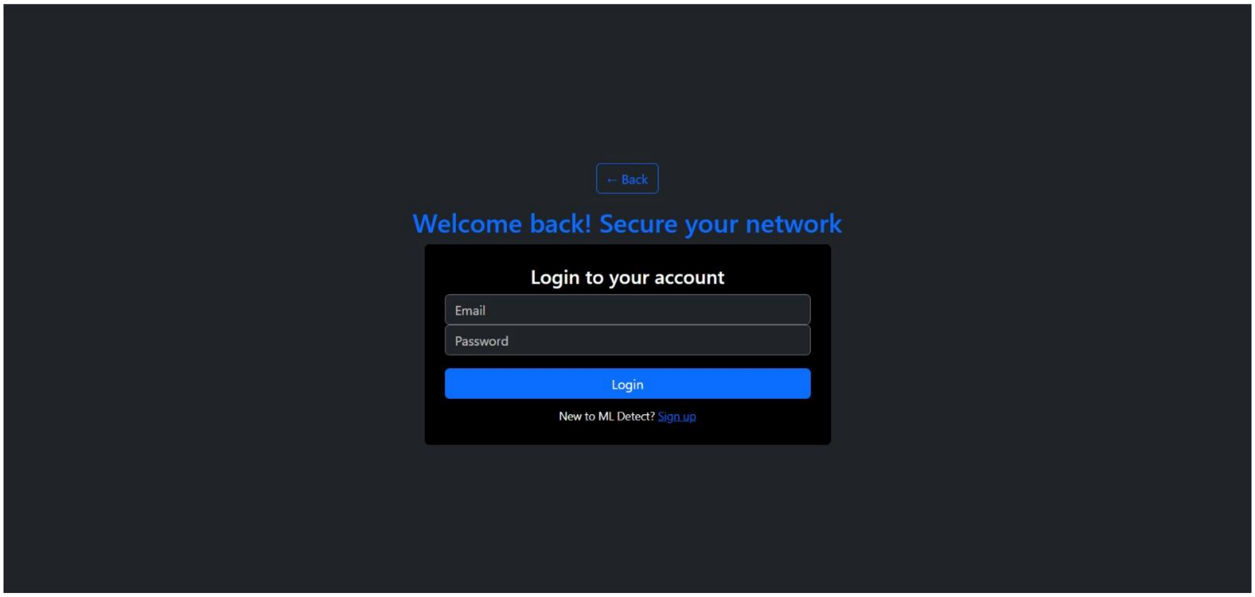
Figure 6:About ML Detect

The image presents the About ML Detect segment of the ML Detect homepage:

Within the About ML Detect segment we present the platform as an advanced cybersecurity technology which uses machine learning algorithms to detect DDoS attacks. The function of our security system remains clear in this part because it emphasizes its ability to shield networks against both malicious traffic and cyber threats.

4.3.2.2 Login Interface

The login page of ML Detect has a network security welcome message followed by a well-designed login panel with space for the user's email and password. In blue color, there is the "Login" button. The platform provides easy sign-up access through the "Sign up" link for new users. A "← Back" button located at the top enables users to go back to previous pages to enhance navigation.



Welcome back! Secure your network

Login to your account

Email

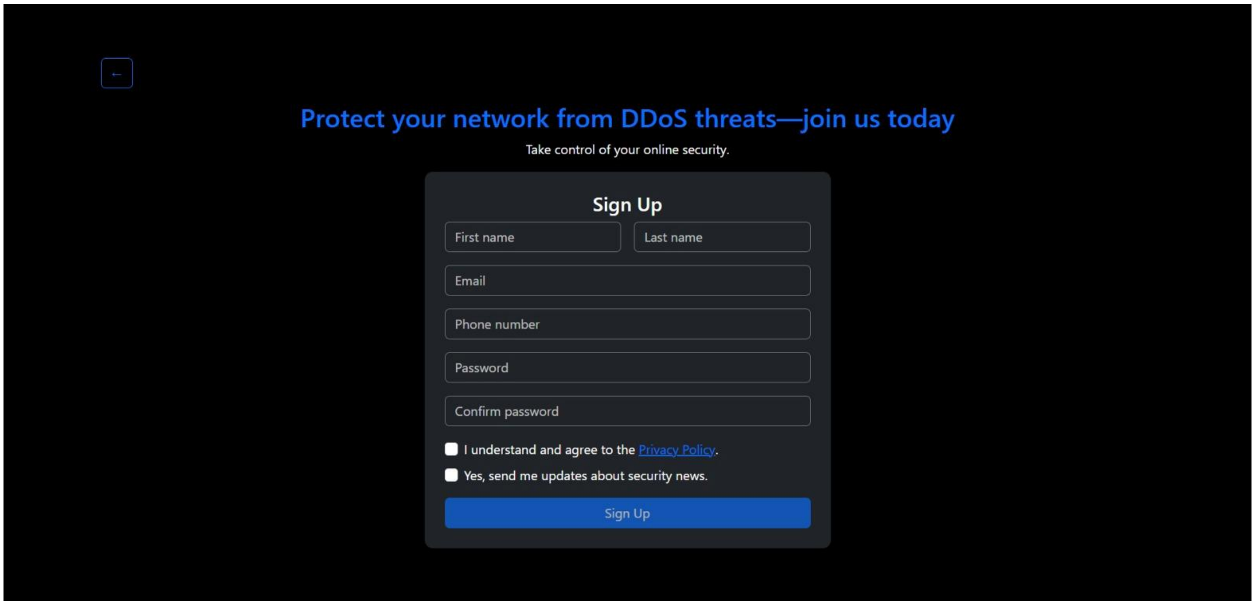
Password

New to ML Detect? Sign up

Figure 7: Login

4.3.2.3 Sign up Interface

ML Detect begins its sign-up process by presenting a header that warns about protecting networks from DDoS incidents and encourages users to defend their online safety. Users can find the ‘Sign Up’ form which asks for first name, last name, email, password and phone number field. Users can view the Privacy Policy by choosing its mandatory checkbox before they can understand how their data will be processed yet they can also join a cybersecurity update subscription via an optional checkbox. Users can easily locate the "Sign Up" blue button which has a prominent position with a back navigation arrow to guide their process.



Protect your network from DDoS threats—join us today

Take control of your online security.

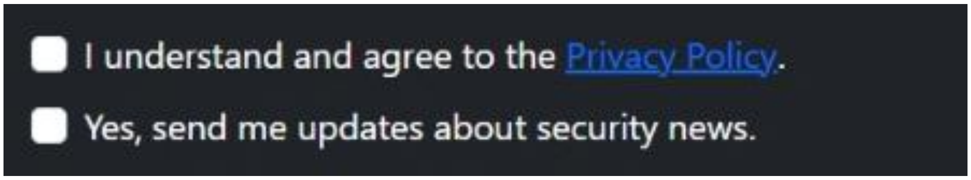
Sign Up

☐ I understand and agree to the Privacy Policy.

☐ Yes, send me updates about security news.

Figure 8: Sign Up

4.3.2.3.1 Privacy and Policy



☐ I understand and agree to the Privacy Policy.

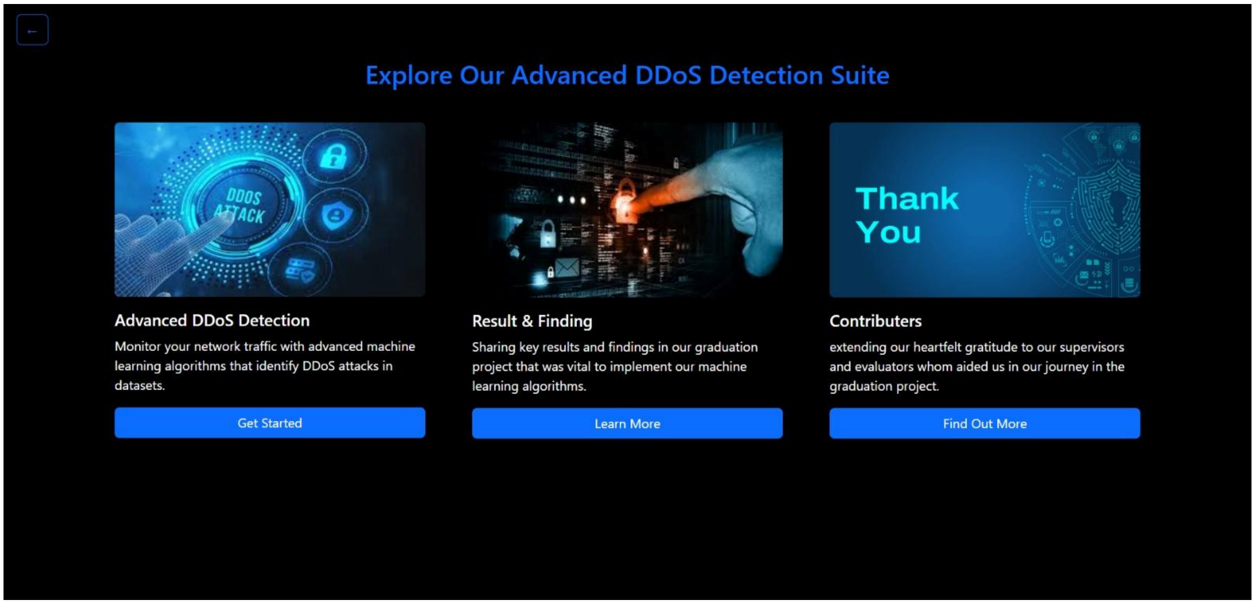
☐ Yes, send me updates about security news.

Figure 9: Privacy and policy

The two checkboxes on the sign-in page serve two important purposes:

- The first checkbox is the standard one that makes the user agree with the company's Privacy Policy regarding their personal data processing.
- The second checkbox allows users to subscribe to the company's newsletter that informs them about new threats, adversary tactics, and customer success stories.

4.3.2.4 Solutions Interface



Explore Our Advanced DDoS Detection Suite

Advanced DDoS Detection

Monitor your network traffic with advanced machine learning algorithms that identify DDoS attacks in datasets.

Result & Finding

Sharing key results and findings in our graduation project that was vital to implement our machine learning algorithms.

Contributers

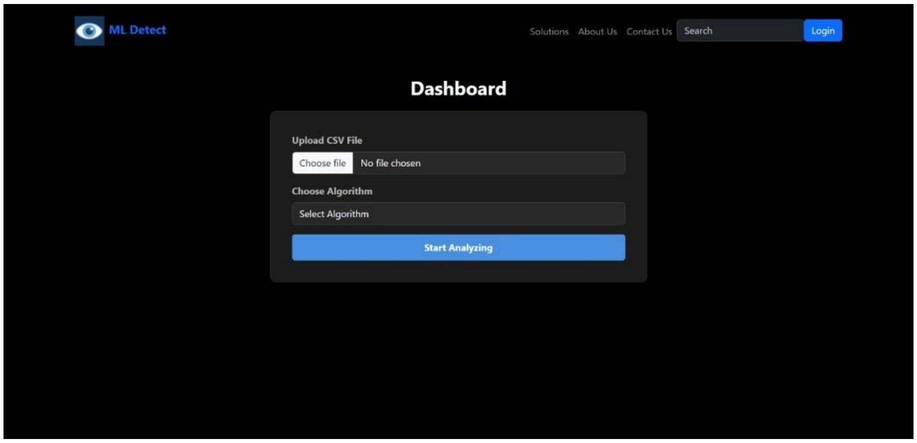
extending our heartfelt gratitude to our supervisors and evaluators whom aided us in our journey in the graduation project.

Figure 10: Solution Interface

Users are invited to discover the advanced DDoS Detection Suite through its title displayed on the Solution page. The page presents three distinct sections which the user can easily navigate.

- **Advanced DDoS Detection:** The DDoS detection functionality depends on advanced real-time network traffic monitoring which uses superior machine learning algorithms together with other features to detect attacks. All users can reach the system using the main "Get Started" button.
    - After clicking the “Get started" button users can use the ML Detect dashboard to perform straightforward examination of DDoS detection information.
        1. Proceeding with file upload involves using the "Choose File" button to let users choose analytical data from their CSV files.
        2. Users can choose their preferred algorithm from a list which includes K-Nearest Neighbors, Artificial Neural Network and Naive Bayes etc.
        3. Among the available actions users must select their file and algorithm before clicking the "Start Analyzing" button to commence analysis.

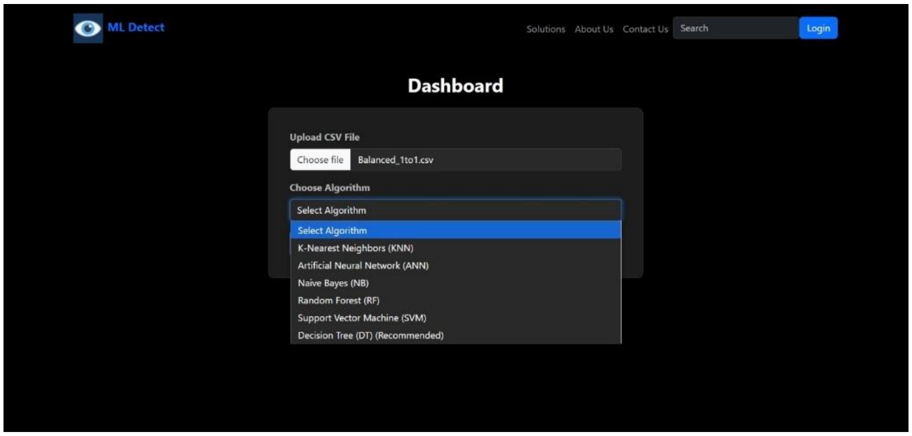
4. The system reveals analytical findings that feature accuracy together with precision and recall rates as well as F1-score and false positive rate to demonstrate the efficiency of selected algorithms.



Upload CSV File: No file chosen

Choose Algorithm: Select Algorithm

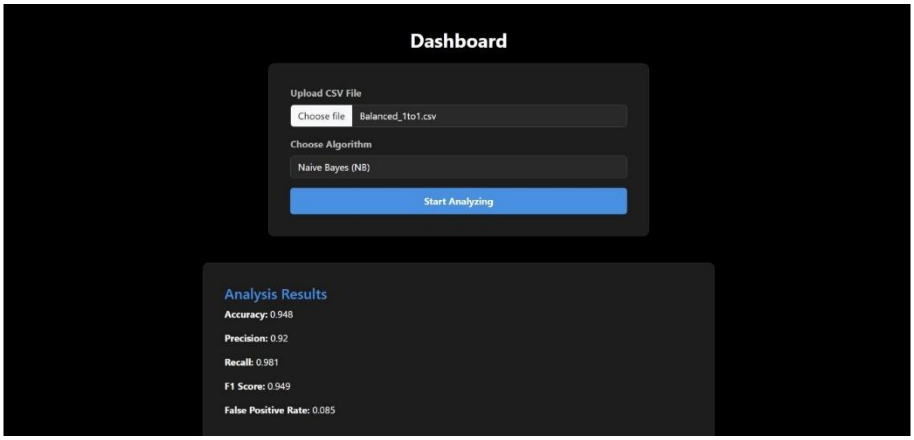
Figure 11: Dashboard Interface (1)



Upload CSV File: Balanced_1to1.csv

Choose Algorithm: Select Algorithm

Figure 12: Dashboard Interface (2)



Dashboard

Upload CSV File: Balanced_1to1.csv

Choose Algorithm: Naive Bayes (NB)

Analysis Results

Accuracy: 0.948

Precision: 0.92

Recall: 0.981

F1 Score: 0.949

False Positive Rate: 0.085

Figure 13: Dashboard Interface (3)

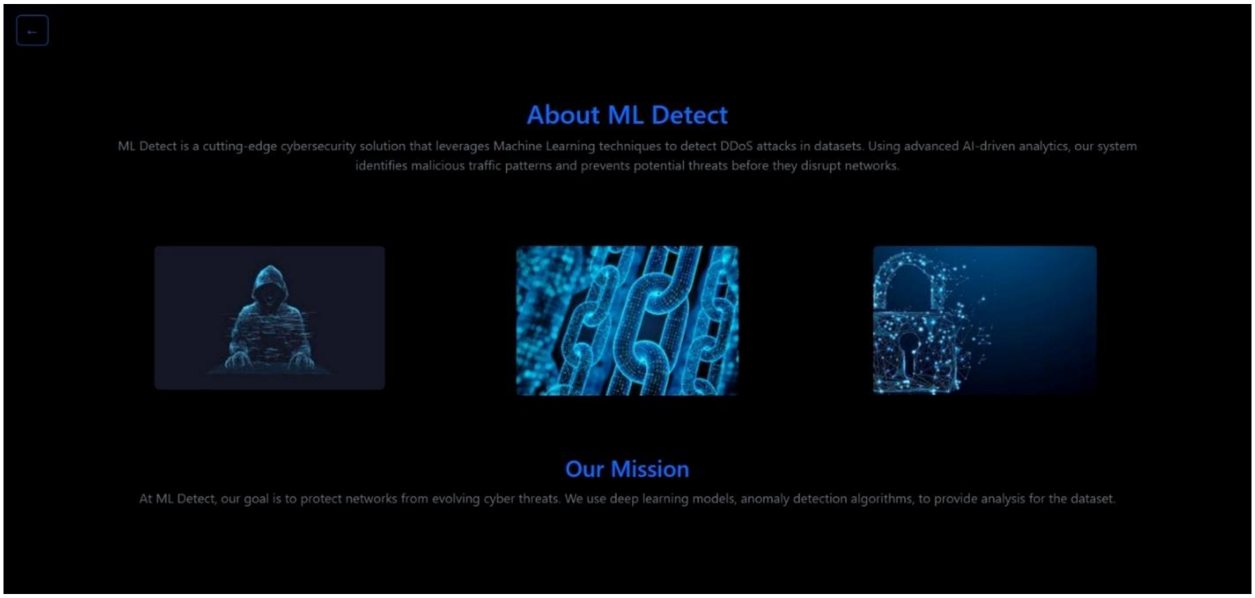
- **Result & Findings:** This section focuses on presenting results as well as findings from a graduation project which developed optimal DDoS detection methods. Further information becomes accessible when users press the “Learn More” button.

- **Contributors:** This section acknowledges supervising staff and evaluators who backed the project and provides additional information about their impact through the "Find Out More" link.

The website design presents a clean professional appearance through a dark framework which combines blue elements to boost cybersecurity aesthetics together with user-friendly interfaces.

4.3.2.5 About Us

ML Detect is described in the "About Us" section as a complete cybersecurity solution that uses machine learning algorithms for DDoS attack identification in datasets. The AI algorithms implemented in ML Detect security patterns of malicious traffic which allows threat interruption before network disruption occurs. The platform serves as an advanced security solution which protects networks from changing cyber threats by using deep learning models and anomaly detection algorithms while presenting a design that displays the critical nature of cybersecurity.



About ML Detect

ML Detect is a cutting-edge cybersecurity solution that leverages Machine Learning techniques to detect DDoS attacks in datasets. Using advanced AI-driven analytics, our system identifies malicious traffic patterns and prevents potential threats before they disrupt networks.

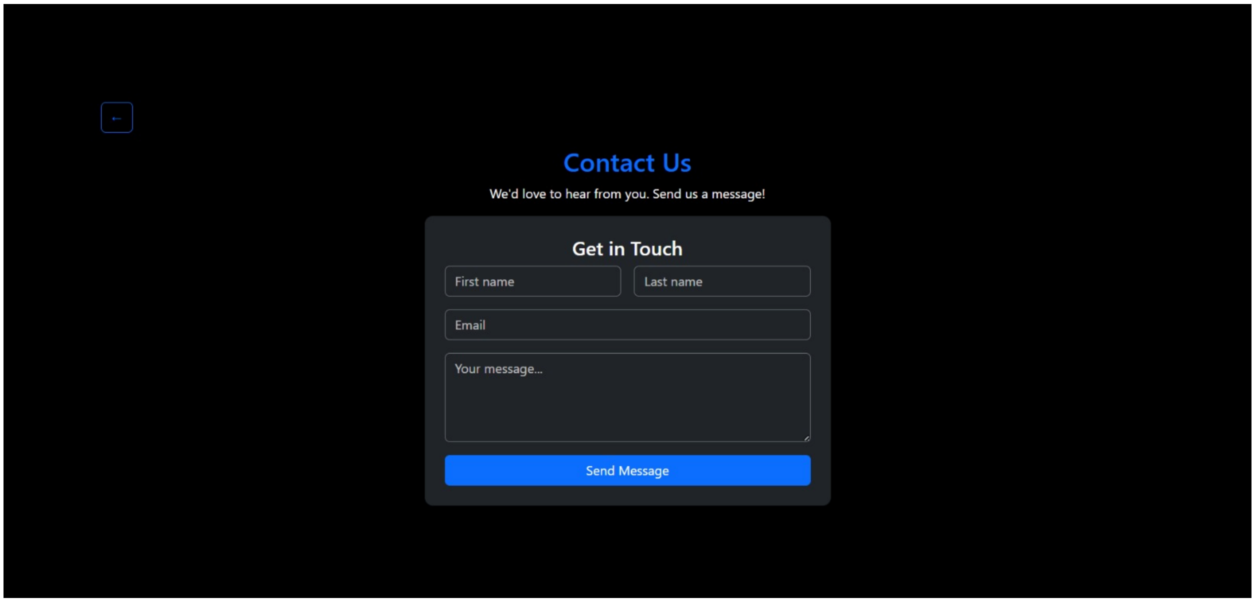
Our Mission

At ML Detect, our goal is to protect networks from evolving cyber threats. We use deep learning models, anomaly detection algorithms, to provide analysis for the dataset.

Figure 14: About Us Interface

4.3.2.6 Contact Us

Through the "Contact Us" section users who wish to contact the website can use the interface and send their messages through its provided form. The contact form provides users with a basic interface that asks for first name, last name, email address and message text for their feedback. A prominent "Send Message" button serves users by providing an easy option to submit their contact form requests. Users maintain a focused user experience through the dark background because it optimizes input field visibility for better interaction with the website.



Contact Us

We'd love to hear from you. Send us a message!

Figure 15: Contact Us Interface

4.3.3 Interface Object and Actions

4.3.3.1 Homepage Interface

NO	Object	Type	Action
1	ML Detect Logo	Image	Navigates to homepage when clicked
2	Navigation Menu	Buttons	Navigates to "Solutions," "About Us," "Contact Us"
3	Login Button	Button	Opens the login page
4	Search Bar	Input	Allow users to search for content
5	Motivation sentence to join us Section	Text Block	Displays information about DDoS threats and Motivation sentence.
6	About the ML Detect Section	Text Block	Explains ML Detect briefly
7	Join Us Button	Button	Navigates to sign-up page

Table 12: Homepage Interface

4.3.3.2 Login Page Interface

NO	Object	Type	Action
1	Back Arrow	Buttons	Navigates back to the previous page
2	Heading Text	Text Block	Contain The Login text
3	Email Input Field	Input	Allow users to enter their email
4	Password Input Field	Input	Allow users to enter their password
5	Login Button	Button	Submits login credentials
6	Sign Up Link	Hyperlink	Submits login credentials

Table 13: Login Page

4.3.3.3 Sign-Up Page Interface

NO	Object	Type	Action
1	Back Arrow	Button	Navigates back to the previous page
2	Heading Text	Text Block	Displays sign-up encouragement
3	First Name Field	Input	Allow users to enter first name
4	Last Name Field	Input	Allow users to enter last name
5	Email Input Field	Input	Allow users to enter their email
6	Phone Number Field	Input	Allow User to enter their phone number
7	Password Input Field	Input	Allows user to set a password
8	Confirm Password	Input	Allows user to confirm password
9	Terms Checkbox	Checkbox	Accept terms and conditions
10	Newsletter Checkbox	Checkbox	Subscribe to newsletters
11	Sign Up Button	Button	Submits registration information

Table 14:Sign Up Page

4.3.3.4 Solutions Page Interface

NO	Object	Type	Action
1	Back Arrow	Button	Navigates back to the previous page
2	Heading Text	Text Block	Displays solutions overview
3	Advanced DDoS Detection Card	Card	Use the ML Detect dashboard to perform straightforward examination of DDoS detection information.
4	Result & Finding Card	Card	Presenting results from a graduation project
5	Contributors Card	Card	Acknowledge supervising staff and evaluators who backed the project and provides additional information
6	Button Get Started (Each Card)	Button	Navigates to a detailed page about the solution

Table 15:Solutions Page

4.3.3.5 Dashboard Interface

NO	Object	Type	Action
1	ML Detect Logo	Image	Navigates to homepage when clicked
2	Search Bar	Input Field	Allow users to search across the platform.
3	Login Button	Button	Opens the login page
4	Navigation Menu	Buttons	Navigates to "Solutions," "About Us," "Contact Us"
5	Heading Text	Text Block	Displays the word "Dashboard"
6	Upload CSV File	Label + File Input	Opens file picker dialog to select a local CSV file.
7	Choose Algorithm	Label + Dropdown	Let the user select a machine learning algorithm.
8	Start Analyzing Button	Button	Use the chosen file and algorithm to start the analysis process then show the results.

Table 16: Dashboard Page

4.3.4 Interface Sequence Diagram

4.3.4.1 Sequence Diagram Login

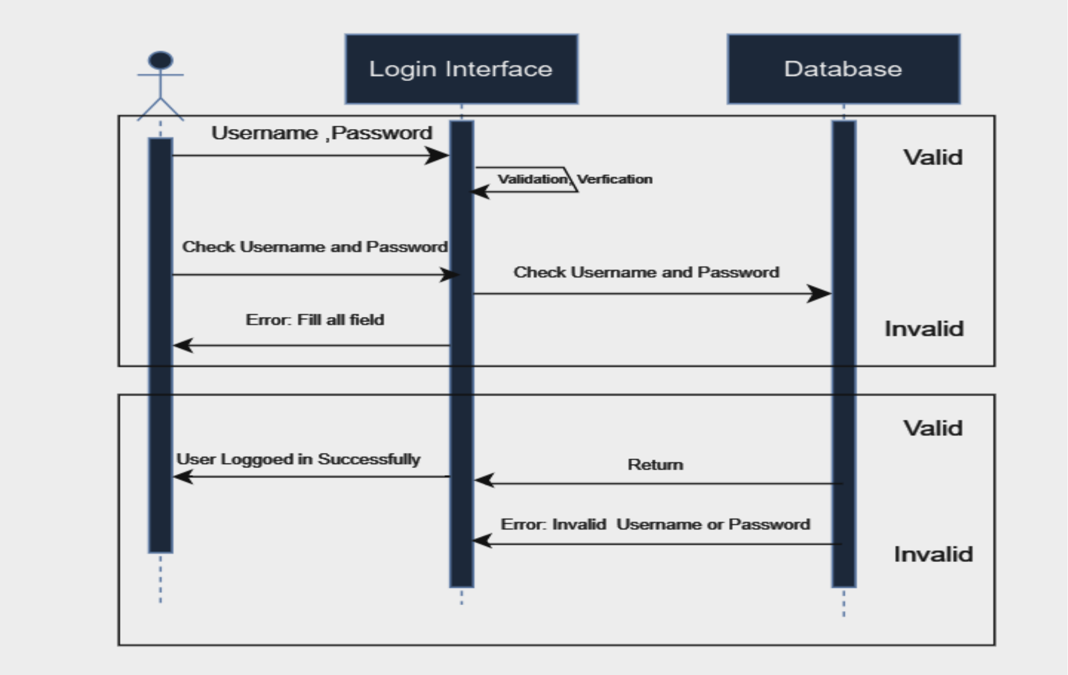

Login Interface
Database
Username ,Password
Validation
Verfication
Valid
Check Username and Password
Check Username and Password
Error: Fill all field
Invalid
Valid
User Loggoed in Successfully
Return
Error: Invalid Username or Password
Invalid

Figure 16: Sequence Diagram Login

Table 16 shows the description, inputs, outputs, and constraints of the "Login component."

Description	The user enters the unique ID and password, and then it would be matched against the database to validate
Input	ID Password
Output	Redirection to the main page
Constraints	ID and password must be valid

Table 17: Login Diagram

4.3.4.2 Sequence Diagram Detection function

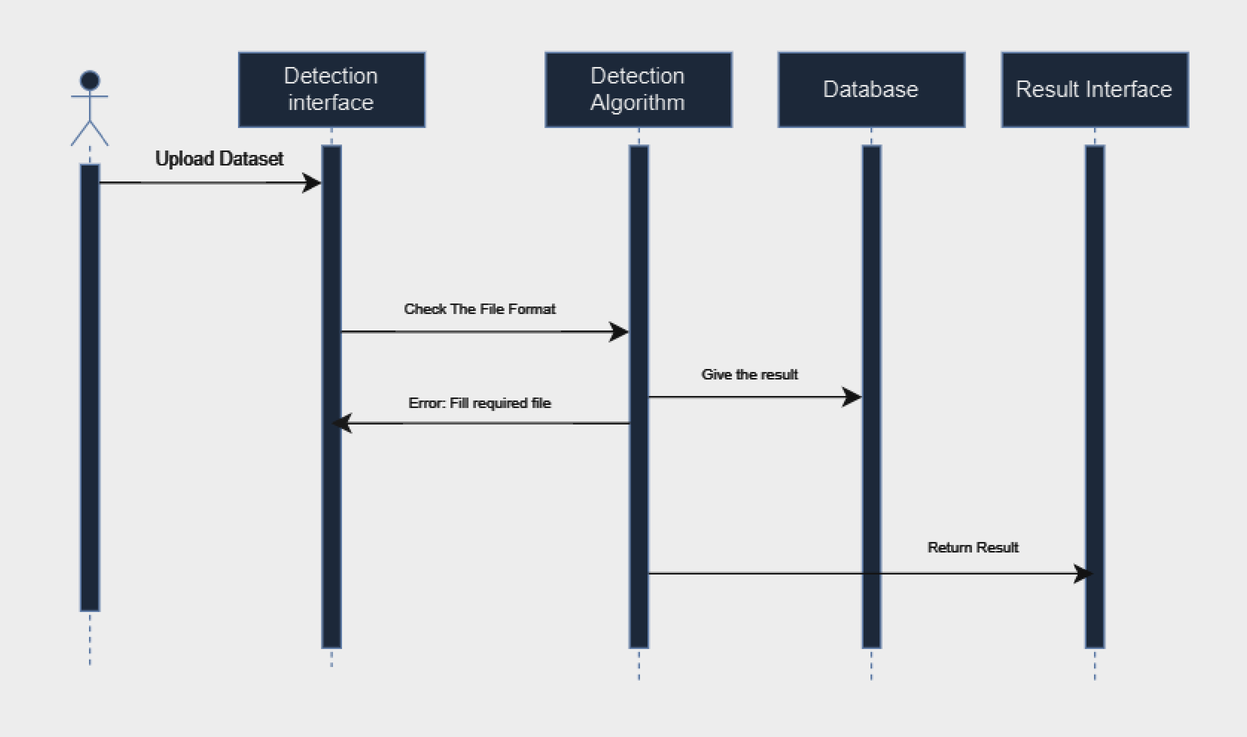


Figure 17: Sequence Diagram Detection function

Table 17 shows the description, inputs, outputs, and constraints of the "Detections component."

Description	The user should start a new detection by uploading dataset and filling the required fields.
Input	Dataset
Output	Display Results interface
Constraints	The dataset should be in a valid format.

Table 18:Detection Diagram

5. Implementation

5.1 Introduction

The chapter details step-by-step how the proposed machine learning-based Distributed Denial of Service (DDoS) detection model should be implemented. The methodology section selects machine learning algorithms to apply them to a specific dataset which was chosen by careful research and was done by the literature review. The implementation involves traditional machine learning processing by acquiring data while performing preprocessing actions before choosing models for optimization and evaluation.

In this section, the process of implementing this project is discussed. The implementation process can be represented in three parts: Data collection, Data Preprocessing, Feature engineering, and Generating Models.

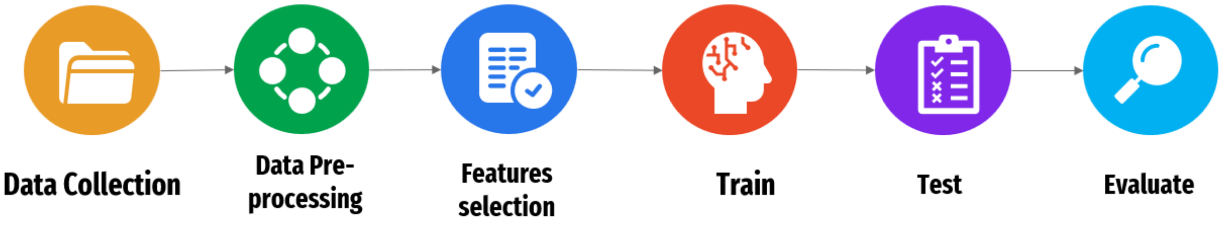


Figure 11: Machine Learning Pipeline [5]

5.7 Testing Plan

In the testing plan stage, we test our model in order to see how well our model is and if there is room for improvement. Performance metrics such as Accuracy, Recall, Precision and F1-Score as well as the reduction of false positives are the core data that will indicate how well the model is, furthermore this will aid in the optimization step and improvement of the model which will indicate the room for improvements. The targeted Accuracy would be more than 94% to be qualified as good accuracy, any percentage under the 94% margin will show that the model is insufficient and lacks necessary adjustments. For the Recall and Precision and F1-Score are also as same as the Accuracy to be precise. The model should be able to differentiate between false positives and real DDoS attacks, hence the necessity for the false positive's percentage being very close or 0% to be credible and authentic.

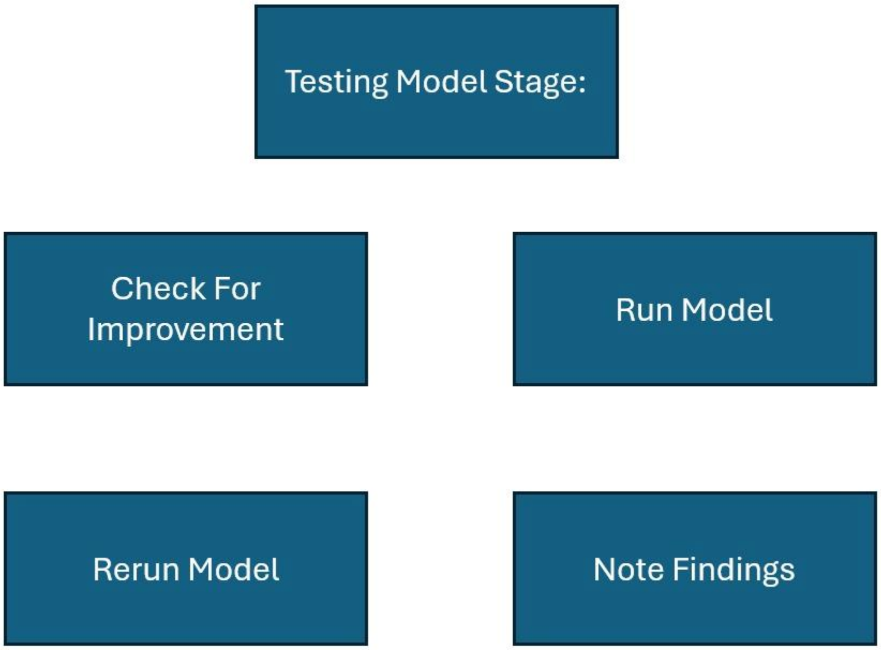


Figure 12: Testing plan stages for our project

5.4.1 Data collection

A comprehensive data collection process forms the foundation of our project since it determines how well our machine learning models will function based on the quality of the dataset selected. Our main task focused on obtaining research data representing genuine DDoS attacks which also holds a validated status and enjoys widespread research usage.

Our team conducted a comprehensive review through Google Scholar for research papers which attained high citation stats among respected members of cybersecurity and machine learning fields. The selection process focused on studies that made available datasets which numerous research works have utilized along with providing references for ensuring reliable and consistent analysis. Our selection of the CICDDoS2019 dataset followed a comprehensive review process because researchers commonly use it for DDoS detection research.

5.4.1.1 Data description

CICDDoS2019 includes both benign and contemporary standard DDoS attacks that closely represent actual world PCAPs. The analysis leads include time stamps along with source IPs and destination IPs and source and destination ports and protocols and attack types supplied using CICFlowMeter-V3 and CSV files. [7]

As their main focus they dedicated efforts toward creating background traffic that resembled realistic scenarios. The B-Profile system developed by Sharafaldin et al. (2016) has generated naturalistic benign background traffic within the proposed testbed (Figure 2) by profiling abstract human behavior through its method (Sharafaldin, et al. 2016). The abstract user behavior of 25 individuals was constructed by utilizing HTTP, HTTPS, FTP, SSH and email protocols. [7]

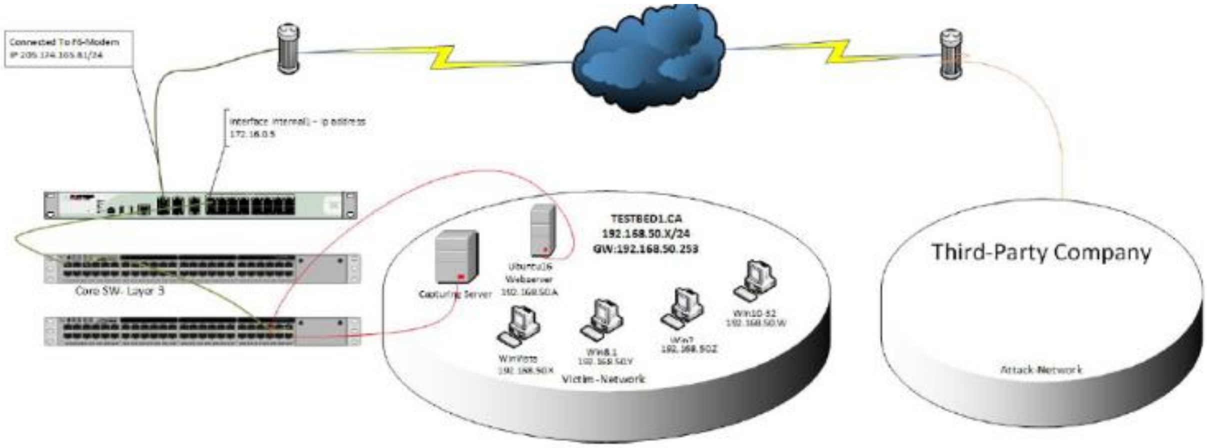


Figure 13: Testbed Architecture [7]

5.4.1.1.1 Number of Attributes & Feature Description

The dataset consists of more than 80 various attributes which originated from network traffic log analysis. The features in the dataset fall into three categories which comprise packet-based characteristics together with flow-based indicators and statistical properties for detecting normal and malicious activities.

Key Feature attributes:

1. **Basic Flow Features**
   - Flow Duration – Total duration of the network flow.
2. **Time-Based Features**
   - Flow IAT Mean – Average time between packet arrivals in a flow.
   - Fwd IAT Mean – Mean inter-arrival time for forward packets.
3. **Packet Size Features**
   - Fwd Packet Length Max – Maximum packet size in the forward direction.
4. **Flow Behavior & Statistical Features**
   - Fwd Packet Length Mean – Average packet size in the forward direction.
   - Flow Packets/s – Rate of packets per second in a flow.
5. **Header & Flag-Based Features**
   - SYN Flag Count – Number of packets with the SYN flag set.
6. **Label Attribute**
   - Label – Identifies whether the traffic is **normal or a specific DDoS attack type**.

5.4.2 Pre-processing:

Data pre-processing is a critical step that must be applied in machine learning before training the models on the dataset to get the best results, and it contains four stages which are data Integration, data cleaning, data reduction, data transformation [6]

5.4.2.1 Data Integration

We discovered that the CICDDoS2019 dataset contained nineteen Excel files when we downloaded it which held network traffic records. We integrated all files into one dataset through data processing to optimize model training operations.

The merged information expanded to reach a 50GB size with more than 50 million records of network traffic data. The data integration step combines all critical attack and normal traffic records to establish a unified dataset for extensive training process analysis.

5.4.2.1 Data cleaning

The following step involved cleaning the CICDDoS2019 dataset which integrated into a single file for improving its quality and consistency. We applied the following cleaning approaches on the dataset since it contained inconsistent information

- **Removing Duplicates:** We deleted duplicate records to control model bias through removal of redundant information.
- **Handling Missing Values:** We eliminated both missing and incomplete data from rows to keep the dataset free from inaccuracies and unreliable data points.
- **Removing Empty Rows:** We eliminated all completely empty rows through this process which helped minimize unnecessary computing costs and maintain data validity.

5.4.2.3 Data reduction

We performed data reduction through random sampling of 1 million records from the 50 million-row dataset and maintained class balance for the data. Data sampling of one million records enabled us to handle a reduced dataset while retaining important information.

5.4.2.4 Data transformation

Our procedure converted the two-class nominal targets into numeric format with 0 and 1 values to enable model training in machine learning.

Additionally, Excel performed an automatic conversion of big numbers to scientific notation while maintaining the unchanged actual value quantities. Data transformation enables more effective handling of substantial numeric data sets

5.4.3 Feature engineering

Feature Engineering is the process of improving ML model performance that involves selecting and transforming existing data features along with generating new features from the data. The practice involves analyzing the domain together with data and the specific problem statement to build features which help models identify critical data relationships. This process includes two fundamental methods: first selecting specific features while the second method extracts new features from existing data. [5] [6]

5.4.3.1 Feature selection:

We performed feature selection by eliminating Source IP, Destination IP as well as Similarity HTTPS from our dataset. The model excluded these features since they either provided irrelevant information or they were redundant or introduced unnecessary bias.

5.4.3.2 Feature extraction:

No feature extraction techniques were applied in our approach, as we did not create new features or transform existing ones into different representations.

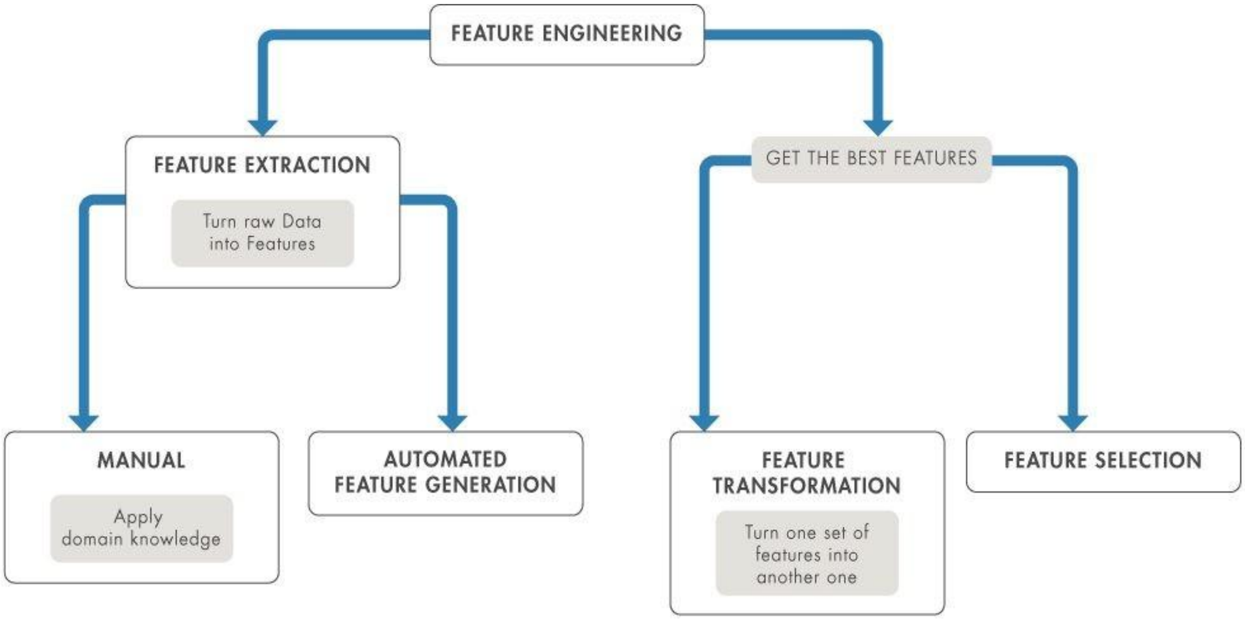


Figure 14 Feature Engineering structure [5]

5.5 Building Machine Learning models:

After applying the pre-processing phase, building the models phase started. To gain the best results and performance, several models were bult using different Machine Learning algorithms. The results of each model recorded first before the pre-processing phase. After that, they trained with the processed data. The performance of each model measured with Accuracy, Precision, Recall, F1 score, and False positive rate.

We divided the extensive dataset into various chunks while each portion contained 250,000 rows. We adopted this data partitioning method for maximum computational efficiency on our available hardware platform to achieve better execution speed. Data was partitioned so 70% served for training with the remaining 30% reserved for testing purposes.

5.5.1 Naïve Bayes

The Naive Bayesian (NB) classifier implements Bayes' theorem for object classification while NB identifies available knowledge to determine specific outcome probabilities because its Independence Assumptions lead the algorithm to view all features and attributes apart from each other. NB presents an easy-building model which performs better with big datasets compared to various sophisticated algorithms that Weka offers a Naïve Bayesin tool for implementing NB. [3]

5.5.2 Random Forest

The automated Random Forest system functions as a very strong technique. The model reaches high levels of success through the utilization of a small dataset. NB can solve both regression and classification issues as part of its operation. The classification problems will perform feature selection through information gain or gain ratio indexes or Gini index methods. The algorithm selects the major voting class among its options. [2] [3]

5.5.3 AdaBoost

The ensemble learning algorithm Adaptive Boosting (AdaBoost) joins weak classifiers into one strong classifier through multiple iterations. The algorithm changes the weights of wrong classification instances to raise their weighting importance for following training rounds. The classification performance improvement algorithm known as AdaBoost finds its most common application when using decision trees as weak learners. The algorithm stands out because it decreases both biases and variances while improving overall model sturdiness. [3] [4]

5.5.4 CatBoost

Gradient boosting algorithm Categorical Boosting (CatBoost) operates specifically to optimize efficiencies involving categorical data sets. CatBoost represents an advancement over typical gradient boosting approaches since Yandex's development team implemented ordered boosting to prevent model overfitting and these enhancements include creative encoding methods for features with categories. CatBoost stands out because it delivers high accuracy together with rapid training speed while managing very large datasets through minimal modification of parameters. [4]

5.5.5 Logistic Regression

The statistical model named Logistic Regression allows users to classify binary inputs. The sigmoid function evaluates the probability of an input belonging to a particular class. Logistic regression functions effectively with basic models even though it maintains simplicity when basic linear boundaries fit the problem. This technique finds applications in spam detection as well as medical diagnosis and fraud detection because of its ability to provide easy interpretation and efficient processing. [4]

5.5.6 XGBoost

Extreme Gradient Boosting (XGBoost) serves as an advanced machine learning tool which uses the gradient boosting framework to operate. The system optimizes speed by parallel processing and regulizes performance while efficiently processing missing values and data. The use of XGBoost spreads across competitions and practical implementations because it delivers top-level outcomes for classification alongside regression tasks. [2] [4]

5.5.7 KNN

The k-Nearest Neighbors (KNN) algorithm functions as a basic instance-based learning approach which performs classification tasks as well as regression activities. The algorithm selects k nearest data points from an input while assigning the majority class label from this selection. KNN demonstrates non-parametric characteristics which enable it to handle diverse data distributions though its operation becomes slower when processing big data collections. [1] [2] [5]

5.5.8 Decision Tree

As a supervised learning method, the Decision Tree (DT) algorithm operates through tree-like structure nodes where each decision depends on feature values. The algorithm divides data incrementally into subsegments through Gini Index or Information Gain or Gain Ratio evaluation which stops when it obtains the final classification result. At the top of the tree structure the root functions as the most important variable whereas the final endings or leaves display the predicted output classes. Due to their interpreter-friendly arrangement DTs provide usable visual displays which help both feature selection and classification efforts. The algorithm has difficulties with overfitting when excessive dimensions make up the tree yet lack appropriate pruning techniques. The algorithm finds its main applications in medicine to diagnose patients as well as credit assessment and the detection of intrusions within systems. [2] [5]

5.5.9 ANN

The computational design of Artificial Neural Networks (ANNs) derives from the structure of human brain neurons. Each Artificial Neural Network (ANN) uses connected nodes (neurons) to process data within weighted connection paths between nodes. ANNs serve as standard tools for solving difficult pattern recognition problems including image processing and speech recognition together with anomaly detection. The implementation of ANN-based classification through Multilayer Perceptron (MLP) which exists in Weka forms the main approach of our study. [1] [5]

5.5.10 SVM

Support Vector Machine provides advanced supervised learning capabilities that enable classification as well as regression operations. SVM uses an algorithm that identifies the best hyperplane which creates the largest margin between dataset classes. SVM succeeds particularly well in dimensions with many variables and operates effectively on data points which cannot be separated by simple lines through kernel functions including polynomial and radial basis function (RBF). The SVM algorithm finds uses in three domains: bioinformatics research, text classification analysis and fraud detection projects. [2] [3][5]

5.5.11 Result and Finding

Before pre-processing and feature engineering we ran a test for sample dataset (1 million) randomly, and this is the result that appeared with us:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**FPR**
NB	0.468	0.561	0.468	0.497	0.874
RF	0.628	0.623	0.628	0.676	0.545
DT	0.521	0.453	0.632	0.513	0.543
AdaBoost	0.446	0.436	0.446	0.499	0.612

CatBoost	0.632	0.628	0.632	0.683	0.879
LR	0.578	0.566	0.578	0.622	0.859
XGBoost	0.633	0.629	0.633	0.677	0.954
KNN	0.716	0.717	0.716	0.755	0.244
ANN	0.593	0.633	0.593	0.632	0.168
SVM	0.582	0.629	0.582	0.656	0.142

Until now we have walked with the research steps, and we have appeared with these results yet after applying data pre-processing and feature engineering:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**FPR**
NB	0.951	0.921	0.992	0.955	0.854
RF	0.984	0.986	0.997	0.991	0.145
DT	0.991	0.987	0.997	0.995	0.141
AdaBoost	0.946	0.945	0.999	0.971	0.582
CatBoost	0.918	0.920	0.995	0.956	0.861
LR	0.921	0.928	0.990	0.958	0.768
XGBoost	0.915	0.916	0.998	0.955	0.914
KNN	0.891	0.857	0.939	0.896	0.157
ANN	0.941	0.946	0.934	0.940	0.053
SVM	0.988	0.992	0.995	0.994	0.081

5.5 Discussion

The implementation of different machine learning algorithms helped achieve enhanced detection performance for DDoS attacks after applying strong preprocessor methods and creating engineered features.

The study confirms that proper data handling techniques result in better performance of ML models for identifying benign and malicious traffic. When the model was created, the main objective of it is to detect DDoS attacks with high accuracy, precision, recall, F1 – score and finally reduced false positive rate. Many ML techniques can perform either poorly or greatly depending on the labels in the dataset. With a little improvement to the labels in a dataset the performance could enhance greatly, which will be shown in this section.

After data preprocessing and feature engineering and the optimization techniques, the ML techniques with the highest performance metrics were Decision Tree, Support Machine Vector and Random Forest. Decision Tree had a very low accuracy for starters due to an unbalanced dataset with many labels that could possibly distract the ML techniques to perform in its best possible way, the accuracy of Decision Tree became 99% which increased significantly after the optimization process along with the other two ML techniques which also had a 98.8% for the SVM and 98.8% for the Random Forest. The optimization also increased the other performance metrics significantly.

We compared our model to an existing paper that showed that the KNN got an accuracy of 89% and the ANN got an accuracy of 93.6%, while our model enhanced the accuracy figure and provided improved results. KNN accuracy was 89.1% and the ANN improved to 94.1% with just optimization for the dataset.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
KNN	0.89	0.89	0.91	0.92
ANN	0.93	0.94	0.93	0.94

Figure 15: Result of the research paper that we are compared with [1]

6. conclusion

6.1 introduction

We performed a complete examination of machine learning algorithms to determine their ability for detecting Distributed Denial of Service (DDoS) attacks within this research. The necessity for intelligent adaptive detection methods in cybersecurity has surged be-cause DDoS attacks keep becoming more complex in traditional and emerging digital infrastructures. The research constructed the framework consisting of various supervised ML models which achieve accurate DDoS pattern detection with minimal wrong alert activations from network traffic analysis.

The research paper had a single defined purpose which was to measure machine learning algorithms' capacity for detecting DDoS attacks from labeled network traffic datasets. The CICDDoS2019 dataset served our needs because it delivered exceptional realism and academic validity as well as detailed dataset features. The dataset included equal numbers of records from benign traffic and attack sources which made it an ideal choice for detecting security threats under real-world simulation.

Multiple crucial phases composed the research process beginning with information acquisition followed by information merging before handling data. Our model training became efficient after we processed raw traffic logs through thorough data cleaning and noise reduction followed by transformation procedures. Model generalizability increased significantly after feature selection executed necessary removal of attributes that were unneeded or duplicative and irrelevant. Clean data processing methods transformed 50 million raw records into one million balanced data rows which became optimal for performing thorough experiments.

Our experiments utilized six machine learning classifiers namely Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) as well as Artificial Neural Networks (ANN). The trained models received performance evaluation through accuracy and precision metrics as well as recall and F1-score and false positive rate (FPR).

The first assessments of raw data produced mediocre results mainly due to elevated FPR and limited accuracy in differentiating hostile from reputable traffic. The ML pipe-line's complete implementation starting from preprocessing and feature engineering yielded substantial enhancement of model performance. The Decision Tree model outperformed other models by attaining a 99% accuracy level and a 99.5% F1-score while Random Forest and SVM achieved close to identical results that reduced incorrect detection rates.

The findings reveal how important it is to maintain excellent data quality during preprocessing before machine learning applications because poor results can waste re-sources and cause security threats to be missed particularly in cybersecurity. Simple Na-ïve Bayes models demonstrated successful performance when researchers applied the models to properly prepared data which proves that model complexity is not the essential factor for effective solutions.

Research findings received backup from the literature review which showed ML and DL approaches growing in popularity for DDoS detection. The analyzed studies in the re-view established how ensemble models along with anomaly detection methods and hybrid methodologies perform effectively in IoT and cloud-based systems. Studies integrating machine learning with Software-Defined Networking (SDN) and transfer learning techniques produced positive findings that provided understanding for creating adaptive defense systems with real-time detection capabilities.

Finally, the research showed that present methods face prolonged understanding deficits. Existing research mainly depends on labeled datasets that stay fixed while enclosing little investigation of dynamic learning environments that can evolve in real time. Few research investigations focus on enabling their models to learn from continuous streaming or unlabeled traffic data and to detect different evolving attack types during minimal resets.

5.2 Findings & Contributions

In the current semester, our work has laid the basis for the construction of a machine learning framework for DDoS attack detection. Some impressive contributions were made during this phase from which future work can be built. Our key contributions are summarized as follows:

- **Comprehensive Gap Analysis**:

  Performing a gap analysis helped us to define the problems of recognizing DDoS attacks with the help of existing methods. The analysis involved exploration of the weaknesses of the general detection approaches, including high false positives, inability to scale, and time inefficiencies. The information derived from this review informed decisions to target machine learning approaches when addressing this issue.

- **Extensive Literature Review**:

  In this study, we provided more than 20 research papers and technical articles studying DDoS attack detection, machine learning, and cybersecurity frameworks. This survey assisted us in determining the current position of DDoS detection and choosing the best algorithms and methodologies for our work. Our research findings were documented systematically to advance a rich theoretical framework for our work.

- **Project Scope and Methodology Definition**:

  In defining the modelling scope, we restricted the project to testing machine learning data using public datasets. Responding to the first research question, we provided a step-by-step approach that encompasses data cleansing, modeling and evaluation, and algorithm comparison. This kind of structure helps to avoid blurriness and aligns well with the objectives of our project.

- **Dataset Exploration and Preprocessing Plan**:

  Even though the project did not get to the implementation stage, we had to look for some of the publicly available datasets and the preprocessing that would be required for the job. Some of these include data cleaning, normalization, feature engineering and Data balancing techniques which are most efficient for imbalanced datasets.

- **Proposal Development**:

  Actually, the work done in the latter part of the semester was spent on preparing a comprehensive project proposal. This encompasses identifying the goals, and potential early map of how these may be met. The proposal forms the framework of the subsequent phases of the project.

- **Collaboration and Planning**:

  Thus, we enhanced the collaborations in tasks and planning through meetings sharing responsibilities, as well as the cyclic review of the results provided to the group members. This helped to keep our project objectives well achieved within the set timeframe.

By doing those activities, we formed a basis for project, a prerequisite for efficient future activities. The features of the quantitative and qualitative methods were clear in the proposal, while the literature review and gap analysis centered on what we were researching and how we would proceed. These contributions represent an important step in the way to creating a proper DDoS detection system.

5.3 Issues Faced

Given the fact that now the company is operating within the proposal stage, the mining difficulties were the ones which referred mostly to the research, planning and preparation steps of the subsequent project stages. These concerns were not connected to testing or implementation given that both have yet to occur. The main challenges we faced are as follows:

- Identifying the appropriate machine learning algorithms: Identifying and evaluating appropriate algorithms for DDoS detection took considerable time and energy due to the identified gaps in the literature.
- Understanding technical concepts: Acquiring an understanding of cybersecurity threats, approaches to machine learning, and methods of evaluating the results generally took some time and effort to understand more profoundly.
- Dataset preparation: Although in this work, no first practical elucidation was done, some challenges which may be faced when working with the network traffic datasets such as imbalance data were theoretical to accomplish during the planning stage of this work.
- Literature review: Literature surveys to know the deficiencies in detection of DDoS using machine learning and to understand the existing state of art were indeed time-consuming as there was look at various papers intensively.
- Coordination and collaboration: Although, managing working meetings and coordinating the activities of the members of the team might be problematic because of the time differences and the utilization of the information technologies.

- Scope definition: Deciding to limit the functional requirements of the SMART system into what was achievable strictly through simulation and analysis brought a few ideas that needed further discussion regarding the objectives of the project and their practical feasibility.
- Proposal writing: Subdividing and writing the proposal to ensure the goals, approach, and expected impact of the project are clear and coherent and took multiple reviews.
- Working with the DDoS dataset and trying to align it with the project's core idea was difficult considering the shortage of time.
- Understanding machine learning, in the beginning, was challenging considering the many types of techniques available and trying to figure out which model is most suitable for our data.
- Searching for a functional machine learning algorithms, and making sure it runs without problem consumes too much time.
- Some models need a huge amount of computing power for the testing process, so a supercomputer was needed.
- Trying to keep the dataset balancing in all cases
- Improving accuracy by applying more processing techniques

5.4 Lessoned & Skills Learned

The three major things that were learnt were the first lesson learnt was the need to follow good documentation practices and citing. Finding tools such as Mendeley is helpful in increasing the quality and thus the credibility of our work with citations ordered and formatted correctly. From this, we learnt lessons on the importance of retaining the highest level of ethical practice in research and respecting other workers in the same field.

There is improvement in grouping and in the way people in a group communicate with each other within a given project. It was clear why fair work division, status reporting, and time management were crucial.

5.4.1 Technical Skills

- The level of knowledge about cybersecurity in general and about DDoS attack detection in particular has been raised.
- More so, we have got to understand how machine learning models can be used to approach cyber threats and attain an increased level of knowledge.
- We have got to know about some specific approaches of data preprocessing that are being used in cybersecurity, such as feature extraction as well as anomaly detection.
- Improved understanding of the data set nature of cybersecurity and specifics of data it deals with including traffic diagrams and attack profiles.

- Extended knowledge of Python language intermediate level with emphasis in security and artificial neural networks.
- How to deal with different errors We faced a considerable number of errors with plenty of different types when we were building our models. Thus, we gained lots of experience in debugging and error correction.
- Learn and implement models using Python programing. When we started this project, we didn’t deal with and use machine learning models. Thus, we are learning more about Python language and implementing our model at the same time.
- Learn how to prepare/apply/improve/conclude models of machine learning Prepare, apply, improve, and conclude different machine learning models with different kinds and categories.
- We found Python libraries which can support pre-processing operations.

Improvement in technical writing skills

5.4.2 Managerial Skills

- Effective task time allocation to complete each project within the stated duration as required by the organizations.
- Possessed the skills on teamwork by working with other team members in particular phases of a project.
- Improved communication internal and external through daily virtual meetings and documentation.
- Project management capabilities were enhanced, and project documentation procedures resulted in smooth control of task and goal execution.

6.5 Future work

REFRENCESS

[1] H. Wang and Y. Li, "Overview of DDoS Attack Detection in Software-Defined Networks," *IEEE Access*, vol. 12, pp. 38351-38362, Mar. 2024. doi: 10.1109/ACCESS.2024.3375395.

[2] B. Wang, Y. He, and Z. Shui, "Predictive Optimization of DDoS Attack Mitigation in Distributed Systems using Machine Learning," *ACM Preprint*, Apr. 2024. doi: 10.13140/RG.2.2.15938.39369.

[3] S. Wiguna, R. Maulana, D. Heriyanto, and L. Nur, "Comprehensive Analysis of DDoS Attacks and Detection Methods," *Int. J. Informatics Commun. Technol.*, vol. 6, no. 3, pp. 72-85, Dec. 2024.

[4] M. S. Raza, M. N. A. Sheikh, I. Hwang, and M. S. Ab-Rahman, "Feature-Selection-Based DDoS Attack Detection Using AI Algorithms," *Telecom*, vol. 5, no. 2, pp. 333-346, Apr. 2024. doi: 10.3390/telecom5020017.

[5] Z. Mahdi, N. Abdalhussien, N. Mahmood, and R. Zaki, "Detection of Real-Time Distributed Denial-of-Service (DDoS) Attacks on Internet of Things (IoT) Networks Using Machine Learning Algorithms," *Computer. Mater. Contin.*, vol. 80, no. 2, pp. 2139-2151, Aug. 2024. doi: 10.32604/cmc.2024.053542.

[6] C. Singh and A. K. Jain, "A Comprehensive Survey on DDoS Attacks Detection & Mitigation in SDN-IoT Network," *e-Prime - Adv. Electron. Eng. Electron. Energy*, vol. 8, pp. 100543, Apr. 2024. doi: 10.1016/j.prime.2024.100543.

[7] M. E. Manaa, S. M. Hussain, S. A. Alasadi, and H. A. A. Al-Khamees, "DDoS Attacks Detection Based on Machine Learning Algorithms in IoT Environments," *Inteligencia Artificial*, vol. 27, no. 74, pp. 152-165, Apr. 2024. doi: 10.4114/intartif.vol27iss74pp152-165.

[8] R. Bukhowah, A. Aljughaiman, and M. M. H. Rahman, "Detection of DoS Attacks for IoT in Information-Centric Networks Using Machine Learning: Opportunities, Challenges, and Future Research Directions," *Electronics*, vol. 13, no. 1031, Mar. 2024. doi: 10.3390/electronics13061031.

[9] A. Alomiri, S. Mishra, and M. AlShehri, "Machine Learning-Based Security Mechanism to Detect and Prevent Cyber-Attack in IoT Networks," *Int. J. Comput. Digit. Syst.*, vol. 16, no. 1, pp. 641-656, Aug. 2024. doi: 10.12785/ijcds/160148.

[10] P. Reddy, Y. Adetuwo, and A. K. Jakkani, "Implementation of Machine Learning Techniques for Cloud Security in Detection of DDoS Attacks," *Int. J. Comput. Eng. Technol.*, vol. 15, no. 2, pp. 25-34, Apr. 2024.

[11] Islam, T., Jabiullah, M.I., & Abid, M.H. "DDoS Attack Preventing and Detection with the Artificial Intelligence Approach." Conference Proceedings, 2023, pp. 1-12. doi:10.1007/978-3-030-98457-1_3.

[12] Lee, S.-H., Shiue, Y.-L., Cheng, C.-H., Li, Y.-H., & Huang, Y.-F. "Detection and Prevention of DDoS Attacks on the IoT." Applied Sciences, vol. 12, no. 12407, 2022. doi:10.3390/app122312407.

[13] Nadeem, M.W., Goh, H.G., Ponnusamy, V., & Aun, Y. "DDoS Detection in SDN Using Machine Learning Techniques." Computers, Materials & Continua, vol. 71, no. 1, 2022, pp. 772-791. doi:10.32604/cmc.2022.021669.

[14] Bushart, J., & Rossow, C. "Anomaly-based Filtering of Application-Layer DDoS Against DNS Authoritatives." Conference on Network Security, 2021, pp. 1-15.

[15] Alzahrani, R.J., & Alzahrani, A. "Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic." Electronics, vol. 10, no. 2919, 2021. doi:10.3390/electronics10232919.

[16] Wibowo, M.H.S., Al Ayubi, M.D., & Hardiansyah, N.A. "Real-Time Detection and Prevention of Slowloris DDoS Attacks Using Machine Learning." International Journal of Informatics and Communication Technology, vol. 12, 2024, pp. 35-36. doi:10.11591/ijict.v12i1.

[17] Modi, P. "Towards Efficient Machine Learning Method for IoT DDoS Attack Detection." International Journal of Computer Science and Information Technology, 2023, pp. 1-10.

[18] Chakraborty, M., & Gupta, S. "Detection and Investigation of DDoS Attacks in Network Traffic Using Machine Learning Algorithms." International Journal of Innovative Technology and Exploring Engineering, vol. 11, no. 5, 2022. doi:10.35940/ijitee.F9862.0511622.

[19] Kumar, D., Pateriya, R.K., Gupta, R., Dehalwar, V., & Sharma, A. "DDoS Detection Using Deep Learning." Procedia Computer Science, vol. 218, 2023, pp. 2420-2429. doi:10.1016/j.procs.2023.01.217.

[20] Yungaicela-Naula, N.M., Vargas-Rosales, C., & Perez-Diaz, J.A. "SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning." IEEE Access, vol. 9, 2021, pp. 108495-108509. doi:10.1109/ACCESS.2021.3101650.

[21] Al-Shareeda, M.A., Manickam, S., & Saare, M.A. "DDoS Attacks Detection Using Machine Learning and Deep Learning Techniques: Analysis and Comparison." Bulletin of Electrical Engineering and Informatics, vol. 12, no. 2, 2023, pp. 930-939. doi:10.11591/eei.v12i2.4466.

[22] Ismail, M., Mohmand, M.I., Hussain, H., Khan, A.A., Ullah, U., Zakarya, M., Ahmed, A., Raza, M., Rahman, I.U., & Haleem, M. "A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks." IEEE Access, vol. 10, 2022, pp. 21443-21457. doi:10.1109/ACCESS.2022.3152577.

[23] Anley, M.B., Genovese, A., Agostinello, D., & Piuri, V. "Robust DDoS Attack Detection With Adaptive Transfer Learning." Computers & Security, vol. 144, 2024, pp. 103962. doi:10.1016/j.cose.2024.103962.

[24] Wagner, D., Kopp, D., Wichtlhuber, M., Dietzel, C., Hohlfeld, O., Smaragdakis, G., & Feldmann, A. "United We Stand: Collaborative Detection and Mitigation of Amplification DDoS Attacks at Scale." Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 970-988. doi:10.1145/3460120.3485385.

[25] Nadeem, M., Arshad, A., Riaz, S., Band, S.S., & Mosavi, A. "Intercept the Cloud Network From Brute Force and DDoS Attacks via Intrusion Detection and Prevention System." IEEE Access, vol. 9, 2021, pp. 152300-152314. doi:10.1109/ACCESS.2021.3126535.

[26] Jaszcz, A., & Połap, D. "AIMM: Artificial Intelligence Merged Methods for Flood DDoS Attacks Detection." Journal of King Saud University – Computer and Information Sciences, vol. 34, 2022, pp. 8090-8101. doi:10.1016/j.jksuci.2022.07.021.

[27] Najafimehr, M., Zarifzadeh, S., & Mostafavi, S. "DDoS Attacks and Machine-Learning-Based Detection Methods: A Survey and Taxonomy." Engineering Reports, vol. 5, 2023. doi:10.1002/eng2.12697.

[28] Kalutharage, C.S., Liu, X., Chrysoulas, C., Pitropakis, N., & Papadopoulos, P. "Explainable AI-Based DDOS Attack Identification Method for IoT Networks." Computers, vol. 12, no. 32, 2023, pp. 1-16. doi:10.3390/computers12020032.

[29] Alshahrani, M.M. "A Secure and Intelligent Software-Defined Networking Framework for Future Smart Cities to Prevent DDoS Attack." Applied Sciences, vol. 13, no. 9822, 2023. doi:10.3390/app13179822.

[30] Maksimović, A.N., Nikolić, V.R., Vidojević, D.V., Randjelović, M.D., Djukanović, S.M., & Randjelović, D.M. "Using Triple Modular Redundancy for Threshold Determination in DDOS Intrusion Detection Systems." IEEE Access, vol. 12, 2024, pp. 53785-53795. doi:10.1109/ACCESS.2024.3384380.

[31] A. Sharma, "Decision Tree vs. Random Forest – Which Algorithm Should you Use?," 12 5 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-treevs-random-forest-algorithm/.

[32] M. Di Capua, E. Di Nardo and A. Petrosino, "Unsupervised cyber bullying detection in social networks," IEEE, 2017.

[33] T. Kanan, A. Aldaaja and B. Hawashin, "Cyber-Bullying and Cyber-Harassment Detection Using Supervised Machine Learning Techniques in Arabic Social Media Contents," Journal of Internet Technology, 2020.

[34] A. Pant, "Introduction to Logistic Regression," towards data science, 22 january 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-logistic-regression66248243c148.

[35] Shneiderman B, “The Eight Golden Rules of Interface Design,” 2016. [Online]. Available:

https://www.cs.umd.edu/users/ben/goldenrules.html. [Accessed: 15-Nov-2018].

[36] Random Forest Image

[1] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS Detection using Deep Learning," *Procedia Computer Science*, vol. 218, pp. 2420–2429, 2023. doi: 10.1016/j.procs.2023.01.217.

[2] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, "SDN-Based Architecture for Transport and Application Layer DDoS Attack Detection by Using Machine and Deep Learning," *IEEE Access*, vol. 9, pp. 108495–108510, 2021. doi: 10.1109/ACCESS.2021.3101650.

[3] B. Mahesh, "Machine Learning Algorithms—A Review," *Int. J. Sci. Res. (IJSR)*, vol. 9, no. 1, pp. 381-386, Jan. 2020. doi: 10.21275/ART20203995.

[4] T. O. Ayodele, "Types of Machine Learning Algorithms," in *New Advances in Machine Learning*, IntechOpen, 2010. doi: 10.5772/9385.

[5] Bishop, C. M. Neural Networks for Pattern Recognition. New York: Oxford University Press (1995). (This book offers a good coverage of neural networks)

[6] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big Data Preprocessing: Methods and Prospects," *Big Data Analytics*, vol. 1, no. 9, 2016. doi: 10.1186/s41044-016-0014-0.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2019, pp. 53–64. doi: 10.5220/0006639805780585.

MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY
قسم علوم الحاسب
Department of Computer Science

وزارة التعليم
جامعة الإمام
عبدالرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 511 – Project proposal
Term 1 – 2024-2025
Status Report #
Period from 18/8/2024 to 1/12/2024

Group #:

Members of the group:

#	Name	ID	Role
1	Nasser Amer Alhajri	2210006797	Leader
2	Rayyan B. Al Nahwi	2210002103	Member
3	Ibrahim A. Al Husaini	2210002314	Member
4	Aseel A. Alrudayni	2210001894	Member
5	Noor A. Hadari	2210002310	Member

1. The outcome of the meeting
    - Find a name and logo for the project.
    - Discussed The topic and get a lot of information about it.
    - Agreed to complete the first chapter on the project.
    - Divide the tasks among the members.
    - Asked questions to get more understanding for the reports.
2. Tasks planned to work on during the specified period:

S. No	**Assigned task**	**Members**	**Deadline**	**Status**
1	Read and understand the topic in depth	All	1/9/2024	Done
2	Finish the first chapter for proposal	All	4/9/2024	Done
3	Biweekly progress and introduction.	Nasser A Alhajri	4/9/2024	Done
4	Problem statement and Background.	Rayyan B Al Nahwi	4/9/2024	Done
5	Motivation and justification.	Aseel A. Alrudayni	4/9/2024	Done
6	Aims & objectives and Scope / Limitation of the Study	Noor A. Hadari	4/9/2024	Done
7	Social, Professional, legal, and ethical Implications Project Organization and Chapter summary	Ibrahim A. Al Husaini	4/9/2024	Done

3. We will learn about:

- Task Management.
- Project Prioritization.

- The Topic itself in depth.
- Machine Learning.
- Cyber Threats.

4. Pending tasks:

   None

- ✓ Number of group meetings you had during the specified time: All Members.

--------------------------------------- For supervisor use only --
----

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.1 Students demonstrate the abilities to participate in team activities.	The student does not contribute to discussions, does not let others express opinions	The student contributes occasionally to team activities	The student Contributes equally in team activities	The student Contributes a higher share to team activities without taking over the team

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.2 Students demonstrate the abilities to organize themselves and complete assignment to meet deadlines.	Students are unable to structure themselves as a team and meet the deadlines.	Mainly some individuals of team are contributing and able to meet some but not all requirements by the deadline	Team has a well-defined structure and is generally able to meet the deadlines.	Students structured themselves as a cohesive team and able to finish the project ahead of the time without compromising on quality of work.

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبدالرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

قسم علوم الحاسب
Department of Computer Science

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 511 – Project proposal
Term 1 – 2024-2025
Status Report #
Period from 18/8/2024 to 1/12/2024

Group #:

Members of the group:

#	Name	ID	Role
1	Nasser Amer Alhajri	2210006797	Leader
2	Rayyan B. Al Nahwi	2210002103	Member
3	Ibrahim A. Al Husaini	2210002314	Member
4	Aseel A. Alrudayni	2210001894	Member
5	Noor A. Hadari	2210002310	Member

5. The outcome of the meeting

- Each member searched for 5 research papers.
- Each member summarized each paper.
- Agreed to complete the second chapter on the project.
- Divide the tasks among the members.
- Literature review.

6. Tasks planned to work on during the specified period:

S. No	**Assigned task**	**Members**	**Deadline**	**Status**

1	Search for 5 research papers.	All	23/9/2024	Done
2	Finish the second chapter for proposal	All	5/10/2024	Done
3	Biweekly progress and introduction.	Nasser A Alhajri, Rayyan B Al Nahwi	5/10/2024	Done
4	Background for ch2.	Rayyan B Al Nahwi	5/10/2024	Done
5	Business impact	Nasser A Alhajri	5/10/2024	Done
6	Introduction for ch2	Noor A. Hadari	5/10/2024	Done
7	Literature review, Knowledge Gap.	Aseel A. Alrudayni, Ibrahim A. Al Husaini	5/10/2024	Done

7. We will learn about:
    - Task Management.
    - Project Prioritization.
    - The Topic itself in depth.
    - Machine Learning algorithms.
    - In-depth knowledge about Cyber Threats.
8. Pending tasks:

    None

✓ Number of group meetings you had during the specified time: 7.

-- For supervisor use only ---

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.1 Students demonstrate the abilities to participate in team activities.	The student does not contribute to discussions, does not let others express opinions	The student contributes occasionally to team activities	The student Contributes equally in team activities	The student Contributes a higher share to team activities without taking over the team

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.2 Students demonstrate the abilities to organize themselves and complete assignment to meet deadlines.	Students are unable to structure themselves as a team and meet the deadlines.	Mainly some individuals of team are contributing and able to meet some but not all requirements by the deadline	Team has a well-defined structure and is generally able to meet the deadlines.	Students structured themselves as a cohesive team and able to finish the project ahead of the time without compromising on quality of work.

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER SCIENCE
& INFORMATION TECHNOLOGY
قسم علوم الحاسب
Department of Computer Science

وزارة التعليم
جامعة الإمام
عبدالرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 511 – Project proposal
Term 1 – 2024-2025
Status Report #
Period from 18/8/2024 to 1/12/2024

Group #:

Members of the group:

#	Name	ID	Role
1	Nasser Amer Alhajri	2210006797	Leader
2	Rayyan B. Al Nahwi	2210002103	Member
3	Ibrahim A. Al Husaini	2210002314	Member
4	Aseel A. Alrudayni	2210001894	Member
5	Noor A. Hadari	2210002310	Member

9. The outcome of the meeting

- Divided the work evenly among the group.
- Figured out the risks of the project and added assumptions and constraints to our project.
- Identified the team roles in the project.

10. Tasks planned to work on during the specified period:

S. No	Assigned task	Members	Deadline	Status
1	Determine specific requirements.	All	18/10/2024	Done
2	Manage the process plan for the project.	All	20/10/2024	Done
3	Technical process plan.	All	20/10/2024	Done
4	Biweekly progress	Rayyan Al Nahwi	21/10/2024	Done

11. We will learn about:

- Task Management.
- Project Prioritization.
- Managing project plan.
- Project specification.

12. Pending tasks:

Dataset.

✓ Number of group meetings you had during the specified time: 4.

-- For supervisor use only --

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.1 Students demonstrate the abilities to participate in team activities.	The student does not contribute to discussions, does not let others express opinions	The student contributes occasionally to team activities	The student Contributes equally in team activities	The student Contributes a higher share to team activities without taking over the team

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.2 Students demonstrate the abilities to organize themselves and complete assignment to meet deadlines.	Students are unable to structure themselves as a team and meet the deadlines.	Mainly some individuals of team are contributing and able to meet some but not all requirements by the deadline	Team has a well-defined structure and is generally able to meet the deadlines.	Students structured themselves as a cohesive team and able to finish the project ahead of the time without compromising on quality of work.

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

CS 511 – Project proposal
Term 1 – 2024-2025
Status Report #
Period from 18/8/2024 to 1/12/2024

Group #:

Members of the group:

#	Name	ID	Role
1	Nasser Amer Alhajri	2210006797	Leader
2	Rayyan B. Al Nahwi	2210002103	Member
3	Ibrahim A. Al Husaini	2210002314	Member
4	Aseel A. Alrudayni	2210001894	Member
5	Noor A. Hadari	2210002310	Member

13. The outcome of the meeting

- Work on methodology and design
- Each member searched for a data set that we can apply to.
- Agree on site design and the components of website
- Agreed to complete the Fourth chapter on the project.
- Divide the tasks among the members.

14. Tasks planned to work on during the specified period:

S. No	**Assigned task**	**Members**	**Deadline**	**Status**
1	Methodology framework and introduction	Nasser A Alhajri, Ibrahim A Alhusaini	15/11/2024	Done
2	System design	Rayyan B Al Nahwi, Nasser A Alhajri	17/11/2024	Done
3	Dataset search and comparing to apply on it	All	16/11/2024	Done
4	User interface design and plan	Aseel A. Alrudayni, Noor A. Hadari	16/11/2024	Done
5	Biweekly progress	Ibrahim A Alhusaini	17/11/2024	Done

15. We will learn about:

- Task Management.
- Using Figma for UI/UX design
- Dealing with data set

- Front-end and back-end

16. Pending tasks: None

✓ Number of group meetings you had during the specified time: **2**

--- For supervisor use only --

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.1 Students demonstrate the abilities to participate in team activities.	The student does not contribute to discussions, does not let others express opinions	The student contributes occasionally to team activities	The student Contributes equally in team activities	The student Contributes a higher share to team activities without taking over the team

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.2 Students demonstrate the abilities to organize themselves and complete assignment to meet deadlines.	Students are unable to structure themselves as a team and meet the deadlines.	Mainly some individuals of team are contributing and able to meet some but not all requirements by the deadline	Team has a well-defined structure and is generally able to meet the deadlines.	Students structured themselves as a cohesive team and able to finish the project ahead of the time without compromising on quality of work.

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

MINISTRY OF EDUCATION | وزارة التعليم
IMAM ABDULRAHMAN BIN | جامعة الإمام
FAISAL UNIVERSITY | عبدالرحمن بن فيصل
COLLEGE OF COMPUTER SCIENCE | كلية علوم الحاسب
& INFORMATION TECHNOLOGY | وتقنية المعلومات
قسم علوم الحاسب
Department of Computer Science

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CS 511 – Project proposal
Term 1 – 2024-2025
Status Report #
Period from 18/8/2024 to 1/12/2024

Group #:

Members of the group:

#	Name	ID	Role
1	Nasser Amer Alhajri	2210006797	Leader
2	Rayyan B. Al Nahwi	2210002103	Member
3	Ibrahim A. Al Husaini	2210002314	Member
4	Aseel A. Alrudayni	2210001894	Member
5	Noor A. Hadari	2210002310	Member

17. The outcome of the meeting

- Agreed to complete the last chapter on the project.
- Divide the tasks among the members.
- Choosing a dataset to work on.
- Finishing the diagrams needed

18. Tasks planned to work on during the specified period:

S. No	**Assigned task**	**Members**	**Deadline**	**Status**
1	Finishing the conclusion	All	12/12/2024	Done

2	Choosing a dataset	All	9/12/2024	Done
3	Biweekly progress	All	12/12/2024	Done
4	Sequence Diagram	All	12/12/2024	Done

19. We will learn about:

    - Project Prioritization.
    - How to choose the proper dataset

20. Pending tasks:

    None

✓ Number of group meetings you had during the specified time: 3.

-- For supervisor use only ---

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.1 Students demonstrate the abilities to participate in team activities.	The student does not contribute to discussions, does not let others express opinions	The student contributes occasionally to team activities	The student Contributes equally in team activities	The student Contributes a higher share to team activities without taking over the team

Std #	Rubrics	Comments
1	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
2	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
3	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
4	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	
5	□ Poor or Non-existent □ Developing □ Developed □ Exemplary	

Performance Indicators	Rubrics			
	Poor or Non-existent	Developing	Developed	Exemplary
5.2 Students demonstrate the abilities to organize themselves and complete assignment to meet deadlines.	Students are unable to structure themselves as a team and meet the deadlines.	Mainly some individuals of team are contributing and able to meet some but not all requirements by the deadline	Team has a well-defined structure and is generally able to meet the deadlines.	Students structured themselves as a cohesive team and able to finish the project ahead of the time without compromising on quality of work.

Std #	Rubrics	Comments
1	☐ Poor or Non-existent ☐ Developing ☐ Developed ☐ Exemplary	
2	☐ Poor or Non-existent ☐ Developing ☐ Developed ☐ Exemplary	
3	☐ Poor or Non-existent ☐ Developing ☐ Developed ☐ Exemplary	
4	☐ Poor or Non-existent ☐ Developing ☐ Developed ☐ Exemplary	
5	☐ Poor or Non-existent ☐ Developing ☐ Developed ☐ Exemplary	