A HYBRID INTRUSION DETECTION MODEL FOR APPLICATION LAYER DDOS ATTACKS BASED ON K-MEANS AND CART ALGORITHMS

BY

VICTOR KIPNGETICH CHERUIYOT

A RESEARCH THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE DEGREE IN INFORMATION TECHNOLOGY, DEPARTMENT OF INFORMATION TECHNOLOGY, SCHOOL OF INFORMATION SCIENCES

MOI UNIVERSITY

2026

DECLARATION

DECLARATION BY THE CANDIDATE

This thesis is my original work and has not been presented in any other University for a degree or any other award. No part of this thesis may be reproduced without the prior permission of the author and/or Moi University.

18/02/2026

Victor Kipngetich Cheruiyot — Date

Reg No: MS/IT/7784/20

DECLARATION BY THE SUPERVISORS

This thesis has been submitted for examination with our approval as the University supervisors:

19/02/2026

Dr. John K. Tarus — Date

Department of Information Technology,

School of Information Sciences, Moi University.

18/02/2026

Dr. Irene Moseti — Date

Department of Information Technology,

School of Information Sciences, Moi University.

19/02/2026

Dr. Shadrack K. Metto — Date

Department of Information Technology,

School of Information Sciences, Moi University.

DEDICATION

To my family and friends,

I dedicate this thesis to all of you with utmost gratitude and appreciation. I would like to thank all of you for your steadfast encouragement, support, and love in various ways, which has been a significant source of inspiration and motivation.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported me throughout the journey of completing this thesis.

Firstly, I extend my heartfelt appreciation to Benjamin Kebenei for his generous financial support, which enabled me to pursue my studies with ease.

I am also grateful to my supervisors, Dr. John K. Tarus, Dr. Irene Moseti and Dr. Shadrack Metto, for their invaluable guidance, unwavering support, and constructive feedback throughout my research work. Their expertise and insight have been instrumental in shaping this thesis.

I am also grateful to Dennis Bor for his excellent tutoring and guidance in data science, which enabled me to acquire invaluable knowledge and skills vital in conducting this research.

Additionally, I would like to acknowledge all those who have contributed to this thesis in various ways, including offering their time, expertise, and resources. Your support has been immeasurable and highly appreciated.

Finally, I acknowledge that this achievement would not have been possible without God's blessings and providence. I am humbled and thankful for all that He has done for me.

ABSTRACT

The increase in interconnectivity and advancement in network technologies have influenced a parallel rise in Distributed Denial of Service (DDoS) attacks, and the perpetrators have become sophisticated such that previously dependable tools and techniques have become ineffective. The purpose of the study was to design an intrusion detection model based on K-Means and CART algorithms, and train and test it using the CICDDoS2019 dataset, which represents application-layer DDOS attacks. The objectives of the study were to: Determine the existing application-layer intrusion detection techniques and models; Explore the weaknesses of existing intrusion detection models; Classify the dataset using individual K-Means and CART algorithms; Develop a hybrid intrusion detection model for application-layer DDoS attacks by combining K-Means and CART algorithms; and evaluate the performance of the hybrid model. The study was designed as a quantitative experimental simulation. It adopted the empirical positivist paradigm. A machine learning theory and network security theory formed the theoretical framework. The Scikit-Learn libraries were employed using Python programming to perform the analysis. The study utilised secondary data obtained from the CICDDoS2019 dataset, containing 49.59 million records of 12 unlabelled DDoS attack types including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP. This research used simple random sampling to select 30000 records from each attack type, yielding a dataframe of 110,000 rows and 88 columns. The Unsupervised component of the experiment requires no training and testing sets. For the supervised component using the CART algorithm, the dataset was split into 67% for training and 33% for testing. Individually, the K-Means algorithm achieved homogeneity, completeness, and V-measure scores of 50.76%, 51.95%, and 51.35% respectively. On the other hand, CART was measured on accuracy, precision, recall/sensitivity, and F1-Score and it achieved scores of 74% on all counts. The hybrid model was fundamentally a CART algorithm improved by K-means clustered features and therefore was scored on the CART algorithm metrics basis. It scored 78% on accuracy, 79% on precision, 78% on recall, and 78.5% on F1-score. The dataset proved to have high dimensionality and complexity with multiple overlapping clusters. K-Means had an average performance proving its unsuitability for this type of dataset. CART algorithm had a relatively high success in identifying application layer DDoS attacks. The hybrid model achieved a better performance score compared to its constituent models as shown by the difference between the chosen metrics and their averages. This study concludes that our hybrid intrusion detection model can outperform existing K-Mean and CART algorithms in terms of accuracy, precision, recall and F1 score. The study recommends that future studies should investigate a similar model using density-based clustering algorithms like DBSCAN in place of K-Means in a similar setup.

LIST OF ABBREVIATIONS AND ACRONYMS

C4.5	-	Version 4.5 of a decision tree algorithm created using C.
CART	-	Classification and Regression Tree
CIC	-	Canadian Institute of Cybersecurity
CharGen	-	Character Generator Protocol
CSV	-	Comma Separated Values
DBSCAN	-	Density-Based Spatial Clustering Application with Noise
DDoS	-	Distributed Denial of Service
DNS	-	Domain Name System
DNS	-	Domain Name System
DoS	-	Denial of Service
DT	-	Decision Tree
FTP	-	File Transfer Protocol
GB	-	Gigabyte
HTTP	-	Hypertext Transfer Protocol
HTTPS	-	Hypertext Transfer Protocol Secure
ICMP	-	Internet Control Message Protocol
ID3	-	Iterative Dichotomiser 3
IDS	-	Intrusion Detection System
IP	-	Internet Protocol
LDAP	-	Lightweight Directory Access Protocol
MSSQL	-	Microsoft SQL Server
NA	-	Not Available

NaN - Not a Number

NETBIOS - Network Basic Input/Output System

NTP - Network Time Protocol

OSI - Open Systems Interconnection

PAC - Packet Analyzer Console

PCA - Principal Component Analysis.

PII - Personally Identifiable Information

SMTP - Simple Mail Transfer Protocol

SNMP - Simple Network Management Protocol

SSDP - Simple Service Discovery Protocol

SSH - Secure Shell

SVM - Support Vector Machine

SYN - Synchronize

TCP - Transmission Control Protocol

TCP/IP - Transmission Control Protocol/Internet Protocol

TFTP - Trivial File Transfer Protocol

TTL - Time to Live

UDP - User Datagram Protocol

WCSSE - Within-Cluster Sum of Squared Errors

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

Recent years have seen an increase in the frequency, severity, and complexity of network attacks. This has been contributed, in part, by the increased number of devices connected to the internet. Vishwakarma and Jain (2019) argued that the Internet of Things (IoT) has become massive and encompasses multimillion devices in constant interaction. According to Singh and Jain (2018), Cisco and Ericsson predicted that there would be 50 billion devices on the internet by 2020. They also quote a CERT-India report that shows a relatively parallel increase in the cases of cybersecurity incidences with there being 44679, 49455, 5036, and 27482 cases in 2014, 2015, 2016, and 2017 respectively. Each new device increases the arsenal of network attackers and similarly, the vulnerabilities that have to be protected.

The resultant situation has precipitated a cyber arms race pitting cyber security experts and cyber attackers. This war has become especially high stakes since recent additions encompass systems of immense critical importance to essential service delivery. Rashid et al. (2019) noted that critical infrastructure is increasingly adopting technology and becoming integrated with other larger software systems and, subsequently, the Internet. Attackers are constantly looking for loopholes in the current security features to exploit. They are increasingly becoming innovative and evasive in their tactics. It has become very difficult to detect an attack that is in progress, let alone prevent one that has not already happened. Security experts across the globe are racing against the clock to

develop new measures to shield networks and the contained resources against known attacks, zero-day attacks, and maybe future intrusions.

Denial of Service attacks are increasingly becoming a global cyber threat. Recent attacks have shown how destructive, costly, and disruptive such attacks can be. According to Cybersecurity Ventures, cybercrimes would lead to a loss of $6 trillion by 2021, which is an increase of 100% from 2015's cost (Radev, 2019). The most common and severe form of denial-of-service attack is the Distributed Denial of Service (DDoS) attack. There was more than a 100% rise in Distributed Denial of Service (DDoS) attacks in 2017 (Vishwakarma & Jain, 2019). Such attacks are so severe that they can bring down even the most robust network infrastructure in the world because they can utilize a large number of devices to overload a server with traffic. A cyber-attack on Yahoo systems in 2014 led to immense losses which led to the exposure of the personal information of 500 million of its users (Radev, 2019). The attack cost the company billions of dollars as well as its reputation. Past attacks have mainly been DDoS attacks targeted at the network layer and the transport layer of the Open System Interconnection (OSI) model. The attacks have been commonly done using Internet Control Message Protocol (ICMP) flooding, Synchronize (SYN) flooding, and User Datagram Protocol (UDP) flooding (Obaid & Abeed, 2020; Kumar, 2016). Application layer attacks, by contrast, tend to use true IP addresses. Characterizing the nature of an attack can aid in the development of a solution. Indeed, there is never a silver bullet to address all possible attacks. There are, however, ways that such attacks can be prevented or their impact minimized.

One of the three types of DDoS attacks is the Application-Layer Attack involving a targeted attack on a server or a client mainly using direct web traffic. HTTP, HTTPS,

DNS, and/or SMTP are the most common avenues of this type of attack. Application-Layer Attacks are notoriously hard to detect as the smallest number of botnets can be used potentially fooling the traffic monitor into believing them to be high-volume legitimate traffic. Such attacks are problematic because they are camouflaged as genuine traffic, very efficient, require few resources to launch, can be focused, simple, and can affect several applications without breaking the network traffic rules (Obaid & Abeed, 2020). The advent and subsequent growth of Machine Learning have presented many solutions to previously difficult problems (Kumar, 2016). One such problem, that can be addressed by Machine Learning techniques, is Application-Layer DDoS Attacks.

Transport and Network layer DDoS attacks can easily be detected by scanning and analysing IP addresses, UDP ports, and TCP ports. Such techniques are ineffective when the attacker simulates a normal web user or many web users and uses them to request a resource from the application server. Due to their ingenuine ability to emulate a normal user and behave like one (by making a full and successful TCP connection), it is difficult to detect an application-layer DDoS attacker using IP and TCP attributes.

Over the years, various statistical and machine-learning techniques have been introduced, and their supporters have claimed success in the detection and prevention of network intrusion. However, with the advent of any new technologies, more sophisticated forms of cyberattacks come into existence, hence rendering the earlier methods less relevant. Among the most commonly employed techniques in developing Intrusion Detection Systems (IDS) are K-means clustering and decision tree algorithms. K-means works by organizing data into a pre-defined number of clusters, aiming to minimize internal variation within each cluster while maximizing distinction across clusters (Scikit-Learn,

n.d.-a). On the other hand, decision trees function by analysing training datasets to identify patterns, which are then used to classify similar incoming data (Scikit-Learn, n.d.-b). K-means is particularly effective for uncovering inherent groupings in unlabelled datasets, ensuring that elements within a cluster are closely related, whereas elements across clusters remain significantly distinct (Scikit-Learn, n.d.-a).

It is important to identify the right attributes to use in building a predictive model for cyber-attacks. As has been noted, application-layer DDoS attacks employ true IP addresses. Source and destination IP addresses therefore formed part of the attributes used in model development. Time To Live (TTL) is also another attribute that boosted the depth of detection. The characterization of normal web traffic was used to highlight divergence and hence put suspicious IP addresses in the spotlight (Sharafaldin et al., 2019).

Flagging suspected botnets first required that their inherent difference with normal traffic or users be properly defined. Several features can distinguish a legitimate user from a simulated one. Legitimate users behave differently while online. They access a specific resource and access it for a certain given time. Each user or group of users can be classified on the resource they access. Normal users also have a defined request rate and an average access frequency or request per session. Their total number and the size of requests per single session for normal users are also distinguishable. These features provided a basis for building a model that identified abnormal behaviours and identifying DDoS attacks on a network.

Although many researchers have been using K-means clustering for the building of hybrid IDS models, many have used K-means clustering as a stand-alone technique to

cluster IDS data into normal operations and attack incursions (Raval & Jani, 2016). Enhancements to the K-means algorithm have also been proposed to boost clustering precision and lower the rate of false positives. Meanwhile, Classification and Regression Trees (CART) are commonly used to develop hierarchical representations of both attack signatures and legitimate behaviours (Krzywinski & Altman, 2017). This method relies on selecting features that maximise information gain and on minimal data partitioning to build the decision tree. The present study adopts a combined model that integrates CART with K-means to form an unsupervised anomaly detection framework.

Conversely, other researchers used ID3 combined with *K*-means in a supervised attack detection, and more recently K-Means coupled with C4.5 (Muniyandi et al. 2012 as cited in Boutaba et al., 2018) in an unsupervised anomaly detection approach. ID3 (Iterative Dichotomizer 3) is an algorithm that was developed during the 1970s by Quinlan (Boutaba et al., 2018). It employs entropy maximization for the selection of features to use during classification of datapoints. ID3 is the precursor of C4.5 and C5.0 (Boutaba et al.). C4.5 is also the brainchild of Quinlan, developed in 1993 (Boutaba et al.). According to Reddy and Chittineni (2021), C4.5 algorithm uses gain ratio to split the datasets. This approach overcomes the problem posed by problems that present several outcomes. C4.5 uses information gain to reduce the subset entropy and gain ratio to split information through the assistance of test outcome data.

1.2 Statement of the problem

The behavior of illegal network traffic has been constantly evolving, becoming more and more disguised. It is very difficult today to distinguish between genuine network traffic

and illegitimate ones. Consequently, network management has become a daunting task. Distributed Denial of Service application layer attacks have, as a consequence, increased in severity and complexity in the past few years. Many methods of detection and prevention have been proposed and implemented over the years with varying levels of success.

Most existing techniques to resolve Distributed Denial of Service application layer attacks employed outdated datasets to train and test them (Sharafaldin et al., 2019), while others require vast amounts of computing resources to implement (Sze et al, 2020). Existing intrusion detection models are resource intensive and employ old datasets. K-Means and CART algorithms have been employed in varying techniques to deal with different issues with mixed results. Various combinations have been realized to improve network security. DDoS attacks, however, remain an unresolved problem for internet users.

1.3 Aim of the Study

To design an intrusion detection model based on K-Means and CART algorithms, and train and test it using the CICDDoS2019 dataset, which represents application-layer DDOS attacks.

1.4 Objectives

The objectives of the study were to:

i. To critically evaluate existing application-layer intrusion detection methods and models, determine their major strengths, weaknesses, and limitations in relation to contemporary DDoS attacks.
ii. Classify the dataset using individual K-Means and CART algorithms.
iii. Develop a hybrid intrusion detection model for application layer DDoS attacks by combining K-Means and CART algorithms.
iv. Evaluate the performance of the proposed hybrid model.

1.5 Research questions

i. What are the existing application-layer intrusion detection techniques and models, and what are their key weaknesses and potential improvements in the context of modern DDoS attacks?
ii. How do individual K-Means and CART algorithms perform in classifying intrusion detection datasets?
iii. How can K-Means and CART be combined into a hybrid model to detect application layer DDoS attacks?
iv. How effective is the hybrid K-Means and CART algorithms in detecting and preventing application layer DDoS attacks?

1.6 Scope of the Study

The given study is limited to the research problem of application-layer DDoS attacks detection with machine learning-based intrusion detection methods. In particular, the design, implementation, and evaluation of a hybrid intrusion detection model combining

the unsupervised K-Means clustering with supervised CART are considered in the research.

The assessment of the experiment is performed based on the CICDDoS2019 dataset that comprises labelled network traffic flows of benign traffic and various types of DDoS threats at the application-layer. To cope with the complexity of computations and to make it feasible, the study uses a representative sampled sub-set of the data, after preprocessing, encoding of features, scaling and dimensionality reduction via Principal Component Analysis (PCA).

The investigation is also confined to offline investigation based on previous network traffic data and not to real-time detection or implementation on live production networks. The measures of performance of the proposed hybrid model are evaluated based on the standard evaluation metrics such as accuracy, recall, precision, F1-score, homogeneity, completeness, and V-measure.

Additionally, the research focuses on the DDoS attacks of application-layer and neglects other types of cyber threats like malware, phishing, or network-layer denial-of-service attacks. Although the performance of the model is put against the performance of the selected traditional and machine learning-based intrusion detection methods, the study does not aim to make a comprehensive comparison between the model and the rest of the available detection methods.

It is within these confines that the study is expected to offer empirical evidence on the efficacy of hybrid forms of learning in enhancing detection accuracy in sophisticated high-dimensional application-level DDoS attack cases.

1.7 Justification

Application layer DDoS attacks are fast evolving and becoming increasingly severe. Machine-learning-inspired methods are particularly effective in creating detection and prediction models for such attacks (Ford et al., 2014). However, the fast-evolving nature of the IT environment presents the attackers with new techniques too. Therefore, the characteristics of DDoS attacks have evolved. There is a need to test them against new datasets and tune them to reflect the emergent changes. K-Means (She et al., 2016) and CART (Radoglou-Grammatikis & Sarigiannidis, 2019) algorithms have been individually used to create IDSs. Individually, the algorithms have had relatively commendable success. K-Means is an excellent algorithm in dealing with data with high dimensionality which is a common feature of network traffic. It is also highly capable of handling data with categorical features. On the other hand, CART algorithm does well with data with mixed datatypes. According to Xiao et al. (2019), supervised classification algorithms often work well in highlighting the links between sample attributes and their class labels. However, they neglect the potential structural attributes in the sample space. This downside can be alleviated by creating a hybrid of a supervised and unsupervised algorithm (Xiao et al., 2019). The proposed model leverages the strengths of K-Means (unsupervised algorithm) and CART (supervised algorithm) and seeks to reduce the impact of the respective weaknesses. Specifically, the model combines two none-resource intensive algorithms which reduces the overall computing resources required to detect network intrusion. The simplicity of the algorithms transfers to the model and gives the users the ability to easily interpret its data and processes making it easy to modify to suit

specific scenarios. Finally, the model can be applied to other domains which may have features similar to those in the current network traffic context.

1.8 Significance of the Study

Models based on old datasets have inherent limitations brought about by the ever-changing characteristics and features of network traffic behaviours. There is a need to re-test previous models with this new dataset. It has also become imperative to explore new techniques that work better with the current dataset. Therefore, a knowledge gap exists in this area.

K-means and CART are relatively established algorithms that have been employed widely in varying predictive models. With time, a much-used technique becomes well documented, and best practices in using them are established. By undertaking research using these well-tested techniques but applying them to new contexts, the current study contributes to this important work of exploring and documenting the relatively unknown efficiency of existing techniques in the contemporary network environment.

1.9 Limitations

It is important to recognize that despite the promise shown by the current approach, extant studies including Yaseen et al. (2015), Akkaya and Çolakoğlu (2019) and Fränti and Sieranoja, (2019) have shown that K-means and CART have inherent limitations that hinder their effectiveness. While the findings of this research contribute to the growing knowledge of machine learning approaches and network fraud prevention, some limitations call for further exploration and refinement of this model. The limitations are as follows;

Firstly, K-means and CART algorithms posted relatively low performance. Specifically, CICDDoS2019 dataset data groups are distributed in overlapping clusters that have irregular geometric shapes. K-means, according to Rodriguez et al. (2019), k-means performs dismally in cases where the dataset does not have convex distribution attributes. They also noted that it is not suited to datasets where clusters have significantly varied cluster sizes. Additionally, the CICDDoS2019 dataset is a vast collection of traffic from a simulated real-world network. Therefore, is highly complex. Such complexity and the irregular distribution and shapes of the clusters make k-means disadvantaged in initializing its cluster centres. It has a lot of outliers and noise. Rodriguez et al. (2019) noted that k-means is sensitive to noise. These deficiencies are bound to be transferred to any model, like the one under consideration in this study, that incorporates k-means. Even so, K-means considerably low requirements for computational resources, its long existences and researchers 'affinity to use and propose modification have resulted in walkarounds for several of its limitations.

1.10 Operational Definition of Terms

This section defines conceptual operational meanings of the main terms that are applied in this study. These definitions allow providing the consistency of terms in the study.

Intrusion Detection Model:

An intrusion detection model is system or piece of software that keeps track of network traffic and looks for any suspicious or malicious activity that can imperil the network's security or operation. When an intrusion detection model detects possible risks including

unauthorized access, malware infection, denial-of-service attacks, or data breaches, it may send out notifications or take action.

Application Layer DDoS attacks:

DDoS attacks on the application layer of the network protocol OSI model, which is in responsible for delivering services and interfaces for user applications, are a kind of cyberattack that target this layer. Application layer DDoS attacks make a lot of requests that seem genuine but are really malicious in order to overwhelm the resources of the targeted application or service, such as web servers, email servers, or online gaming platforms. DDoS attacks at the application layer are harder to identify and counter than attacks at lower layers because they need for more advanced analysis and filtering methods.

K-means:

This study defines K-means as an unsupervised learning algorithm that partitions datasets into a fix number of clusters such that the data points within clusters will be more similar to each other than to those in other clusters. To begin with, randomly selected cluster centroids are selected. Then, each data point is assigned to its nearest centroid to recalculate new centroids, which is the mean of all points in their cluster. This step of assigning and reassigning continues until the centroids no longer change or until a predefined number of iterations have been accomplished. K-means finds applications in data analysis, image segmentation, data compression, and anomaly detection.

Classification and Regression Trees (CART):

In this study, CART refers to decision tree algorithms, both for classification and regression. The algorithm splits the data into binary subsets by applying the feature value that optimizes the criterion of interest (Gini for impurity-based classifications or mean squared error for regression). Thus, the resultant tree structure makes decisions at internal nodes and issues predictions at the terminal leaves. CART can be used with numerical or categorical data, can handle missing values, and can also handle outliers. CART also serves as a basis for many ensemble techniques such as bagging, boosting, and random forests.

Hybrid Model (K-Means + CART):

Within this research, any hybrid model means a combination of two algorithms, K-Means and CART, which are set to increase the accuracy and efficiency in intrusion detection. It then uses K-Means for clustering network traffic between cluster groups of feature similarity. Each cluster is inspected by the CART classifier to build decision trees that classify individual instances as either normal or malicious. The hybrid framework leverages the benefits of clustering and classification, allowing for better pattern acquisition and lower data complexity during anomaly detection.

Python:

Python refers to a high-level programming language whose versatility in data science is highly acclaimed. It has a vast library of ecosystems supporting data processing, analysis, visualization, or any other tasks, including implementing machine-learning models. This

language was chosen to develop and test the intrusion detection system because of its easy readability and great community support.

Scikit-Learn:

Scikit-learn is a Python-based machine-learning library, and it was used in this study for the implementation and evaluation of algorithms like K-Means and CART. It provides utilities for data preprocessing, feature selection, model training, and performance evaluation. Under this research, Scikit-Learn is used for creating the hybrid model and for the assessment of the model's performance using different metrics such as accuracy, precision, recall, and F1-score.

Pandas:

A library that offers high-performance data structures and operations for working with tabular data is called **Pandas**. I read, cleaned, explored, and transformed the network traffic data into a machine learning-friendly format using pandas.

NumPy:

NumPy is a package that performs linear algebra operations on multidimensional arrays quickly and effectively. In order to manipulate and calculate the network traffic data and turn it into NumPy arrays for machine learning, I utilized NumPy.

Matplotlib:

A package that offers thorough plotting and visualization capabilities for making different styles of graphs and charts is called **matplotlib**. The network traffic data

distribution, the clustering findings, and the decision tree structure were all visualized using matplotlib.

DDoS Attacks:

In this study, DDoS attacks are considered the malicious attempt of disrupting the functioning of a website or online service by flooding it with excessive traffic coming from multiple sources. These attacks cause the organization in inefficiency, followed by loss in customer confidence, revenue, and notoriety. These attacks can also act as a smokescreen for more serious threats like data breaches or malware installation, which increases their threat in cybersecurity and network stability.

OSI Model – Application Layer:

The OSI model is a seven-layer framework for network communication. The application layer is the top layer and is therefore concerned with user-facing services and in defining the syntax, semantics, and structure of data exchanged between applications. Some protocols working at this layer include HTTP, FTP, SMTP, DNS, and SNMP. These protocols allow users to interact with network services. The focus of this research lies in the application layer because of its being susceptible to DDoS attacks. **Efficiency** is a metric for how quickly and effectively an intrusion detection model can handle network traffic data. Efficiency may be affected by a number of variables, including the volume and complexity of the data, the algorithm's speed and scalability, the system's hardware and software requirements, etc. Efficiency is crucial for intrusion detection because it may impact how quickly and accurately intrusions or network attacks are identified and

countered. An intrusion detection model that is more effective may handle more data with fewer resources and in less time.

Intrusion Detection:

The intrusion detection process is the monitoring and inspection of network traffic to identify unauthorized access, misuse, or malicious activity affecting network resources or data. Intrusion detection is a very important network security process in this study since its early detection of threats would lead to incident response and collection of forensic evidence.

Clustering:

The term clustering mainly involves grouping data points with similar characteristics. It is usually used for data reduction, visualization, or anomaly detection. In this study, clustering was performed on the network traffic data with features such as protocol, packet size, and IP address to know pattern structures in the data and reduce the data size.

Classification:

Classification is a sort of supervised learning that labels the data according to some characteristics. It finds its applications in areas such as fraud detection and spam filtering. Here, classification was applied to label network traffic as normal or abnormal, thereby possibly aiding in tracing threats or intrusions.

Accuracy:

Accuracy is one of the parameters and as such is used to evaluate the performance-based on how good the model classifies data points correctly. It is computed by finding the number of correct instances over the total number of instances. In this study, accuracy gives a measure of how well the actual intrusion detection model can separate normal and abnormal network traffic, with higher accuracy being synonymous with better performance.

1.11 Chapter Summary

The chapter outlined the setting and rationale of the study, by looking at the increasing danger of application-layer DDoS attacks and the weaknesses of current intrusion detection systems in dealing with the attacks. The study issue was clearly described with the gaps in the efficacy of the traditional and single-algorithm detection methods. The chapter established a well-organized basis of the study by defining the purpose, goals, and research questions that were to be used in the investigation. Other important concepts were also presented that pertain to intrusion detection, clustering, classification, and performance evaluation to put the technical focus of the research in perspective. Lastly, the study outline was given outlining the extent, importance and scope limitations of the research and providing a clear scope of the study. All these factors form the conceptual and contextual background of the literature review found in Chapter Two.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The growing complexity of computer networks and, indeed, the computing field has brought with it a host of challenges but not without an equally large variety of solutions. It can be said that computers were created to solve human problems, those that existed before computers. It has come to pass, however, that computers have brought with them novel challenges that exclusively face them. Cybercrimes never existed before computers but they are a big challenge today. The advent of machine learning offered a solution for complex human problems as well as emerging challenges associated with computers. Early machine-learning techniques had several limitations. For one, they were rigid and could not be made to take in varying data (Boutaba et al., 2018). The models made could only work with data bearing the exact identity of its training data. Progress has been made to alleviate this problem, partly because of the growth in the amount of digital data and due to the emergence of new algorithms.

Among the many challenges that face computers, DDoS attacks rank very high. The growth of the internet has precipitated a situation where anybody with access to the internet can conduct a pre-emptive strike on a network and bring it down with little computing power. An event that happened during the initial days of 2018 can illustrate this situation perfectly. Akamai is a tech company that is at the forefront of network security research and innovations. The company has a customer base from all over the world. A company of its stature and in its field of work is often thought to be secure and

able to repel network attacks. This was not the case, however, when they experienced a pre-emptive application layer network attack, or so they thought. A client from an overseas location was attracting huge network traffic. After several hours of analysis and initial confusion, the company experts discovered that a single program had a bug that autonomously launched massive amounts of requests to their client's network (Akamai, 2019). While malicious intent was absent, the damage was just as severe.

App-DDoS attacks are the most prevalent type of intrusion facing world networks today. According to Singhal et al. (2020), network-layer DDoS attacks were rampant in the past and provoked concerted efforts by researchers. Several solutions have been preferred to mitigate network layer attacks. Attackers have had to devise a new approach. The relative lack of studies into application layer network attacks has made it an obvious choice for attackers. Most of the current network intrusions are carried out using application-layer DDoS attacks (Singhal et al., 2020). Informed of this situation, new efforts have been made to bridge the gap. A growing number of studies have been done on the subject.

The chapter examines intrusion detection systems, application-layer DDoS attacks, and hybrid recognition systems as applicable to the objectives of this study.

2.2 Types of Intrusion Detection Models and Techniques

The act of systematically monitoring and analysing network traffic and system activities with the aim of identifying and preventing unauthorized, malicious, or unusual activities that can harm network resources or data is known as intrusion detection (Khraisat et al., 2019). Intrusion detection is important in network security due to the ability to detect and deal with possible threats since they are detected before they become a full-scale danger

and thus protect the confidentiality, integrity, and availability of information systems. The intrusion detection system (IDS) refers to a software-based or a hybrid software-hardware solution that helps identify the intent to attack a network by detecting abnormal activity or similarity of an observed activity with already known attack signatures (Yusof and Sulaiman, 2022). Next to the real time detection, IDS technologies give useful forensic pieces of evidence to aid in incident response and post incident investigation.

The literature suggests a broad array of intrusion detection methodologies and frameworks to counter the changing cyber threats which have also included application-layer distributed denial-of-service (DDoS) attacks. Application-layer DDoS attacks attack a specific layer on the top of the network protocol stack that provides the services and the interfaces to the end-user applications. A set of these attacks targets to cripple application resources, including web servers, email servers, or online gaming environments, by overwhelming them with high quantities of requests that appear to be a normal user response (Tripathi and Hubballi, 2021). Application-layer DDoS attacks are more difficult to detect because they are less noticeable but can also pose more challenges than lower protocol attacks, because more advanced mechanisms are often required to analyse and filter traffic.

Known intrusion detection methods may be categorised into numerous groups depending on their logic of detection and the principles of operation. These classes have been extensively researched and tested in various areas of application, datasets, or attacks.

Signature-based Intrusion Detection utilizes a predetermined database of known attack signatures or indicators of compromise (IOCs), to detect malicious operations (Yusof and Sulaiman, 2022; Malek et al., 2020). Compromise indicators can be malicious traffic

patterns, a specific sequence of bytes, file hashes or IP addresses of previous attacks. Signature-based IDTG works by referencing observed activity or traffic to system memory and sending off an alert when a match is found (Malek et al., 2020). This method works well in detecting known attacks having a comparatively small false positive. Although, it can only be considered to be limited by a failure to recognize novel or a zero-day attack that is not reflected in the signature database, which could result in frequent updates of signature to be effective (Alsoufi et al., 2021).

Anomaly-based Intrusion Detection is aimed at detecting abnormalities when compared to the normal baseline of normal traffic on the network or the operation of the system (Alsoufi et al., 2021; Yusof and Sulaiman, 2022). Many of them are performed by learning normal behaviour by means of statistical models or machine learning algorithms, and the incoming traffic is observed and compared with significant deviations. Anomaly-based IDS are especially useful in detecting new types of attacks that have never been witnessed before; such as zero-day exploits. Nevertheless, research has found that these systems are typically characterized by high false positive incidents, particularly in dynamic settings where the legitimate patterns of traffic change rather frequently (Hajj et al., 2021). Also, the detection based on anomalies needs large-scale amounts of data and computing capacities to build and operate robust base models.

Behaviour-based Intrusion Detection work on the basis of explicit rules or policies of acceptable network and system behaviour (Agate et al., 2022). These predefined rules are used to draw a comparison between activities observed and the rules to determine policy violations or unauthorized activities. Behavior-based IDS are useful in detecting both misuse attacks which are based on known vulnerabilities and ensuring the security

policies are being adhered to. They are, however, often susceptible to inaccurate manually crafted rules and cannot accurately reflect new attacks that do not contravene the existing rules (Agate et al., 2022).

Hybrid Intrusion Detection methods combine two or more detection methodologies, including signature-based, anomaly-based and behavior-based detection methodologies, to enhance the detection accuracy and strength (Smys et al., 2020). The hybrid IDS designs tend to implement the various mechanisms of detection in multiple layers of the network protocol stack, and might use dissimilar mechanisms on the various phases of the attack lifecycle, such as reconnaissance, exploitation, persistence, and exfiltration. The combination of various causes of evidence allows the hybrid systems to counteract the weakness of each strategy, which lowers the false positive and false negative results.

The efficiency of hybrid solutions has been proved in many different fields of application. An example is Chen et al. (2017) who used a hybrid model based on a combination of K-Means clustering and the J48decision tree algorithm to enhance the predictive performance in medical diagnosis. As a technique of data refinement, K-Means was applied to the data to find and eliminate any misclassified samples before classification. Even though it is not a study about cybersecurity, this work demonstrates how the clustering concept is worth being used as a preprocessing process to improve the working of a classifier. On the same note, Torabi et al. (2018) developed a hybrid approach of clustering and classification short-term energy consumption prediction, and showed that forecasting similar data cases before making a classification can be beneficial to identify complex patterns and increase the predictive ability.

Intrusion and DDoS detection have been subjected to clustering in the cybersecurity domain. Lee et al. (2008) developed a DDoS detection technique that relied on cluster analysis with regard to entropy values of the traffic, proving beneficial in early detection of various stages of the attack considering the anomaly detecting process. Firdaus et al. (2020) also proved that hybrid detection can significantly enhance the efficiency of DDoS detection in Software Defined Networks (SDN), which is the strength of ensemble learning models in contemporary network settings. On the same note, Yzzogh and Benaboud (2025) have suggested a hybrid DDoS detection model that combines various K-Means algorithms with a Naive Bayes classifier that demonstrates enhanced results of detecting DDoS attacks in SDNs, attributable to the feature-wise clustering.

Semi-supervised and decision tree-based hybrid models used in intrusion detection have also been the subject of other studies. Aamir and Zaidi (2019) have explored clustering-based semi-supervised machine learning on the categorization of DDoS attacks showing a better performance of detection among the use of labelled and unlabelled data. Aung and Min (2018) suggested a hybrid intrusion detection system consisting of K-Means clustering and Classification and Regression Trees (CART). Their work is not specifically on application-layer DDoS attacks but it forms a relevant basis on how to achieve integration between clustering and decision tree-based classification in intrusion detection systems.

Along these investigations, the present research concentrates on application-layer DDoS attacks, that has some different peculiarities with network-layer attacks. The proposed hybrid allows one to improve the accuracy and efficiency of detection to application-layer DDoS scenarios, by combining the K-Means clustering algorithm and the CART

algorithm. It is tested against the actual datasets and suitable metrics in order to determine the effectiveness of the proposed model in catching application-layer DDoS attacks.

The effectiveness of intrusion detection models is typically measured by a number of quantitative measures, such as detection rate, false positive rate, false negative rate, accuracy, precision, recall, F1-score, response time, and resource consumption (Thakkar and Lohiya, 2021; Salih and Abdulazeez, 2021; Abushark et al., 2022; IBM, n.d.). These measures give a holistic foundation of measuring the detection ability and the efficiency of operations.

2.3 Application Layer DDoS Attacks

Application-layer DDoS attacks occur on top of the OSI network model and do so by providing service and interface directions to support end-user applications. The attacks are carried out to drain the computational resources of servers, such as CPU, memory, disk access, or input and output (I/O) processes of the database, by generating massive numbers of requests purporting to be innocent when in actuality they are malicious (Tripathi & Hubballi, 2021). The aim is to reduce system performance, to slow down response time, or deny the capability of access by legitimate users to the particular resource. Also known as layer 7 attacks, these DDoS threats are the hardest to mitigate and detect as they simulate legitimate user behaviour in the first place, hence needing a robust detection mechanism.

Application layer DDoS attacks may be categorized into many forms according to the techniques and protocols used to produce and transmit the requests. There are many prevalent forms of application layer DDoS attacks:

An HTTP flood attack is characterized by the deliberate flood of a web server with a substantial volume of HTTP requests. This attack strategy often employs numerous connections and uses randomized parameters to circumvent cache systems, hence intensifying the burden on the server (Tripathi & Hubballi, 2021). HTTP flood attacks may be categorized into two types: GET flood and POST flood, based on the specific HTTP method used in the queries.

A DNS flood attack is characterized by the deliberate swamping of a DNS server with a substantial volume of DNS requests (Tripathi & Hubballi, 2021). This attack strategy often employs falsified IP addresses and arbitrary subdomains to impede cache mechanisms and intensify the burden on the targeted server. DNS flood attacks can target either recursive DNS servers or authoritative DNS servers, contingent upon the specific level within the DNS hierarchy.

The SMTP flood attack, on the other hand, involves the transmission of substantial volumes of Simple Mail Transfer Protocol (SMTP) requests to an email server (Tripathi & Hubballi, 2021). This attack strategy often employs several connections and uses random parameters to circumvent filtering methods and intensify the server's workload. SMTP flood attacks can selectively target certain email accounts or domains, contingent upon the hierarchical position inside the email system.

The application layer is a crucial component of the network architecture, responsible for facilitating communication between end-user applications and the underlying network

infrastructure. DDoS attacks provide considerable obstacles in the realm of network security and vulnerability management. Several issues that arise include:

The detection of Application layer DDoS attacks poses challenges due to their use of genuine protocols and requests that imitate typical user behavior. Additionally, they use low bandwidth and high-frequency approaches that can circumvent conventional network-based detection methods, which heavily depend on analysing traffic volume or packet examination.

The mitigation of application layer DDoS attacks poses challenges due to the need for advanced analysis and filtering systems capable of discerning between genuine and malicious requests. Additionally, there is a need for more precise and adaptable countermeasures that may safeguard certain apps or services while minimizing any impact on other network functionalities.

Application layer DDoS attacks pose challenges due to the use of diverse strategies aimed at concealing or falsifying their source IP addresses. These techniques include using proxy servers, VPN services, and botnets. In addition, perpetrators use many methodologies to obscure or encrypt their payload or parameters, including compression, encoding, or encryption.

In recent years, there has been a noticeable rise in the occurrence, intensity, and intricacy of DDoS attacks targeting the application layer. An analysis conducted by NETSCOUT reveals that in the year 2021, application layer DDoS attacks constituted a significant majority, amounting to 59% of all DDoS attacks (Azure Network Security Team, 2023). The survey also disclosed that there was a 5% increase in application layer DDoS attacks

in 2021 as compared to the year 2020. Several variables have contributed to this observed tendency including;

Nefarious entities possess a lot of tools, equipment and knowledge that facilitate their capacity to execute application-layer DDoS attacks more effectively. These entities include internet-based platforms that supply stresser or booter services (DDoS-for-Hire). These open-source code bases provide DDoS attack scripts or modules, and infected machines that constitute botnets or zombies (Azure Network Security Team, 2023).

Cyber crooks display a diversity of goals and motivations in perpetrating application-layer DDoS attacks. These include financial gain via extortion or ransomware, political goals through hacktivism or cyberwarfare, economic advantage through sabotage or disruption, personal vengeance through retribution or trolling, or just curiosity or experimentation (Azure Network Security Team, 2023)

The tools and tactics used by malicious actors for initiating application layer DDoS attacks have undergone evolutionary changes. Various strategies can be employed to carry out DDoS attacks, such as the utilization of multi-vector or blended attacks that incorporate diverse forms of application-layer DDoS attacks or lower-layer DDoS attacks. Additionally, attackers may exploit zero-day vulnerabilities or exploits that have not yet been addressed or made known by vendors or developers. Furthermore, artificial intelligence or machine learning algorithms can be leveraged to automate or enhance the efficiency of the attack procedure (Azure Network Security Team, 2023).

2.4 Existing Models and Datasets

Singhal et al. (2020) proposed a log analysis system that uses big data technology. They trained their model using the CLARKNET dataset (163MBs containing 1.2 million packets). The researcher first fed the data into R for cleaning and then fed the resultant data (153MB) into APACHE PIG for analysis. The analysis tried to highlight normal IP addresses perpetrating DoS or DDoS attacks. Three different criteria were used to analyse the data; the frequency of hits of each IP on a given date, an IP's number of requests for a particular resource, and the frequency of hits for a resource at a particular date. The resultant file was loaded back into R. This third and final phase aimed to refine the data by removing redundant indentations and tokenizing the data. The final results were then saved on a file that would be sent to the personnel charged with taking action against such attacks.

Several models and datasets have been employed to train and test models to analyse DDoS attacks. Kumar et al. (2022) noted that the most popular models include Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF), Nearest Neighbour (KNN), Naïve Byes (NB), and Neural Networks (NN). They also mentioned some of the common datasets that have been used in the last decade including CAIDA UCSD, KDD Cups '99, CICIDS2017, internally collected data from IoT devices, data sourced from (SDNTrafficsDS) KDDCups'99 and CICDDoS2019.

2.5 K-Means Algorithm and Related Models

While there has never been a perfect model for application-layer DDoS attack detection, machine-learning techniques have proved very useful. The K-Means algorithm has been particularly applied widely in data mining and in particular to sort the massive network intrusion data. She et al. (2016) investigated the applicability of K-Means in detecting application layer network attacks. They, like Singhal et al. (2020) noted that little research has been done on the methods that can be used to detect application-layer DDoS. The study modelled the real features of legitimate clients of a network. She et al. (2016) suggested that there exist several disparities between a normal user and a DDoS attack 'user'. A real user can be identified by the consistency of their session's features (She et al., 2016). They have steady and small volume request rates and mostly target a familiar resource. Building the user behavior model would help the researchers to contrast and detect fraudulent network traffic.

Many times, an excellent method is ruined because of poor application. K-Means is an excellent clustering algorithm and works excellently in detecting application-layer DDoS attacks (She et al., 2016). Its effectiveness can, however, be greatly diminished by unbalanced cluster sizes and poor choice of clustering method (Fränti & Sieranoja, 2019). Fränti and Sieranoja (2019) pointed out that while K-Means have some limitations, they can be overcome by properly applying them. They noted that this is possible because the algorithm has been studied widely and best application practices have been suggested to neutralize its limitations. In their study, they suggested that the K-Means algorithm's clustering accuracy can be greatly improved by a better initialization approach and iterations of the process. They pointed out that K-Means has problems in optimizing the

initial centroids globally once initiated especially when the distance between the clusters is significant. Therefore, precision during initialization can greatly improve the algorithm's performance.

Fränti and Sieranoja (2019) sought to add their research to the growing body of studies on the K-Means algorithm. They noted that the algorithm is the basis of the emerging crop of genetic algorithms such as the particle swarm optimization and was still preferred by many researchers. Their study utilized the Clustering Basic Benchmark dataset, which the K-Means is known to cluster correctly. Subsequently, they eliminated the flaw that would have been introduced by poor clustering function selection. They subjected the data to different initializations and varying iterations. They concluded that proper initialization improved the chances of accurate clustering of data and that subsequent iterations of the algorithm refined the alignment of the clusters further. The better the initialization, the fewer iterations are needed to achieve the accuracy required (Fränti & Sieranoja, 2019).

Refining the application and precision of the K-Means algorithm is not the only way to improve its performance. Jia et al. (2017) noted that while K-Means was very capable of handling intrusion detection, its results could be greatly improved by coupling it with another algorithm. The study explored the performance of a hybrid of K-Means and C4.5 algorithms. Jia et al. chose to use the KDD Cup 1999 dataset. It used stand-alone models featuring the two algorithms. The K-Means model clustered the data. The aim at this level was to identify meaningful structures and patterns. The C4.5 model builds a decision tree for every cluster generated by the K-Means algorithm. Jia et al. (2017)

noted that a system combining the K-Means algorithm and another, specifically a decision tree algorithm, improved the accuracy of intrusion detection.

2.6 CART Algorithm and Models Based on It

C4.5 is a very capable regression tree algorithm but it has some limitations. The algorithm does not support boosting to optimize the performance, and cannot work if the data contains some missing values. The CART algorithm overcomes these challenges of C4.5. CART according to Krzywinski & Altman (2017), is a powerful prediction algorithm. When data is fed into a CART model, it splits it along predictor axes effectively separating the dissimilar groups. The target is to have homogenous groups of data. The algorithm uses a binary tree. It considers features and thresholds that can lead to the most information gain at each node. It can use the GINI diversity index (GI), Information Gain (IG), the twoing criteria, and Cost Complexity Pruning (CCP) to split the data at each successive node and to make a decision on which branch contains the sought-after group.

CART was employed successfully by Radoglou-Grammatikis and Sarigiannidis (2019) in a smart grid intrusion detection system. Their research noted that with many critical systems depending on ICT infrastructure, attacks could be catastrophic. Therefore, it is important to develop intrusion detection systems that would detect such intrusions in real time and implement a solution just as fast. CART, like its regression tree relatives, is good at predicting unknown attacks, an advantage that is seriously needed in the fight against fast-evolving network attacks (Radoglou-Grammatikis & Sarigiannidis, 2019). The research chose to use the IG approach to identify the features that differentiated the groups of data or identified them as being similar.

Radoglou-Grammatikis and Sarigiannidis (2019) proposed a modular IDS. Their model consisted of four modules; network monitoring, network flows extraction, an analysis engine, and response modules respectively. The first module monitors the network by scanning all incoming and outgoing traffic's Transmission Control Protocol/Internet Protocol (TCP/IP) (Radoglou-Grammatikis & Sarigiannidis, 2019). The second module takes the data from the first one and extracts bidirectional network flows. To properly select the network flows of interest, the researchers characterized them by their duration, packet length, packet size, and the number of packets sent in the initial window. The third module takes in data from the preceding step and passes them through a decision tree model based on CART to identify attack traffic from legitimate ones.

Radoglou-Grammatikis and Sarigiannidis (2019) employed the CICDDoS2017 dataset, which is a precursor of the CICDDoS2019 dataset. The dataset was chosen due to its ability to be characterized by the researchers' selected features. Radoglou-Grammatikis and Sarigiannidis (2019) also noted that the dataset contained network flows synonymous with DoS and DDoS attacks apart from other frequent network attacks. They trained the model with the dataset and deployed the testing process over the sci-kit learn library. In the final module, the results of the data passed through the algorithm would then be passed to the network administrator for decision and action. Radoglou-Grammatikis and Sarigiannidis's (2019) decision tree model employing CART had an accuracy of 0.9966 and a True Positive Rate of 0.9930.

K-means and CART are two algorithms that have been widely studied and applied individually to detect network intrusions. The strengths of both algorithms can be

harnessed in a hybrid model that also aided in minimizing or cancelling out the weaknesses of the individual algorithms.

2.7 CICDDoS2019 Dataset

Cognizant of the increasing frequency and potency of Distributed Denial of Service (DDoS) attacks, cyber security agencies are always looking to stay ahead of the curve. One of the persisting limitations of the current strategies against cyber-attacks has been the dependence on old datasets to train models expected to protect against more novel attacks. Sharafaldin et al. (2019) analysed the existing datasets intending to make improvements. Ultimately, they generated a dataset that considered important features present in current types of DDoS attacks. The new dataset best describes more recent types of attacks such as application layer DDoS attacks carried out using TCP/UDP-based protocols (Canadian Institute of Cybersecurity, 2019). This section describes the new taxonomies and how it informs the creation of this dataset.

2.7.1 Reflection-Based DDoS

These are attacks where the perpetrator stays hidden courtesy of legitimate third-party components (Canadian Institute of Cybersecurity, 2019) Such an attacker sends packets to reflector servers. They set the source IP to a victim's IP address so that when the server responds, the responses are sent to the victim. By increasing the volume of the requests, an attacker could cripple the victim system by flooding it with unsolicited responses. These attack types are carried out through application layer protocols via transport layer protocols such as Transmission Control Protocol (e.g., MSSQL and SSDP) and User

Datagram Protocol (e.g., CharGen, NTP, and TFTP). Some attacks like DNS, LDAP, NETBIOS, and SNMP can be carried out using either of the protocols.

2.7.2 Exploitation-Based Attacks

In exploitation-based attacks, an attacker behind legitimate third-party components sends packets to reflector servers with the source IP set to a victim's IP address. The server duly floods the victim with responses. Alternatively, an attacker could employ application layer protocols via the transport layer. The attacks can be TCP-based like SYN flood or UDP-based such as UDP-Lag and UDP flood (initiated by a remote host by sending vast numbers of UDP packets) (Canadian Institute of Cybersecurity, 2019).

UDP flood involves the perpetrator sending UDP packets to random ports in the victim's system at very high rates such that it overwhelms the existing bandwidth and crashes it. SYN flood involves an attacker sending recurring SYN packets to a victim system (Sharafaldin et al., 2019). It takes advantage of the TCP three-way handshake. UDP-Lag attacks involve the use of a hardware switch called a lag switch or a software program running on the network to monopolize the bandwidths of other users.

Figure 2.1 shows the classification of the DDOS attacks into these two categories and also into the subsequent divisions based on their characteristics.

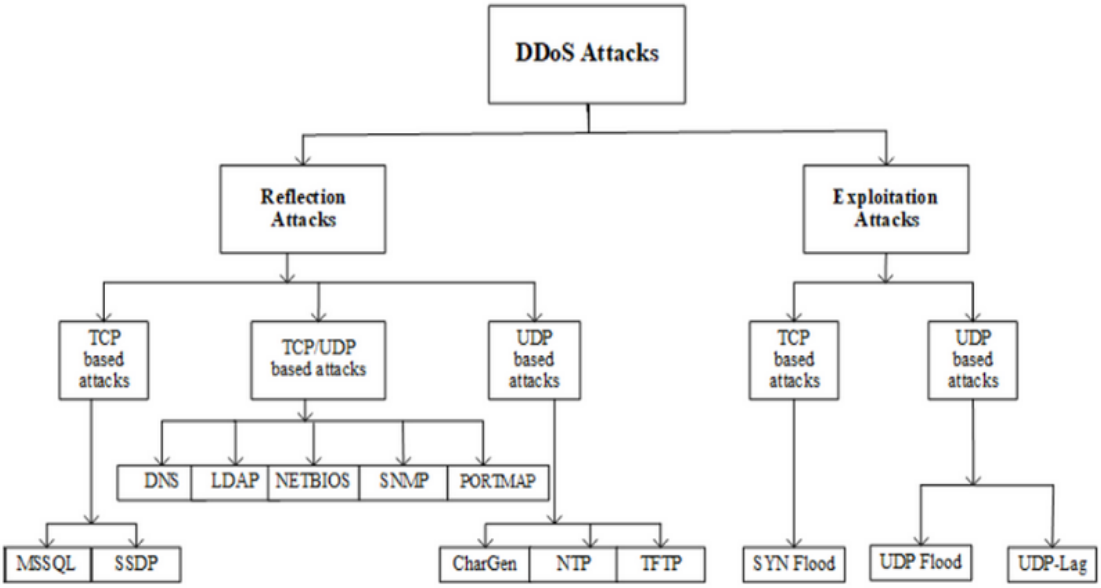


Figure 2.1 DDoS Attacks Taxonomy (Canadian Institute of Cybersecurity, 2019)

The dataset captures the most recent and common DDoS attacks reflecting real-world data. The attack features were extracted using CICFlowMeter-V3. The results show flows based on time stamps, protocols, attacks, and source and destination ports and IPs. The dataset authors wished to create a realistic dataset and tried to replicate the real-world conditions through their setup as shown in Figure 2.2. They profiled abstract human behavior and interactions in a naturalistic and benign network.

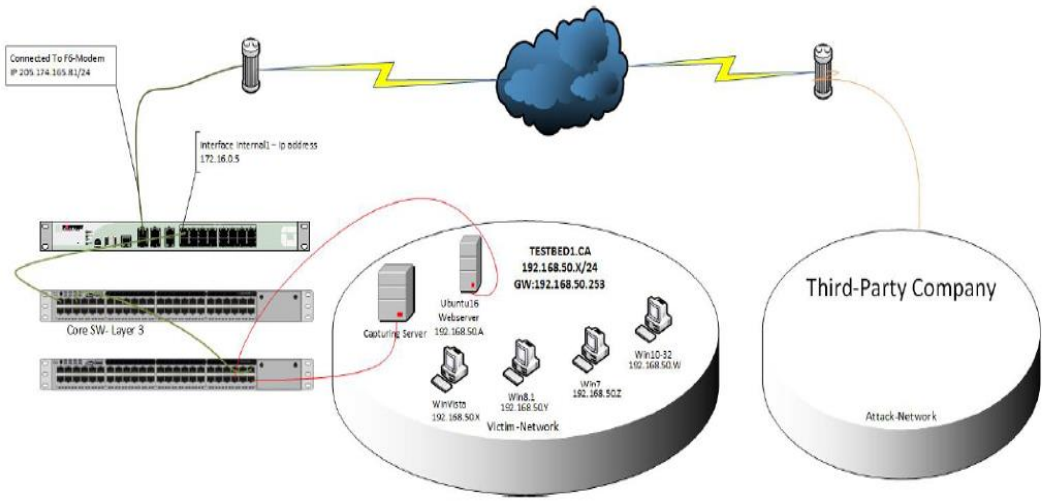


Figure 2.2: CICDDoS2019 Testbed Architecture (Sharafaldin et al., 2019)

Then, the authors built abstract behaviours of 25 users based on the HTTP, HTTPS, FTP, SSH, and email protocols as shown in Figure 2.3.

Machine	OS	IPs
Server	Ubuntu 16.04 (Web Server)	192.168.50.1 (first day)
		192.168.50.4 (second day)
Firewall	Fortinet	205.174.165.81
PCs (first day)	Win 7	192.168.50.8
	Win Vista	192.168.50.5
	Win 8.1	192.168.50.6
	Win 10	192.168.50.7
PCs (second day)	Win 7	192.168.50.9
	Win Vista	192.168.50.6
	Win 8.1	192.168.50.7
	Win 10	192.168.50.8

Figure 2.3: Abstract Behavior of 25 Users (Sharafaldin et al., 2019)

The authors proceeded to launch different reflective DDoS attacks including PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP. They executed 12 DDoS types of attacks including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP during the training day. On the testing day, they launched seven attacks including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN as shown in Figure 2.4.

Days	Attacks	Attack Time
First Day	PortMap	9:43 - 9:51
	NetBIOS	10:00 - 10:09
	LDAP	10:21 - 10:30
	MSSQL	10:33 - 10:42
	UDP	10:53 - 11:03
	UDP-Lag	11:14 - 11:24
	SYN	11:28 - 17:35
Second Day	NTP	10:35 - 10:45
	DNS	10:52 - 11:05
	LDAP	11:22 - 11:32
	MSSQL	11:36 - 11:45
	NetBIOS	11:50 - 12:00
	SNMP	12:12 - 12:23
	SSDP	12:27 - 12:37
	UDP	12:45 - 13:09
	UDP-Lag	13:11 - 13:15
	WebDDoS	13:18 - 13:29
	SYN	13:29 - 13:34
	TFTP	13:35 - 17:15

Figure 2.4: Training and Testing (Sharafaldin et al., 2019)

The dataset is organized by day. Each day captures the record of raw data including traffic (Pcaps) and event logs (Windows and Ubuntu) per machine. The dataset contains more than 80 features of the traffic extracted by the authors. Therefore, there are two versions of the dataset. The raw data (PCAP) and the one where the features have been extracted.

2.8 Theoretical Framework

Varpio et al. (2019) defined a theory within objectivist deductive and subjectivist inductive research as rationally related concepts and arguments. Scholars and subject experts often offer explanations about phenomena. Such proposed explanations draw in abstract explanations of the concepts that ultimately offer insight into something. The theory is founded on compelling preliminary evidence or a vast body of studies dealing with the same subject. Varpio et al. added that theories can be about denoting, portraying, and distinguishing something and its features (descriptive), expounding on the links between phenomena (explanatory), highlighting and speaking about the subjugation of a population (emancipatory), improving or refuting existing concepts and propositions about a phenomenon (disruptive), or forecasting the outcome of a process given certain input (predictive).

2.8.1 Machine Learning Theories and Network Security Theories in DDoS Detection

Machine learning theory, or as is sometimes referred to, computational learning theory, gave us an understanding of how algorithms learn from data, whether in supervised or unsupervised environments (Zhou, 2021). This research considered these two subfields of the theory. Evolutionary theory from network security, meanwhile, informed the study through the addressing of how dynamic and adaptive modern cyber threats including application-layer DDoS attacks are.

2.8.1.1 Supervised Learning Theory

Supervised learning refers to a form of machine learning wherein the algorithms are trained with labelled data to perform either a predictive or a classification task (Sharma,

Sharma & Khanna, 2020). Here, training is performed on pairs of input and output, such that the model can generalise to output predictions on unlabelled data. According to Sharma, Sharma and Khanna (2020), key issues in the theory of supervised learning are as follows:

Statistical Frameworks: Statistical models are used to map inputs to outputs in supervised learning, while minimizing an error function of some form, most typically the mean square error in regression problems and cross-entropy in classification problems.

Algorithmic Analysis: The ability of these algorithms to generalise to new data, which is frequently assessed under common metrics of accuracy, precision, recall, and F1 score, is one of the ways to evaluate their performance.

DDoS Detection Application: In this study, supervised learning is represented using the Classification and Regression Tree (CART) algorithm, which builds decision trees to classify network traffic as either benign or malicious from labelled data from the CICDDoS2019 dataset. Its interpretability and the fact that it can handle both categorical and continuous data make it a candidate method for detecting such sophisticated attack patterns.

2.8.1.2 Unsupervised Learning Theory

Unsupervised learning focuses on studying patterns and structures in unlabelled data (James et al., 2023). Unlike supervised learning, this method does not depend on externally defined labels to associate data; rather, it clusters data based on their inherent similarities.

Sharma, Sharma and Khanna (2020) outline the following major concepts of unsupervised learning theory:

Clustering and Dimensionality Reduction: Unsupervised learning algorithms such as K-Means grouped data into clusters based on feature similarity. This helped identify some anomalies or outliers in network traffic.

Algorithm Analysis: The performance evaluation of unsupervised learning algorithms is made on metrics represented by silhouette score, Davies-Bouldin index, or within-cluster sum of squares (WCSS).

DDoS Detection Application: In this research, a K-Means clustering algorithm clustered network traffic depending on behaviour patterns. Hence, in an unsupervised way, it was be able to unravel anomalies suggesting DDoS attacks.

2.8.1.3 Evolutionary Theory in Network Security

The evolutionary theory in network security is derived from the Darwinian evolution concepts. It emphasises the dynamic and adaptive nature of all cyber threats (Yerriswamy & Murtugudde, 2020). As an attacker evolves with new tactics and techniques, defence mechanisms must also evolve in order to be effective. Important aspects of evolutionary theory as noted by Yerriswamy and Murtugudde (2020) are:

Adaptive Threat Landscape: Application-layer DDoS attacks are constantly evolving, with attackers employing new techniques to bypass traditional IDSs. This necessitates adaptive and intelligent detection mechanisms.

Feedback for System Improvement: Evolutionary theory stresses the feedback loop that needs to be taken into account for the improvement of any security system. Such systems

can learn and adapt to emerging threats by modifying their current models through the analysis of attacks that have already been detected.

DDoS Detection Application: This research uses evolutionary theory by proposing a hybrid model, which combines supervised learning (CART) with unsupervised learning (K-Means). The hybrid model uses the unsupervised approach to cluster traffic and identify behavioural patterns, while the supervised approach classifies the attacks within the clusters. The model's learning capacity is thus maintained through this feedback cycle, ensuring its continuous effectiveness against DDoS threats under evolutionary pressure.

2.8.1.4 Computational Learning Theory

Computational Learning Theory provided a theoretical foundation for analysing the performance of machine learning algorithms (Zhou, 2021). This study borrowed some concepts from this theory including:

Sample Complexity: A question most central to this theory relates to the quantity of data that is required for the model to learn well.

Computational Complexity: How quickly can you prepare, train, and set up a model?

Generalization: How does your model reply when given new and unseen examples?

The computational learning theory provided information to analyse K-Means and CART individually and together. In particular, the Probably Approximately Correct (PAC) learning framework helped to give a mathematical foundation for almost any hybridization-concept-based model whereby the generalization performances of the model were evaluated (Zhou, 2021). As such, the research made sure that the

generalization error of the model is low with high probability in order to derive a solid and trustworthy DDoS detection and prevention mechanism.

2.8.1.5 Unification of Theories in the Hybrid Model

The hybrid model combined essential theories of supervised and unsupervised learning to take account of problems presented by traditional techniques for DDoS detection and prevention. More specifically,

K-Means Clustering: Patterns and anomalies in network traffic are found, allowing for inspection.

CART Algorithm: Classifies traffic from these patterns into benign or malicious behaviour.

Adaptive Evolution: The model adapts to the new attack patterns encountered.

The integration of the three different theories gave a hybrid model that balanced accuracy, interpretability, and robustness in DDoS attack detection.

2.8.1.6 Important Contributions of the Theoretical Framework

Supervised Learning Theory: Provided the classification framework for the usage of CART, thus ensuring accurate detection of known attack patterns;

Unsupervised Learning Theory: Provided K-Means-Based Clustering for identifying possibly unknown attack patterns.

Evolutionary Theory: Guaranteed the model was always flexible to the dynamic nature of DDoS attacks.

Computational Learning Theory: Assisted in ensuring the model gave low generalization error with respect to new threats, being defined as DDoS attacks.

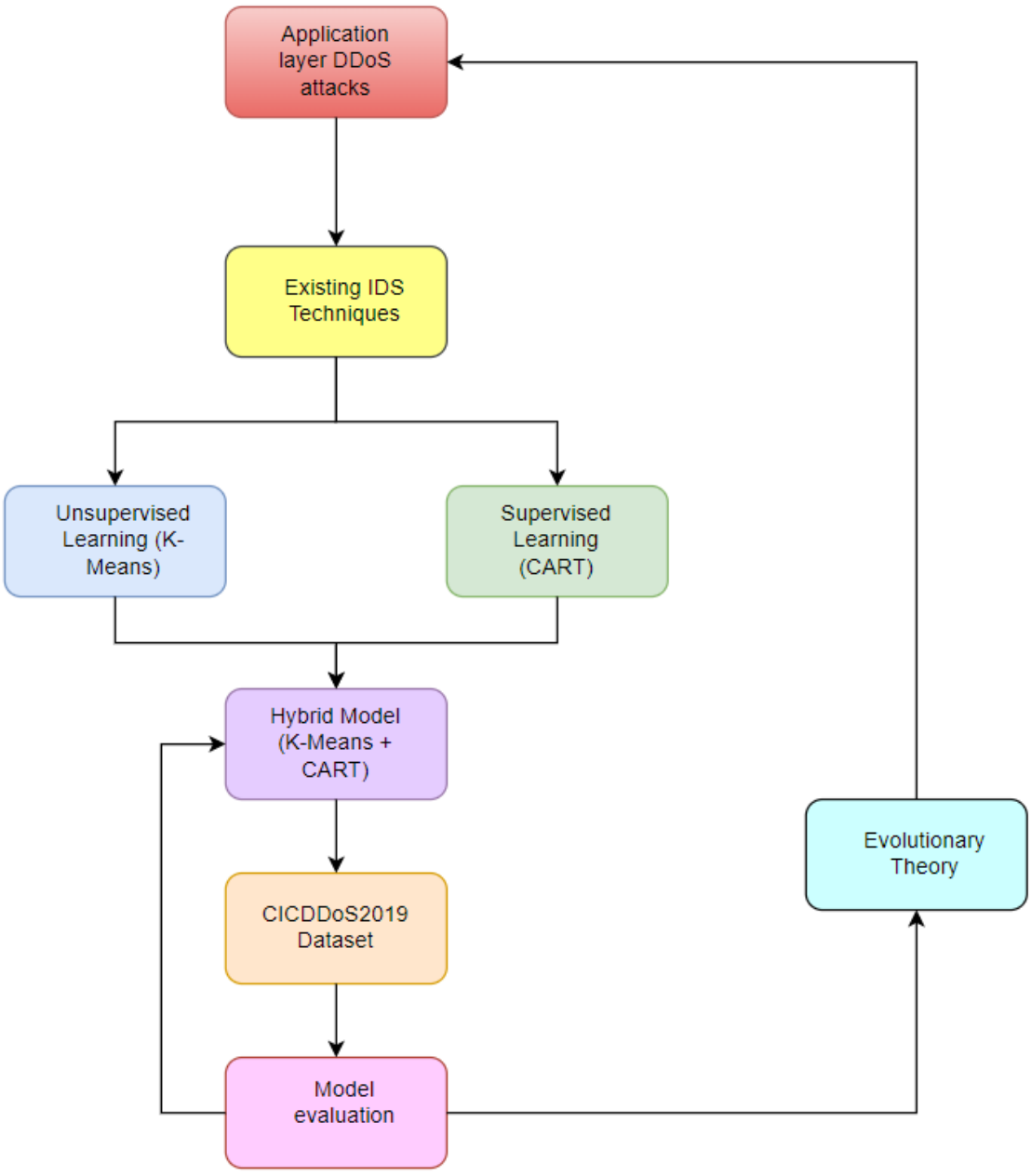


Figure 2.1: Computational Learning Theory Model

The unified theoretical framework drew upon supervised and unsupervised learning theories as well as evolutionary theory to analyse the problem of the detection of application-layer DDoS attacks. The hybrid model hereby gained strength from K-Means and CART, providing a robust, adaptive, and interpretable solution for current network security. The framework is still further corroborated by Computational Learning Theory, which carefully guaranteed that model performance and generalization capabilities were evaluated.

2.9 Conceptual Framework

Security and reliability of modern networks are highly related with efficiently detecting and thwarting application-layer DDoS (Distributed Denial of Service) attacks. With this definition, because of the dynamic nature of these types of attacks and the inability to detect them using traditional intrusion detection systems, a novel hybrid intrusion detection model was necessary.

2.9.1 Main Concepts

Application Layer DDoS Attacks: The focus of this study is on understanding application layer DDoS attack: their characteristics, evolutionary concepts in form, impact on network resources, and services.

Intrusion Detection Systems (IDS): The work also views some of the previous works on different IDS techniques signature-based, anomaly-based, and machine learning-based, and discusses their strengths and weaknesses in those aspects that may be applicable for dubbed purposes for detecting application layer DDoS attacks.

Machine Learning Algorithms: To that end, this research also focuses on extracting two most prominent machine learning algorithms from the literature:

K-Means Clustering: The unsupervised learning-based K-Means clustering is a method used to group specific clusters in a traffic data into similar pools by virtue of similarities in feature by K-Means which works in high-dimensional data to identify hidden patterns and anomalies in network traffic.

Classification and Regression Trees (CART): Another algorithm is called supervised that uses a technique called decision trees for classification. CART can be applied both on continuous and categorical data and can model quite complex relations between features and labels of attack.

Hybrid Model: In this study, we introduce a new hybrid model, K-Means clustering with the CART algorithm. The K-Means algorithm uses processing data that identifies clusters of similar network traffic behaviour. From this step, the output of the clustering is given as input to the CART algorithm for the classification process, thus improving both accuracy and efficiency for attack detection.

CICDDoS2019 Dataset: This research investigation was possible because of the CICDDoS2019 dataset. The dataset was a collection of quite new and complete characteristics of today's application layer DDoS attacks. Using this dataset ensured that the defined model was tested against real-time attacks.

Model Evaluation: Proposed hybrid model performance evaluation was performed using various metrics like accuracy, precision, recall, and F1 score. This evaluation was done to

check how well the model was in detecting attacks coming through the application layer while minimizing false positives and negatives.

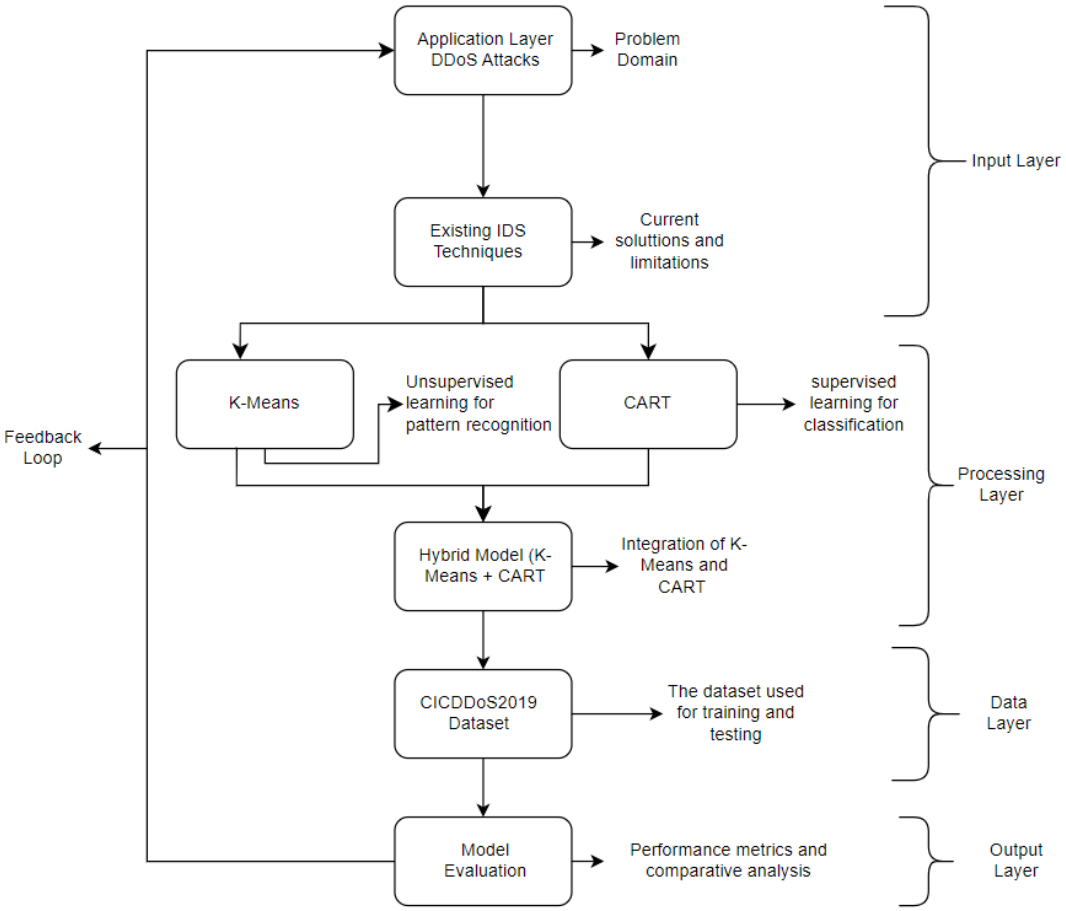


Figure 2.1: Conceptual Framework Diagram

2.10 Existing Technical Gap

2.10.1 Dynamic Nature of Application-Layer DDoS Attacks

While it is true that application-layer DDoS attacks have been one of the most impactful forms of DDoS attack in the present IT world, their ability to go unnoticed by the detection system by using new techniques is one of the most significant technical gaps in

current IDSs. Therefore, even the best application by signature-based detection or anomaly-based detection could not be of any help now in the fight against these kinds of attacks since they are designed to counter almost all of the common forms by which an attacker has previously attacked a site. For example, if a certain attacker has used the pattern of a known attack in attacking their site, and has thus shown the pattern to their signature-based IDS relying on the development of known attack signatures, that attack could never have gotten into network. The same is the cause of why anomaly-based detections fail. Though they can detect other unknown attacks, still their detection has many false positives because they depend on a static setting. This fact has therefore made a call for the adaptive intelligent detection mechanisms that could possibly run in tandem with the changing landscape of threats.

2.10.2 Lack of Hybrid Approaches

This approach resonates with another major gap, which examined hybrid models of supervised and unsupervised learning algorithms while dealing with DDoS detection. Though separate studies have been undertaken on algorithms like K-Means (unsupervised) and CART (supervised), their combination has been among unexplored fronts. Most existing works have proposed individual models, that is, standalone models, and such an approach may fail to fully tap the strength of both paradigms. One such advantage is that the unsupervised learning model is able to detect new attack patterns, while the supervised learning model can be applied to classify known threats with high accuracy. This unification may provide a better solution yet it lacks thorough research in testing and developing, especially on application-layer DDoS attacks. This gap further

brings the necessity of hybrid novel frameworks to improve detection accuracy and adaptiveness.

2.11 Chapter Summary

The chapter critically analysed the literature available regarding intrusion detection systems with particular interest in application-layer DDoS attacks detection. It analysed classical signature-based, anomaly-based and hybrid techniques of detection, and their advantages and weaknesses, in managing high-dimensional and changing attack patterns. The chapter further considered machine learning techniques that apply to intrusion detection such as clustering and classification algorithms and evaluated how they can be applied to complex network traffic data. Moreover, the CICDDoS2019 dataset was also considered to support the appropriateness of the dataset in assessing application-layer attacks. The literature review led to the identification of a scientific gap in the literature on the successful implementation of unsupervised and supervised learning to achieve a higher level of detection. These understanding guided the methodological decisions to be taken in the research with results of Chapter Three.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

Research methodology refers to the comprehensive process that the study took as it seeks to achieve or fulfil its objectives (Oso & Onen, 2008). This section incorporated the research design adopted in this research, the population of the study, and sampling techniques. It also detailed the dataset that was used and the data analysis techniques and tools employed.

3.2 Research Philosophy

The empirical positivist paradigm is the research philosophy that was employed in this study and views knowledge to be based upon observable and quantifiable phenomena using scientific methods (Creswell and Creswell, 2018). Positivism focuses more on objectivity, replicability and application of empirical evidence in testing theories, where the view of the researcher is irrelevant (Creswell and Creswell, 2018; Saunders et al., 2023). In this regard, the paradigm is consistent with machine learning studies, where algorithms were trained and tested on measurable information in order to make generalizable conclusions.

The reason this philosophy was selected is that it is the most appropriate philosophy used in the study since it is focused on creating and testing a hybrid intrusion detection model, applying structured and numerical data to the CICDDoS2019 dataset. Positivism also permitted the systematic experimentation to confirm the model performance indicators (such as accuracy and precision), guaranteeing the results to be verifiable and not

subjective (Alturki, 2021). In contrast to interpretivist methods which emphasize subjective interpretations, positivism was favourable to the objective assessment of network traffic patterns and algorithm effectiveness, where empirical validation is the key to cybersecurity applications. An example is that it facilitated testing of hypothesis on whether the hybrid model will work better than the individual K-Means and CART algorithm, which fostered scientific rigor and possible practicality.

The theoretical basis was based on machine learning theory that considered learning as optimization of the model according to the data patterns (Zhou, 2021), and network security theory, in which it is better to detect the threat by identifying anomalies (Khraisat et al., 2019). The theories were used to supplement positivism as they offered a systematic approach to integrating algorithms and assessing them.

3.3 Research design

The research was based on the quantitative experimental simulation design, which implied controlled experiments with variables in a simulated setting to record the results (Creswell and Creswell, 2018). In particular, it was a pre-experimental form of design, as it was characterized by one group to which interventions are applied (individual and hybrid models), there were no control groups or randomizations, and the comparisons were made before and after actions using the performance measures.

This design was chosen as it made it possible to carry out practical simulations of the DDoS situation with the CICDDoS2019 dataset by repeating the experiments and classifying the attacks and measuring the effectiveness of models. Machine learning studies are best designed through experimental designs, which are suitable because they

allow testing hypotheses in controlled settings, including the division of data into training/testing sets and the quantification of results in terms of F1-scores (Salih and Abdulazeez, 2021). The simulation component focuses on the ethical and practical limitations of real-time network experimentation to prevent disruptions and make simulation of the application-layer attacks (Sharafaldin et al., 2019). Experimental simulation offered causal inferences into the improvement of CART classification by K-Means clustering as opposed to descriptive or correlational designs, which directly served the purpose of creating and testing the hybrid model.

The design is elaborate including secondary data being collected using the CICDDoS2019, pre-processed using Python Scikit-Learn, model was implemented (K-Means using 13 clusters, CART boosted using XGBoost), and evaluated using such metrics as accuracy and V-measure. This systematic methodology provided reliability and compared with the positivist paradigm focus on empirical testing.

3.4 Research Approach

Since this study adopted the positivist research philosophy, it had to adopt a deductive approach. Unlike inductive studies that aim to build a theory after the data collection and analysis, the study works with a pre-conceived hypothesis. It argued that DDoS network attack traffic was unique and peculiar in their frequency, target resource, life span, and source. These features of DDoS attack traffic yield quantitative data that aids in the characterization of the variables. The research's proposal to characterize the variables allowed the investigation of the causal relationship between the flagged network traffic and their malicious source. Any variations in the selected features were proved to be indeed caused by the malicious bombardment of the network by an intruder. The

conclusions inferred from the research were replicable (true to positivism) and generalizable (true to deductive approaches).

3.5 Research Strategy

The study aimed to prove that a hybrid system consisting of two algorithmic models of K-means and CART can efficiently and effectively detect app-DDOS attacks. To do this, it was necessary to train the system to 'know' the difference between legitimate traffic and malicious ones. As such, features of both kinds of network traffic were developed. The system was then trained to know the different data and flag fraudulent requests. The CICDDoS2019 dataset was used to train the system. The data was specifically designed to mirror the current behaviour of DDoS attack traffic.

After training, the system was tested with the data representing real world environment. The hypothesis is that the system was to accurately identify the malicious network traffic fast if not in real time. The proposed study bears all the hallmarks of an experiment hence an experimental strategy was used. The system was tested first against data that is known and labelled appropriately and then with unlabelled data. The results were analysed to gauge the performance of the proposed system. The study was carried out in a cross-sectional time horizon.

The study aimed to measure the performance of the proposed hybrid K-means and CART systems. As such, it measured its performance quantitatively. Quantitative data was collected and analysed statistically. The dataset used to train the model was homogenous. That is, the features of the network traffic considered like the frequency of requests, the packet sizes, and the length of the requests are all quantitative, with few instances of

categorical values which were encoded to numerical values. After simulation, the results were also quantitative. That is, the generated data were values of accuracy which were all represented numerically. Since the research utilized one type of data and, consequently, a single analysis approach, it adopted a mono-method choice of research technique.

3.6 Data Collection and Dataset

This section will describe the data collection procedure and the general description of the dataset applied in the study.

3.6.1 Data Collection

The collection of data with respect to this research was done according to the objectives of the study, and different techniques were applied depending on each objective. Data from the document review was the primary source for the first two objectives, which were about determining what application layer intrusion detection methods are currently in use and finding their weaknesses. This consisted of a systematic analysis of academic articles, technical reports, and industry publications on IDSs as well as application-layer DDoS attacks. The review was to know the weaknesses as well as strengths of contemporary IDS techniques such as signature-based, anomaly-based, and hybrid models for application-layer DDoS attack detection. Furthermore, it created a theoretical foundation upon which this study is built by taking gaps it found from existing literature such as the limited adaptive and hybrid approaches, thereby guiding the derivation of the proposed hybrid model.

Data for objectives 3 and 4, which were the development, evaluation, and comparison of the created hybrid intrusion detection model, were collected from the CICDDoS2019

dataset. This data set was publicly available. It was created by the Canadian Institute for Cybersecurity (CIC) and is relatively relevant to the current application-layer DDoS attacks. It went further to give enough data for training as well as testing the new hybrid model in scenarios similar to the actual environment. The dataset also had significant features such as packet size, flow duration, and attack type to ensure maximum evaluation for the detection performance of known and emerging threats.

3.6.2 Dataset

The CICDDoS2019 dataset was the primary data source for training and testing the proposed hybrid intrusion detection model. The dataset is a relatively comprehensive and publicly available resource that enables research on modern DDoS attacks, even application-layer ones. It was been developed by the CIC and consists of millions of records about normal network traffic along with additional kinds of DDoS attack flow traces, making it apt for intrusion detection evaluation systems.

The dataset comprised of flow duration, packet size, protocol type, attack type, and numerous other parameters of crucial importance in separating legitimate traffic from malicious activity. Such characteristics enabled the creation of a robust model for the detection of advanced application-layer DDoS attacks. This dataset was extremely helpful as it represented a near real-world attack situations and allowed the hybrid model to be tested against real-time current attack scenarios.

To evaluate the performance of the hybrid model, the dataset was split into training and testing subsets. A conventional standard distribution of 67:33 was used. The data comprised and allocated 67% for training purposes while 33% were kept for testing. This

segmentation guaranteed that a sufficient volume is spared for training while leaving a representative amount for evaluation performance. The use of the CICDDoS2019 dataset enabled the research to judge the hybrid model's capability of detecting both existing and emerging application-layer DDoS attacks, which was one of the major issues with the existing intrusion detection systems.

3.7 Population, Sample Size and Sampling Technique

For this research, data with features already extracted by the authors using the CICFlowMeter-V3 was used. Figure 3.1 shows the 12 attack types captured by the CICDDoS2019 dataset, benign traffic, and their flow counts.

Attack Type	Flow Count
Benign	56,863
DDoS_DNS	5,071,011
DDoS_LDAP	2,179,930
DDoS_MSSQL	4,522,492
DDoS_NetBIOS	4,093,279
DDoS_NTP	1,202,642
DDoS_SNMP	5,159,870
DDoS_SSDP	2,610,611
DDoS_SYN	1,582,289
DDoS_TFTP	20,082,580
DDoS_UDP	3,134,645
DDoS_UDP-Lag	366,461
DDoS_WebDDoS	439

Figure 3.1: Attack types in CICDDoS2019 dataset

The model is evaluated using a dataset that consisted of more than 80 network traffic features with benign (legitimate) traffic and 11 DDoS attacks**.** It was a huge constellation of application and network layer DDoS attacks. It came in two folders containing various kinds of Distributed Denial of Service (DDoS) attacks. This research's population is the first file which contained eleven CSV files each with a specific type of DDOS attack used (DrDoS DNS, DrDoS LDAP, DrDoS MSSQL, DrDoS NetBIOS, DrDoS NTP, DrDoS SNMP, DrDoS UDP, Syn, TFTP, and UDPLag).

Figure 3.2 shows a screenshot of the contents of the first folder including their sizes.

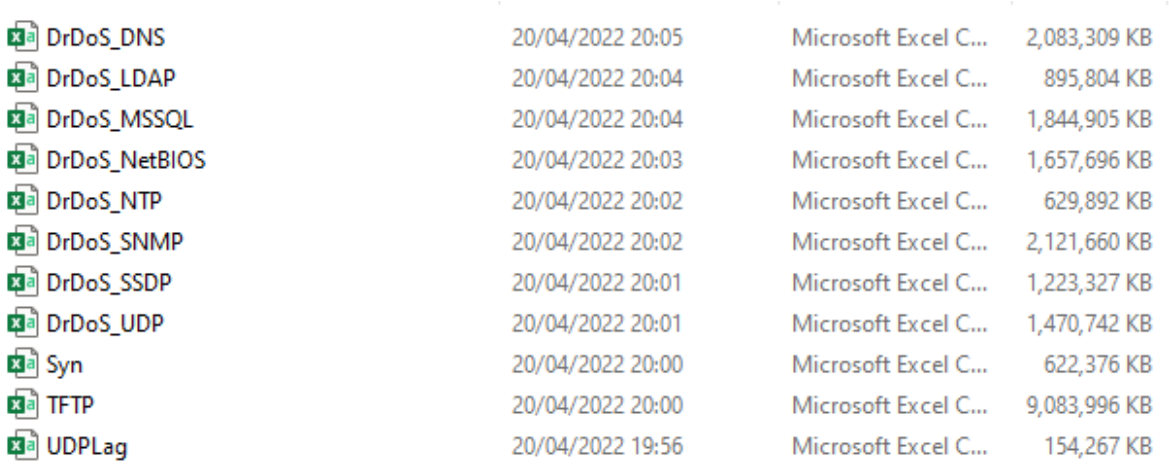


Figure 3.2: CSV files in the dataset

Each CSV file contained more than two million records of each attack. Therefore, the study used simple random sampling to select 30000 records from each attack type as a sample. Also, it had to minimize the computational power required by reading the first one million rows of data in the dataset and thus limiting the sampling to it.

Figure 3.3 shows the utilization of the pandas' framework's data reading function (pd.read_csv) and the random sampling of 100000 records from each attack type (df.sample(30000,random_state=101)).

```
# read the first 200 thousand rows of the dataframe and sample 30 thousand from each dataset

df1 = pd.read_csv('DrDoS_DNS.csv',nrows=200000)
df1 = df1.sample(30000,random_state=101)
df2 = pd.read_csv('DrDoS_LDAP.csv',nrows=200000)
df2 = df2.sample(30000,random_state=101)
df3 = pd.read_csv('DrDoS_MSSQL.csv',nrows=200000)
df3 = df3.sample(30000,random_state=101)
df4 = pd.read_csv('DrDoS_NetBIOS.csv',nrows=200000)
df4 = df4.sample(30000,random_state=101)
df5 = pd.read_csv('DrDoS_NTP.csv',nrows=200000)
df5 = df5.sample(30000,random_state=101)
df6 = pd.read_csv('DrDoS_SNMP.csv',nrows=200000)
df6 = df6.sample(30000,random_state=101)
df7 = pd.read_csv('DrDoS_SSDP.csv',nrows=200000)
df7 = df7.sample(30000,random_state=101)
df8 = pd.read_csv('DrDoS_UDP.csv',nrows=200000)
df8 = df8.sample(30000,random_state=101)
df9 = pd.read_csv('Syn.csv',nrows=200000)
df9 = df9.sample(30000,random_state=101)
df10 = pd.read_csv('TFTP.csv',nrows=200000)
df10 = df10.sample(30000,random_state=101)
df11 = pd.read_csv('UDPLag.csv',nrows=200000)
df11 = df11.sample(30000,random_state=101)
```

Figure 3.3: Random selection of 30,000 records from each attack type

After concatenating the selection, the study got a dataframe containing all the twelve attack types as well as benign traffic entries as shown in Figure 3.4.

```
# check for the labels and remove whitespace
df[' Label'].value_counts()
```

✓ 0.2s

```
Syn              30000
DrDoS_LDAP       29993
DrDoS_SNMP       29988
DrDoS_SSDP       29985
DrDoS_NetBIOS    29918
DrDoS_MSSQL      29904
TFTP             29878
DrDoS_UDP        29848
DrDoS_DNS        29744
UDP-lag          29369
DrDoS_NTP        28151
BENIGN            3169
WebDDoS             53
Name:  Label, dtype: int64
```

Figure 3.4: Attack Labels in sampled dataframe

Ideally, the dataframe should contain 30,000 entries. However, each origin file had benign and WebDDoS unclassified traffic included in it and therefore it intrudes on the final count of each of the other types. The sampled dataframe contains 110,000 rows and 88 columns. Also, it has a variety of data types ranging from strings to objects as shown in Figure 3.5.

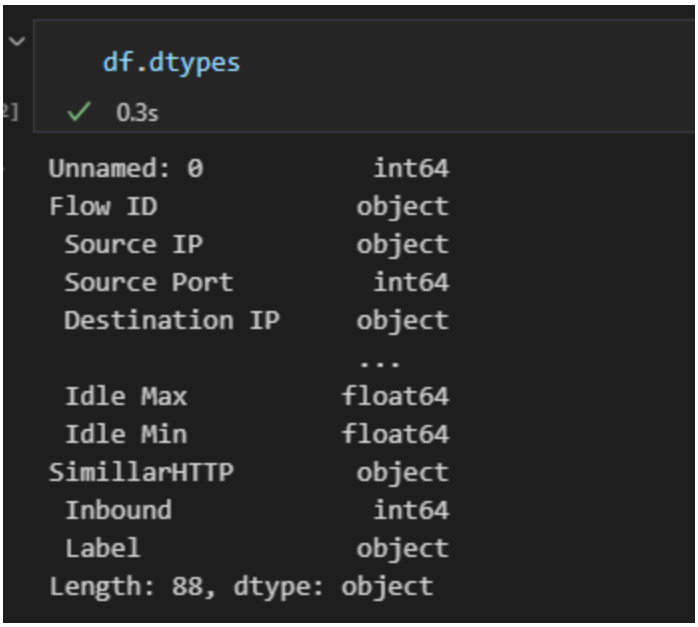


Figure 3.5: Datatypes in the dataframe

3.8Materials

This research utilized a personal computer (Dell Latitude E7470) with an Intel Core i5 processor and installed random access memory of 8GB. The analysis tools used included Microsoft Excel, Jupyter Notebook, Visual Studio Code, and Scikit-Learn Machine Learning Libraries where the implementation of the CART and K-Means individual algorithms were sourced. Windows was the main operating system used for the entire process of data manipulation.

3.9 Data Analysis

The data analysis in this study was systematically done to critically examine the CICDDoS2019 dataset and derive insightful understanding to meet the objectives of the study. Therefore, the analysis, ensured an understanding of the characteristics of the dataset, discovered important patterns, and evaluated the effectiveness of the proposed

hybrid intrusion detection model. Here are the important techniques and steps used in the process of data analysis.

The first stage of data analysis is exploratory data analysis (EDA), which checks data structure, distribution, and relationships. It also involves selecting important summary statistics of the key features. Other techniques considered for identifying trends, outliers, and possible omissions are histograms, box plots, and scatter plots. For example, the distribution of traffic types (benign vs. malicious) is made to be representative of a real-world environment. EDA is an important first step toward a general perspective of the dataset that directs further analysis and model development.

The second stage was cluster analysis with K-Means. The network traffic data is grouped into clusters based on similarities in a manner that any patterns and peculiarities that indicate possible DDoS attacks can be detected. The Elbow Method was used to come up with the optimum number of clusters that ensured the clustering results were the best and most informative. The K-Means clustering process led to the formation of clusters that were later analysed to see if they could differentiate between normal and malicious traffic. This analysis would prove invaluable since it contains critical information about the structure of the data that would be used to develop the hybrid model.

CART was then used for classification analysis. New feature cluster labels generated through K-Means are also added to the dataset, which improves model effectiveness in identifying groups of network traffic as benign or malicious. The newly created data labels that were created from the previous K-Means analysis were used to perform training on the CART algorithm. Evaluation of the performance of the trained CART classifier was done in terms of different performance metrics, including accuracy,

precision, recall, and F1 score. Also, the generated decision tree trained by CART was investigated for the most significant features in relation to DDoS attack detection and provides some insight into the different aspects which increase the predictive power of the model.

The last aspect reviewed was the performance of the hybrid model. Evaluations analysed how well the model could detect application-layer DDoS attacks from the test dataset to minimize false positives and negatives. Evaluation metrics used included accuracy, precision, recall, F1 score, and false positive rate (FPR). The hybrid model was also compared with stand-alone K-Means and CART techniques to validate its superiority in solving DDoS attacks in modern scenarios.

Finally, the data analysis process involved iterative fine-tuning of the hybrid model. Hyperparameter tuning, in this case, includes the number of clusters in K-Means and the depth of the decision tree in CART, as well as the additions of new features or transformations to increase the predictive power of the model. Evaluation inputs were also factored into the refinement phase to ensure that the model kept pace with variations in attack signatures.

3.10 Model Development Methodology

Preprocessing or cleansing data for analysis is the first step in this procedure. With a dataframe of multiple data types, nulls, and vastly ranged integers, the proposed algorithm would not be able to optimally work on the data. At this stage, infinite values with the minimum and maximum values in the dataset were dropped. This data preparation and cleaning stage also included optimizing the dataframe for memory usage

by converting float values to float32. Further, NaN (not a number) values were replaced with minimum values, and NAs (not available) with zero. Unassigned int types were assigned to their respective datatype values. Finally, the unnamed columns were dropped.

The dataset contains two IP address columns. IP addresses are not integers or float data types and as such, would not be identified properly by the algorithms. It was prudent, therefore to convert them to suitable variables for the action of the proposed algorithms. I used one-hot encoding to achieve this goal.

After cleaning the data and processing it using the proposed algorithms, the data was visualized. A graph of the data counts appears as shown in Figure 3.6.

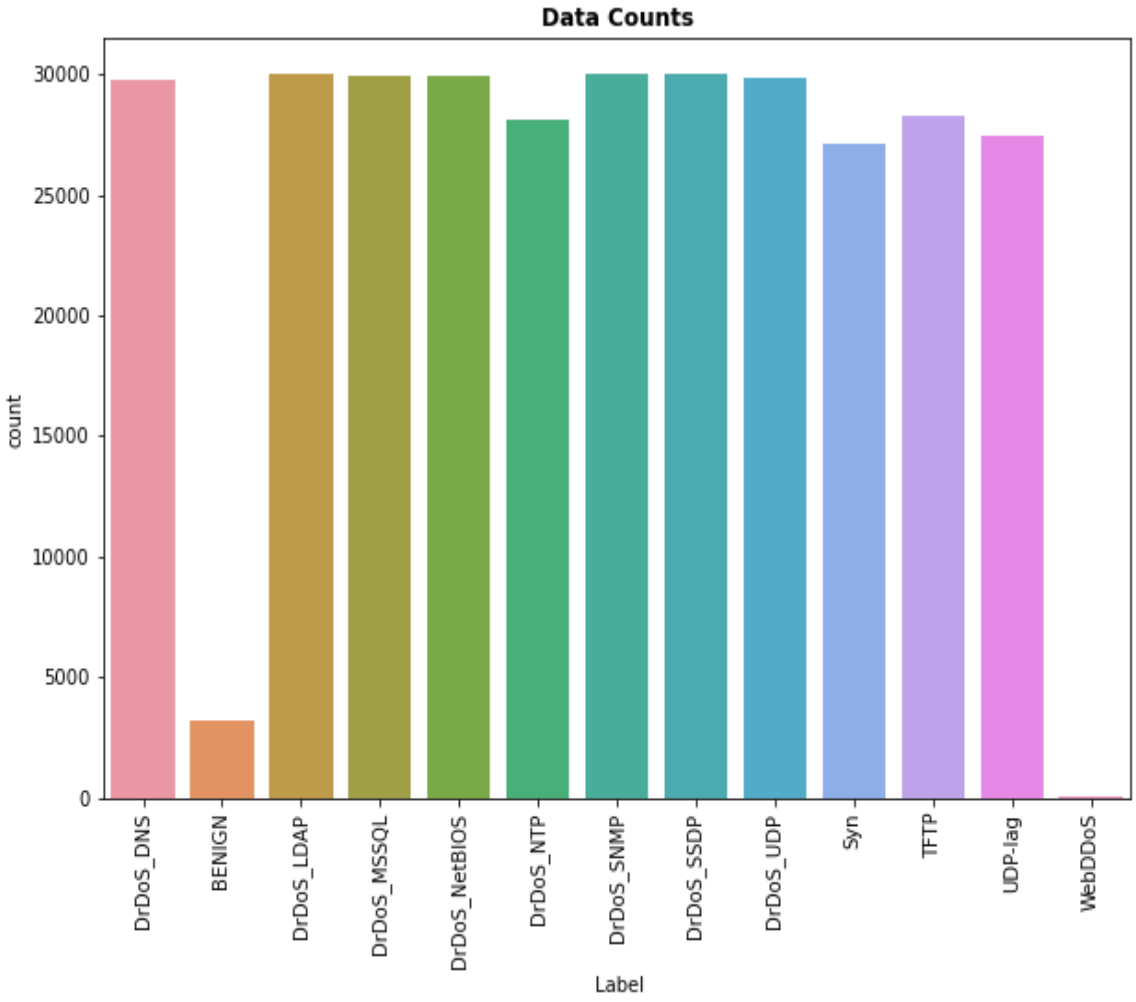


Figure 3.1: Data Counts Bar Graph

The figure 3.7 demonstrates the distribution of the types of intrusion attacks in the dataset on which the current study was conducted. The pie chart shows the proportional number of the different types of DDoS attack and benign traffic, which provides an overview of class composition before the training and testing of the models.

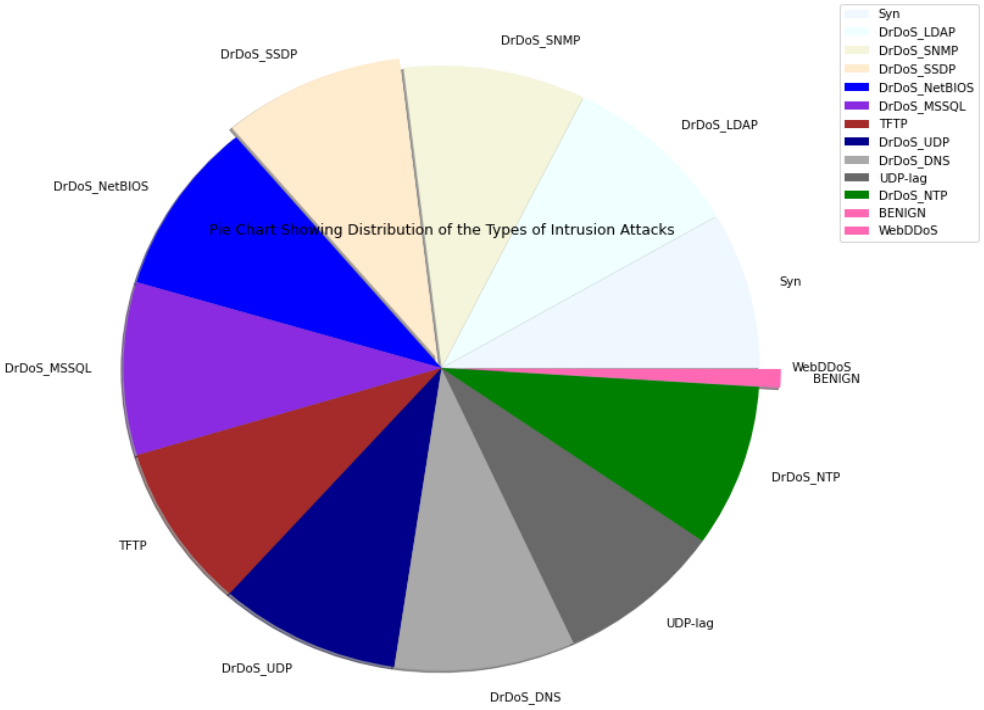


Figure 3.2: Attack type distribution Pie Chart

Next K-Means clustering was applied with an unsupervised learning algorithm that clustered the network traffic data according to its similarity. The determination of the optimal value of k was be done using either the Elbow Method. Each data point was associated to the nearest centroid by the Euclidean distance which thus generated the cluster labels pertaining to an analogous network traffic behaviour. These cluster labels were then used as further features in the classification stage. By identifying hidden patterns and anomalies in the data, K-Means provided an excellent basis for the hybrid model to detect unusual behaviour within the traffic.

At the very next step, the clustered data were classified to benign or malicious traffic through the Classification and Regression Trees (CART) algorithm. A new feature would be added to the dataset by the cluster labels generated from the K-Means algorithm,

enhancing the ability of the model in distinguishing normal traffic apart from the attack traffic. The CART algorithm was then trained on this augmented dataset, building a decision tree in order to classify traffic based on the combination of the above said features. The generated decision tree was pruned by appropriate optimization of its depth and complexity in order to avoid overfitting and ensure better generalization.

K-Means and CART outputs thus integrated were aimed toward making the hybrid model. The advantage of the model was that it was taking advantage of both algorithms; K-Means could capture hidden structures and anomalies generated from network traffic while CART classified all traffic in the cluster as either being benign or malicious. Thus combined, the hybrid model would be capable of achieving higher accuracy in detection and lower false alarm rates compared to such models as those single-handedly developed by their technique.

The test data are used to evaluate the performance of the hybrid model. Important evaluation metrics included accuracy, precision, recall, F1 score, and false positive rate (FPR) because these indicators could cover all aspects of the performance model in case of application-layer DDoS attacks while minimizing the error. Results of the hybrid model were compared with that of lone K-Means and CART models, along with other techniques used commonly in IDS, to convince the difference in superiority against the other proposed models for the case of ongoing DDoS attacks.

Thus, the last thing was iterative improvement in the model. Hyperparameter tuning like the number of clusters in K-Means or the depth of the decision tree in CART, and experimenting with other features or transformations were done to create a better

prediction model. Additional considerations came from the evaluation phases; thus, the model is open enough to be adapted in the future given a new attack pattern.

3.11 Reliability and Validity

Reliability and validity were two factors that were vital in this research so that the findings were credible and accurate. To meet reliability, the CICDDoS2019 dataset, one such well-known public resource for DDoS attack detection, was used. The consistency of the dataset with reproducibility was achieved by applying the standardized preprocessing steps such as data cleaning, normalization, and feature selection. And further tested the hybrid model's performance on different metrics like accuracy, precision, recall, and F1 score to make a strong argument for reliability. Alignment between the objectives and research design, and well-accepted machine learning techniques like K-Means clustering and CART for classification reinforced the study's aim of validity. A 70:30 train-test split further validated against unseen data for the model's generalizability. Thus, it guarantees the results are not much overfitted in favour of the training data set.

3.12 Ethical considerations

Ethical considerations have been incorporated all throughout the conduct of the study. Since the dataset used in the study was publicly available and anonymized by CICDDoS2019, such privacy or confidentiality issues never arose. Personally identifiable information (PII) is also not available and does not link an individual or an organization to the dataset and allows no possibility of harming either. Further, this study upholds and adheres to academic integrity in terms of citation and plagiarism. The development and

validation of the hybrid model were all done with openness in mind towards achieving better network security and less impact of DDoS attacks. With such ethical standards, therefore, the research maintains credibility and responsible contribution to the discipline of cybersecurity.

3.13 Chapter Summary

The chapter outlined the research methodology that was be used to design, implement, and evaluate the proposed hybrid model of intrusion detection. It outlined the research design, data set sample, data preprocessing methods, and feature shrinkage measures used to cope with the complexity of the CICDDoS2019 data set. The chapter described how the K-Means clustering algorithm and the CART classifier were applied as well as how both models were combined into a hybrid one. Measurements of performance evaluation and validation processes were also addressed to give reliability and strength of the findings. The issues of ethics and integrity of data were discussed to promote responsible research. In general, this chapter has created a systematic system of experimental review, which is the basis of the data analysis and interpretation in Chapter Four.

CHAPTER FOUR

DATA PRESENTATION, ANALYSIS AND INTERPRETATION

4.1 Introduction

Data analysis results that seek to meet the research objectives and answer the pertinent questions posed by the study are presented in this chapter. The chapter has been organized according to a logical progression, moving from the presentation of the findings, through their detailed analysis, and ending with interpretation. The findings are analysed based on the research questions directed toward appreciating existing application-layer intrusion detection techniques in their strengths and weaknesses while outlining the extent to which the hybrid model under study is able to detect application-layer DDoS attacks. Whereas the presentation of the results addresses detection accuracy improvements as brought about by the hybrid model, comparisons demonstrating its performance against constituent techniques and arguments for mitigation of traditional intrusion detection system constraints are also presented.

This chapter has been designed to support the aims of research of the study. The results of the data preprocessing and unsupervised learning are introduced initially so that the data properties and structure could be defined. This is succeeded by the analysis of the supervised CART classifier and the suggested hybrid K-Means-CART model. The earlier-identified, appropriate performance metrics are used to analyse the results in order to determine how effectively they detect application-layer DDoS attacks.

4.2 Data Analysis

4.2.1 Unsupervised Learning Using K-Means

For a dataset with more than 80 features, it would be prudent to find the most statistically significant features for the next stages of data analysis. In order to find statistically significant correlations between the dataset features and to derive the further dimensionality reduction, correlation analysis was conducted. As Figure 4.1 demonstrates, only a few pairs of features have a strong correlation, with the rest having weak or no one at all. This validates the large dimensionality and sparsity of the data, which supported the necessity of reducing the number of dimensions before clustering during the preliminary phase.

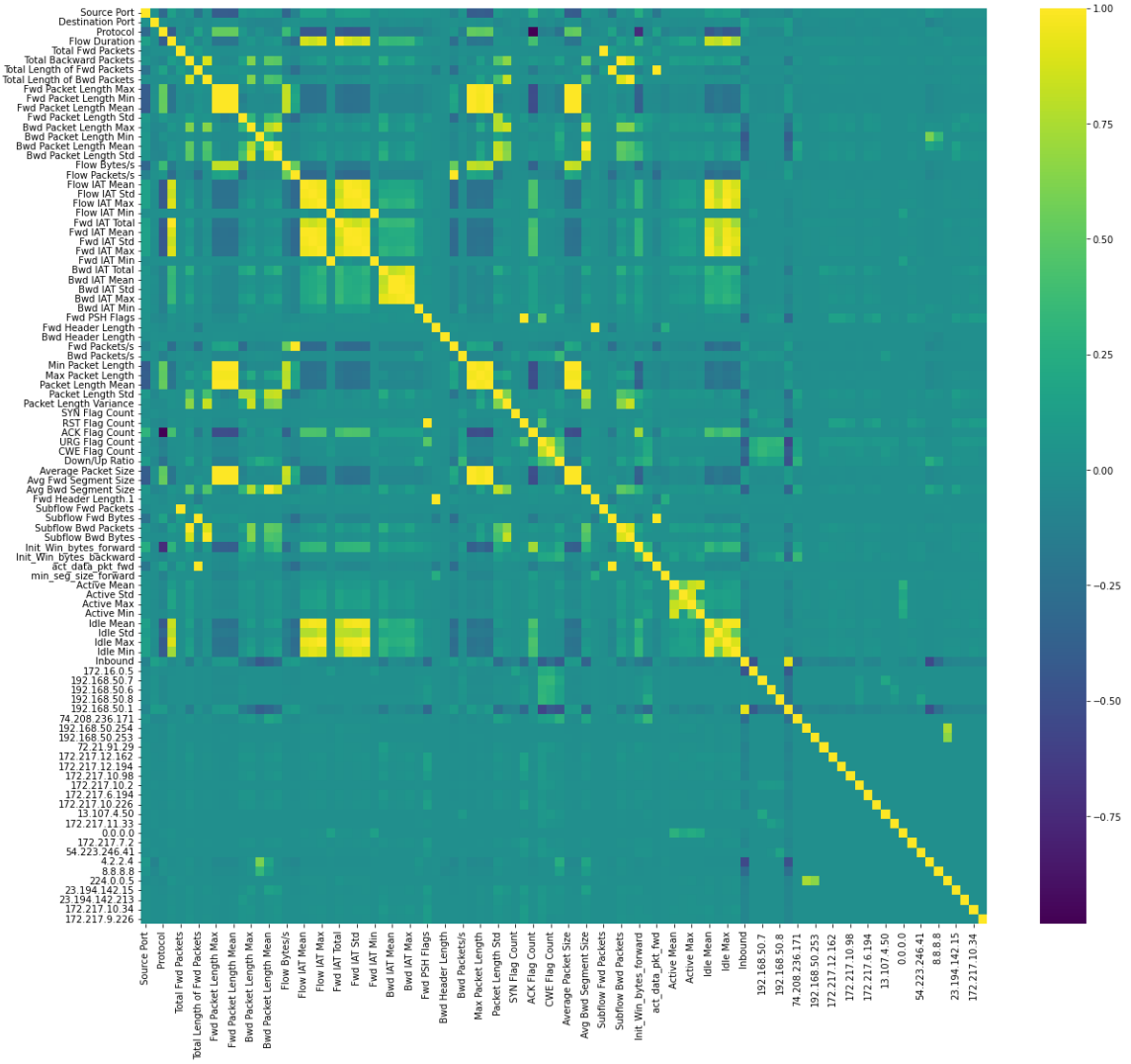


Figure 4.1: Correlation Graph

The step showed that 2 datapoints were negatively correlated, 252 were positively correlated and 9212 were non-correlated.

The K-Means offers an unsupervised learning technique. It works through clustering of unlabelled data by trying to separate samples into n groups of equal variances. This approach is based on the pursuit of minimizing inertia. Inertia in this regard is defined as within-cluster sum-of-squares (Scikit Learn, n.d.). The algorithm requires the user to

specify the value of n, which represents the number of clusters that the algorithm will associate the data with.

The algorithm divides a set of ***N*** samples ***X*** into ***K*** disjoint clusters ***C***, each specified by the mean $\boldsymbol{\mu_j}$ of the cluster samples. This mean is referred to as centroids. K-Means aims to select centroids that minimize inertia or the heterogeneity of cluster members and homogeneity of clusters.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

K-Means, also referred to as Lloyd's algorithm, is implemented in three steps. Firstly, the selection of the initial centroids is carried out. This can be done randomly, through the discretion of the implementer, or by choosing ***k*** samples from the dataset ***X*.** After this step the algorithm loops between the other two steps. The first one of these steps includes assigning each sample to its nearest centroid. The final activity of the first loop is to create new centroids by calculating the new mean of the items assigned to each centroid. The final two steps are repeated until the means become constant or centroids stop changing in any significant manner.

4.2.1.1 Data Standardization

K-Means could not process the dataframe as it was currently constituted. The dataset contained rows that contain values that were neither floats nor integers. The numerical processing by machine learning algorithms required that the categorical attack labels be encoded into numerical form. A simple convention of assigning each label an integer

starting from 0 and ending at 12 covering 12 attacks and 1 benign traffic was applied. Table 4.1 shows this step.

Table 4.1: Label Encoding

True Label Names	New Encoded Name
Syn	0
DrDoS_LDAP	1
DrDoS_SNMP	2
DrDoS_SSDP	3
DrDoS NetBIOS	4
DrDoS_MSSQL	5
DrDoS_UDP	6
TFTP	7
DrDoS_DNS	8
UDP-Lag	9
DrDoS_NTP	10
BENIGN	11
WebDDoS	12

Class representation was checked with the distribution of class labels of codes in the dataset. Figure 4.2 shows frequencies of each of the encoded classes, which shows that the dataset is fairly well balanced between the types of attacks and the benign traffic.

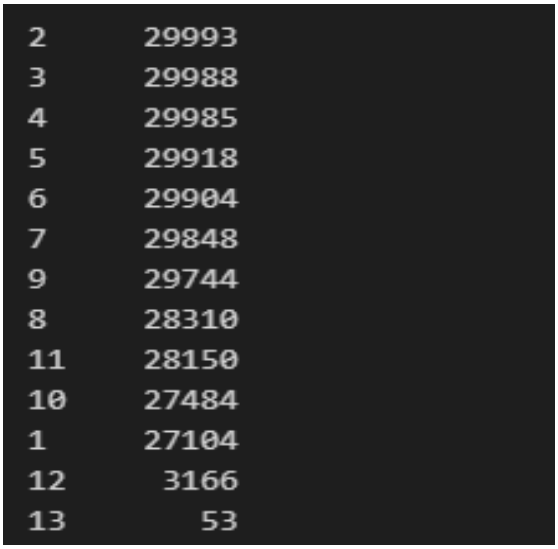


Figure 4.2: Encoded Labels and Respective Value Counts

As an unsupervised technique, K-Means does not require the labels column in the dataset. As such, the next step in using K-Means is to drop the labels column. Also, columns that are neither int nor float are dropped at this point. It is also important to note that the data values are so spread out that visualizing the results would not be visually informative. Therefore, it was essential to scale the data.

Also, the data had a large number of variables such that the dispersion matrix would be too large to study or render any informative visualizations. To achieve a more informative result, it was necessary to decompose the scaled data and reduce the number of variables using Principal Component Analysis (PCA). This technique works by generating linear combinations of variables and representing them as principal components. It converts correlated features into linearly uncorrelated features using orthogonal transformation. Importantly, it reduces the dimensions of a ***d***-dimensional dataset by projecting it onto a (***k***)-dimensional subspace (where $k < d$). This reduces the computational resources needed to perform analysis without sacrificing significant

informational features of the data. Ultimately, it improves the efficiency of any subsequent computations. It was imperative to select a convenient value for ***k*** to maximize the efficiency of the technique. To make this important decision, I used eigen decomposition using a covariance matrix and dropped eigen pairs with the least values corresponding to the information they represent.

4.2.1.2 K-Means Data Preparation with PCA

After selecting a relatively large percentage (95%) of the targeted variance to be explained by eigen components, PCA was used to project them into a subspace. To determine the optimal number of principal components for dimensionality reduction, cumulative explained variance was analysed. Figures 4.3 demonstrate that seven principal components capture approximately 97.38% of the total variance.

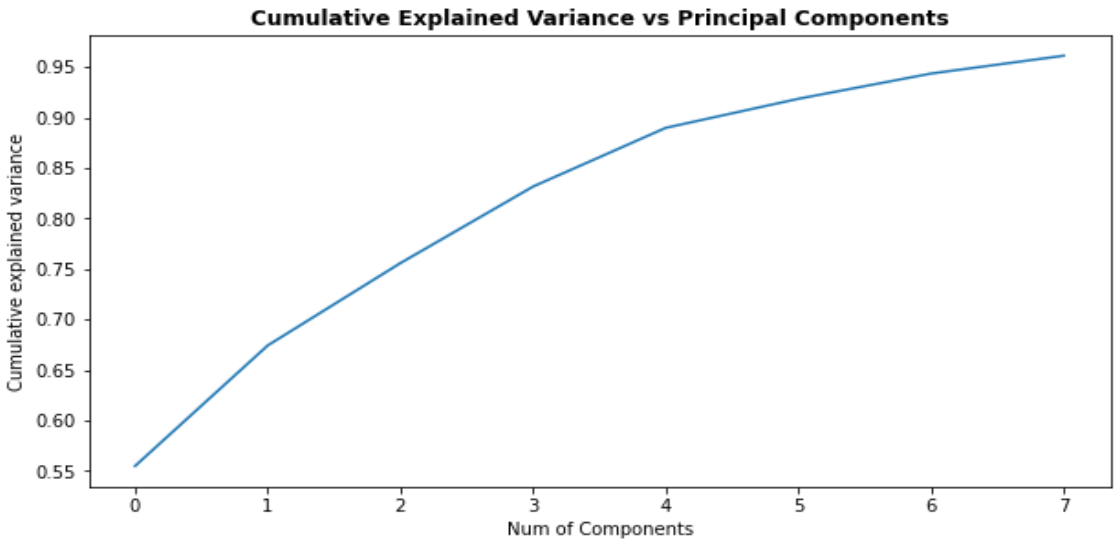


Figure 4.3: A Graph of Cumulative Explained Variance against Principal Components

An extended analysis of cumulative variance showed that twelve components account for nearly 100%. Based on this observation, seven components were selected to balance dimensionality reduction and information retention.

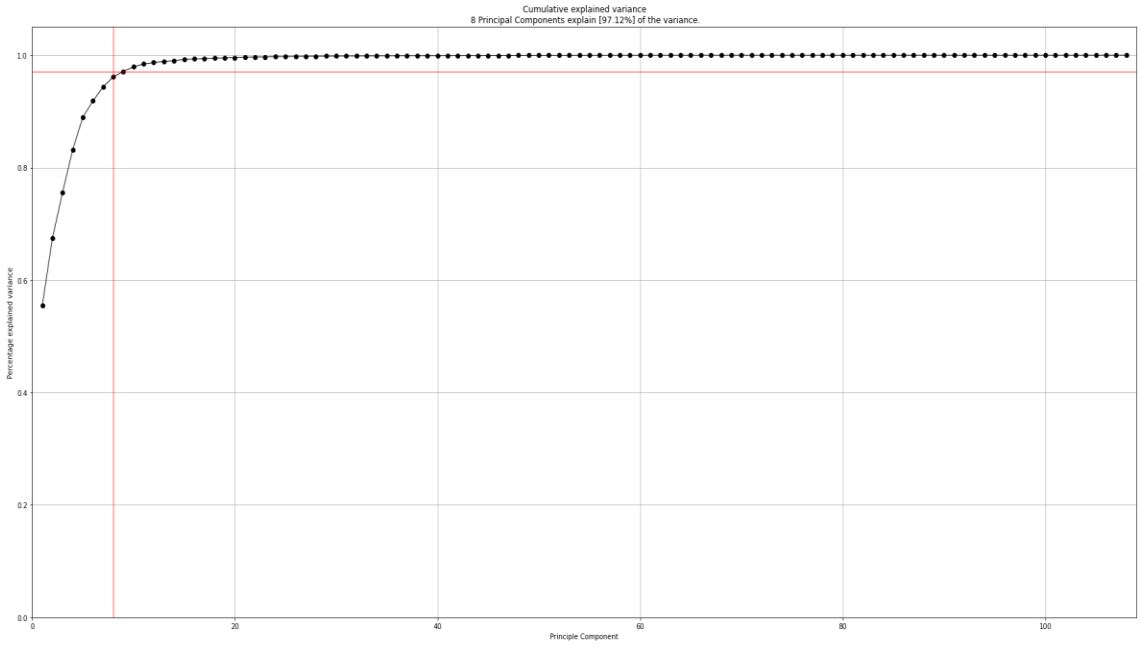


Figure 4.4: An extended Graph of Cumulative Explained Variance against Principal Components

A biplot of how PCA models the data is as shown in Figure 4.5. PCA biplots were produced to visualise the distributions of the transformed features in reduced dimensional space to show that application-layer DDoS traffic bear similar features among multiple categories.

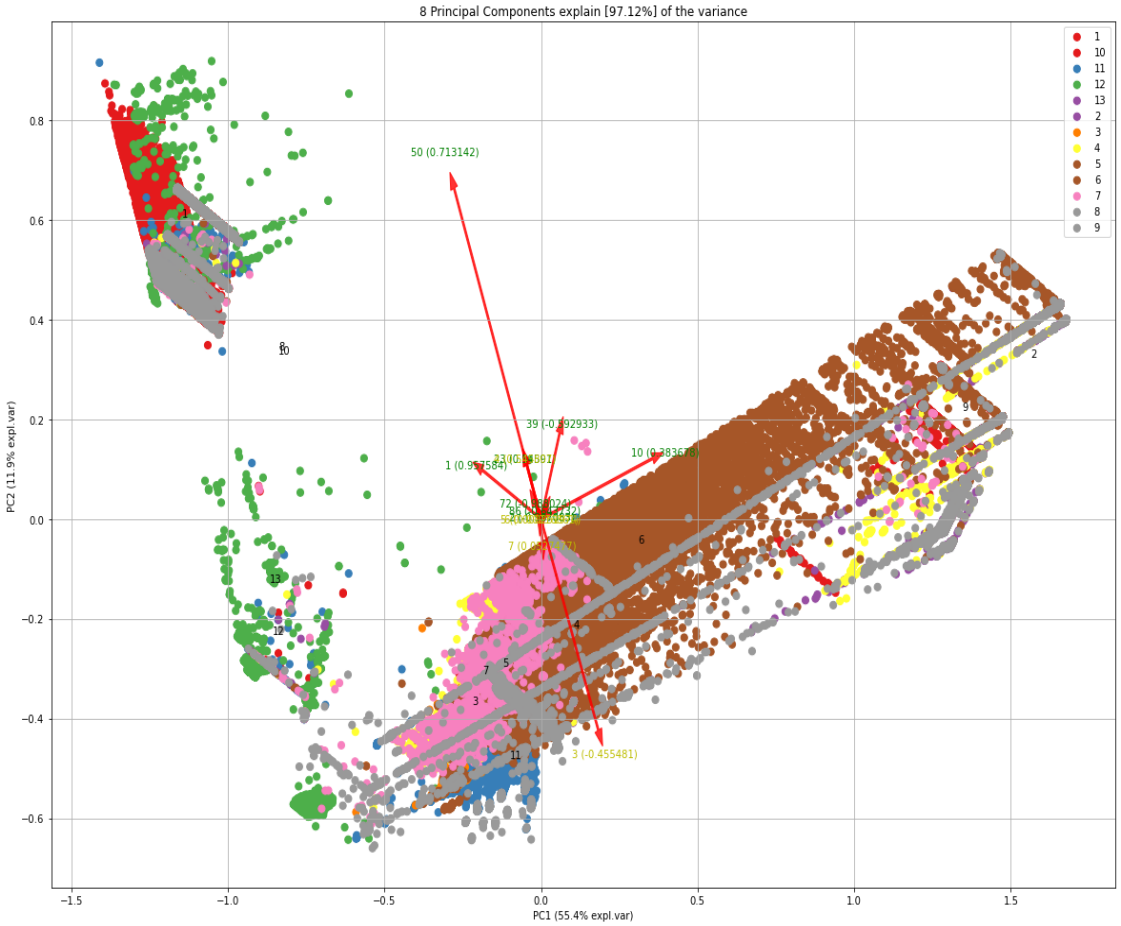


Figure 4.5: A Biplot of the PCA Model of the Dataframe

A similar Biplot using the first principal component PC1 against the second one PC2 is as shown in Figure 4.6. The biplots in Figures 4.5 and 4.6 display that there is a high overlap of multiple attack classes suggesting that application-layer DDoS traffic exhibit similarities to many categories. This overlap poses a problem to entirely unsupervised clustering methods.

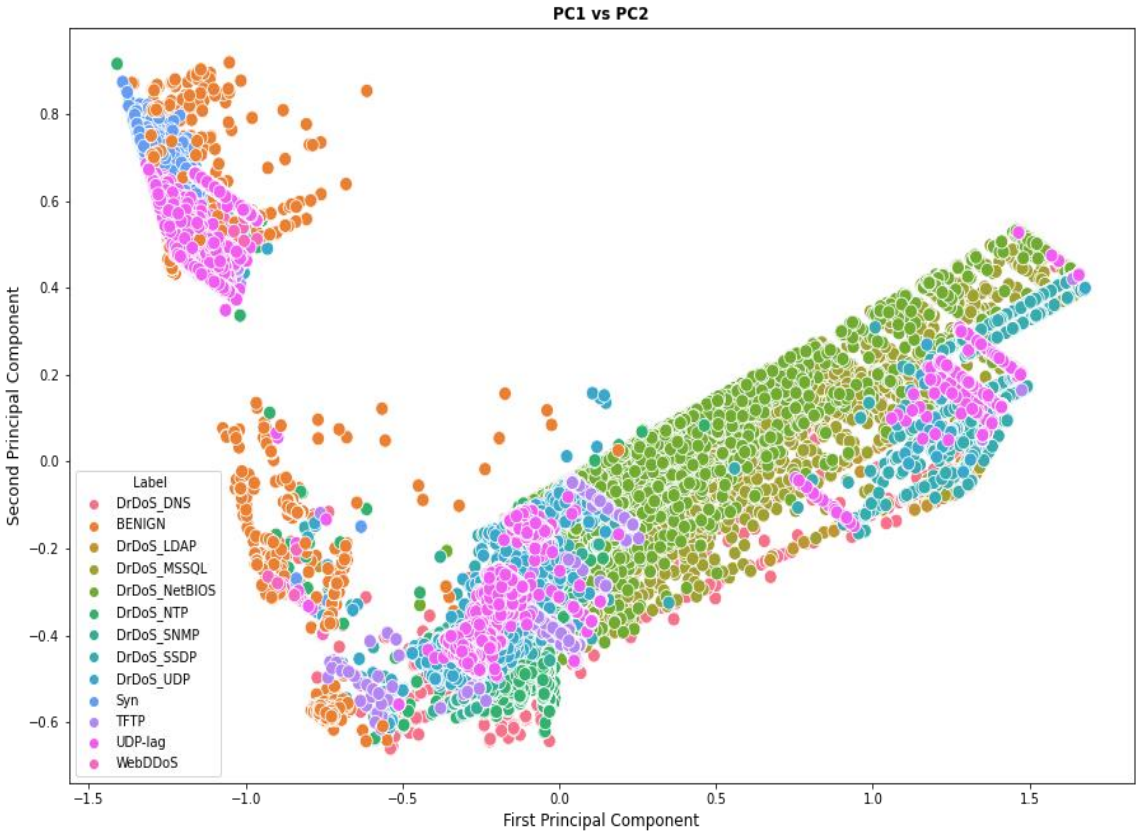


Figure 4.6: A Biplot of the First Principal Component against the Second Principal Component

The dataset is now conveniently structured for clustering with the K-Means algorithm. The algorithm allows the selection of the value of ***k***. This step is often achieved by selecting a conveniently random value or through a more statistical method like the elbow method by inertia. To decide how many clusters the algorithm should use during the K-Means implementation, the Elbow Method was used to calculate the inertia values. The graph of ***k*** against inertia yields a rounded curve where an elbow should have been as shown in Figure 4.7. Figure 4.7 indicates that there is no clear elbow point, which is an indication of the lack of clearly defined natural clusters in the data. Therefore, domain

knowledge was implemented to pick K = 13 which is equal to the known number of classes of traffic.

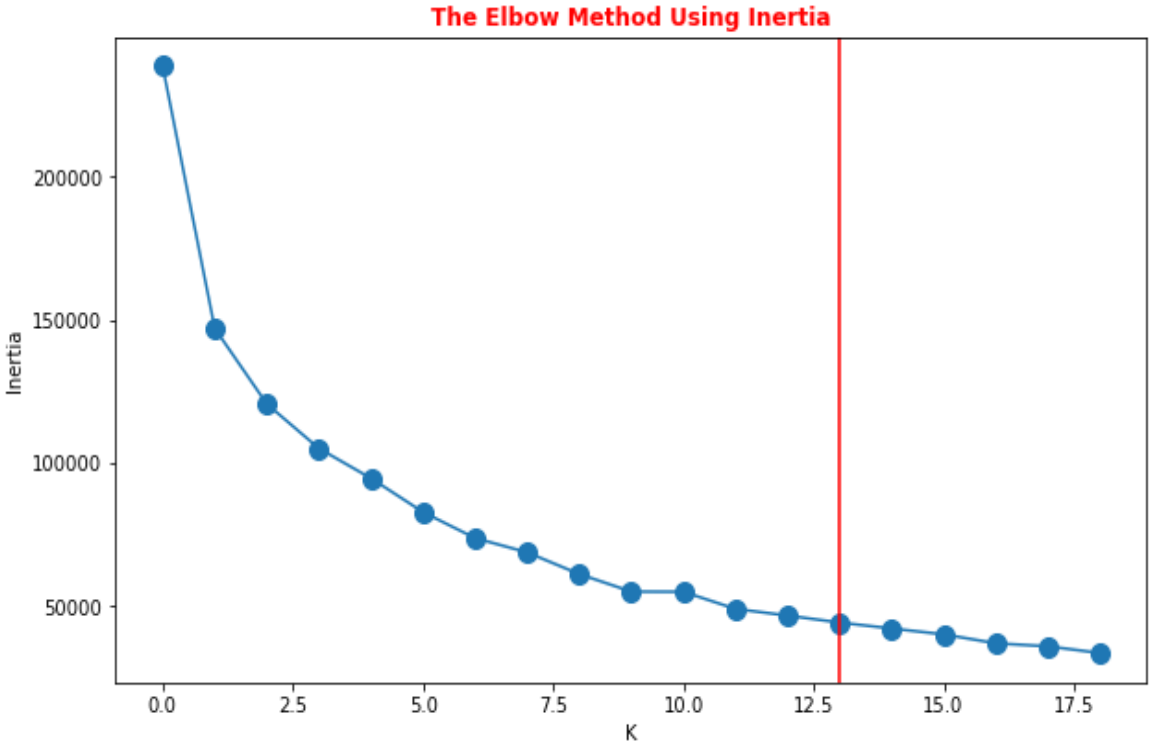


Figure 4.7: The Elbow Method Using Inertia

This drawback is inconsequential because prior knowledge about the dataset and the distinct groups of data contained therein is available. The dataset contains 13 types of data including 12 attack types and benign network traffic. It would be prudent and mathematically convenient to adopt 13 as the value of ***k***.

4.2.2 Supervised Learning Using CART Algorithm

CART is a binary classification algorithm that factorizes data using Gini's impurity index. The algorithm splits data into child nodes repeatedly with an increasing tree depth. It works in the following three steps;

i. Find the best split for a feature. Assuming that each feature has **K** different values, there exist **K**-1 possible splits. A split that maximizes the splitting criterion is chosen such that each has the best split for the respective feature.

ii. Find the node's best split. From step, i. splits, find the one which maximizes the splitting criterion.

iii. Split the node using step II best node split and repeat from step I until the stopping criterion is met.

The splitting criterion uses Gini's impurity index, which is defined for the node ***t*** as follows;

$$i(t) = \sum_{i,j} C(i|j)p(i|t)p(j|t)$$

Where *C(i|j)* is the cost of misclassifying a class *j* case as a class *i* case (in our case *C(i|J)* = 1, if $i \neq j$ and *C(i|j)* = 0 if $i = j$), *p(i|t).(p(j|t)* respectively*)* is the probability of a case in class *i (j)* given that falls into node *t.*

In CART, training and test sets should be established from the dataset. For the training set, 67% of the data and 33% for testing the model was used.

4.2.3 K-Means, CART Hybrid Model

Both algorithms work in a manner that allows a hybrid prediction model to be created using their features. Specifically, a feature of K-Means allows its results to be included as part of the input data for the CART algorithm. K-Means creates ***k***-clusters of the dataset. If ***k*** is specified to be equal to the number of groups of data in the dataset, it can create categories that are roughly aligned with the true labels of the data. This will improve classification accuracy if these categories are encoded and passed as categorical values into the CART. Conveniently, CART factors in categorical values during classification.

K-Means clusters were appended as a feature of the dataset. The clusters were represented as numeric categorical values. Figure 4.8 illustrates the hybrid model that combines K-Means clustering and CART classification.

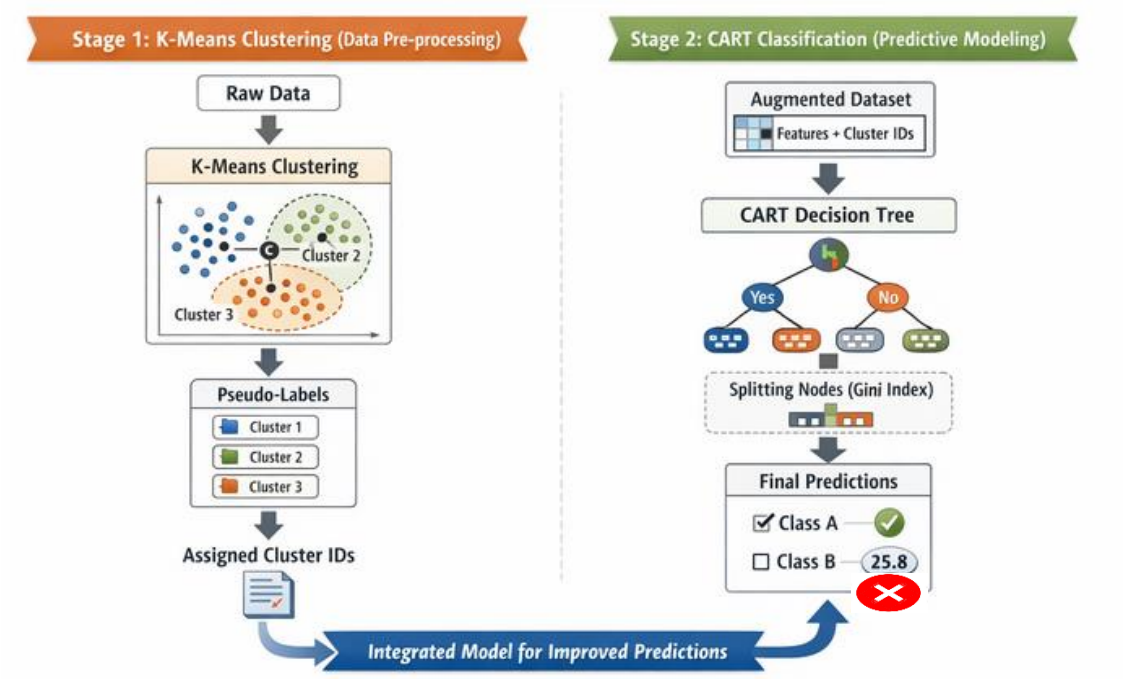


Figure 4.8: Hybrid Model

4.3 Results

This section presents the results obtained from the analysis and evaluation conducted in this study.

4.3.1 Existing Application-Layer Intrusion Detection Techniques and Models

Various techniques and models for application-layer DDoS attack detection exist, according to the explored literature. Most approaches have been traditional, which include signature-based detection, anomaly-based detection, and hybrid models, albeit with varying success. For instance, Singhal et al. (2020) proposed log analysis system based on big data technological feature detection of DoS and DDoS attacks by analysis of network traffic patterns. Support vector machines (SVM), decision trees (DT), random forests (RF), k-nearest neighbour (KNN), naive Bayes (NB), and neural networks (NN) are also other good, commonly used models (Kumar et al., 2022). These models were experimentally tested using datasets such as CAIDA UCSD, KDD Cups '99, CICIDS2017, and CICDDoS2019, with much variability. For instance, Radoglou-Grammatikis and Sarigiannidis (2019) reported an accuracy of 99.66% from the CART algorithm on the dataset, indicating the possibilities for intrusion detection via decision tree-based models. The literature has also pointed out the shortcomings of these models in dealing with the ever-dynamic, ever-evolving nature of application-layer DDoS attacks.

4.3.2 Weaknesses of Existing Intrusion Detection Models and Proposed Improvements

Some existing models for intrusion detection these days have several shortcomings which can be high false-positive rate, the inability to detect zero-day attacks, and three reliance upon assumptions like feature independence or linear decision rule boundaries. For example, Naïve Bayes and Logistic Regression failed to perform well on the CICDDoS2019 dataset, scoring an F1-score of 5% and 4%, respectively, due to the assumption they made about data distribution. Yet again, the K-Means clustering algorithm identifies patterns but is not very good at unbalanced cluster sizes and overlapping data points (Fränti and Sieranoja 2019). To face these weaknesses, researchers proposed a few improvements among which we find hybrid models that combine algorithms. For example, Jia et al. (2017) showed that merging K-Means with C4.5 helps improve the accuracy of intrusion detection. Also, the work of Radoglou-Grammatikis and Sarigiannidis (2019) emphasized the effectiveness of modular IDS systems combining various detection techniques. These improvements imply that an increase in detection accuracy and adaptability against evolving threats may be achieved if we combine unsupervised and supervised learning approaches, K-Means as unsupervised and CART as supervised.

4.3.3 Performance of Individual K-Means and CART Algorithm

This subsection shows the results of performance of all machine learning algorithms that in turn were tested individually and then were combined to create a hybrid one.

4.3.3.1 Performance of Unsupervised Learning with K-Means

K-Means achieved moderate success in clustering the dataset. To evaluate the performance of K-Means clustering, the generated clusters were compared against the true class labels. Figure 4.9 shows the comparison of K-Means clusters with the true clusters of the dataset.

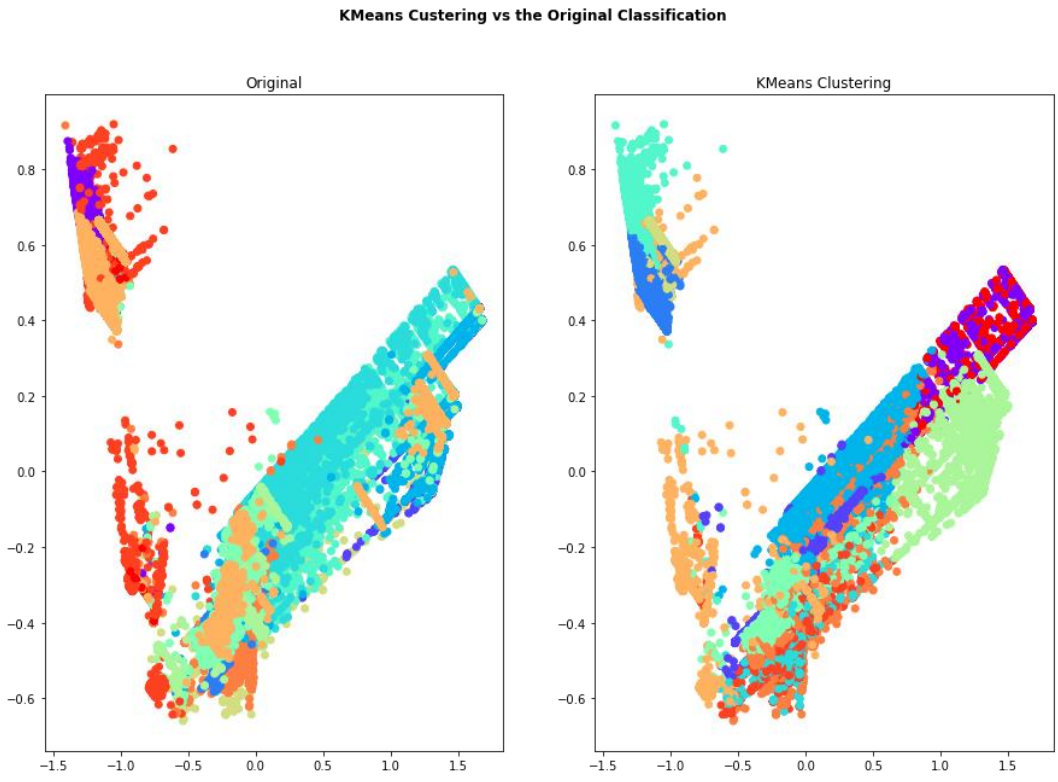


Figure 4.9: K-Means Clusters Compared to Original Clusters in the Dataset

Figure 4.10 illustrates the measures that were used to evaluate the K-Means model and the respective scores achieved by the model. In the figure, the performance metrics of the clusters are summed up, which is found to be moderate, because of overlapping classes and disproportional cluster structures.

```
    # Get homogeneity, completeness, and V-measure scores
    hcv = homogeneity_completeness_v_measure(labels, k_labels)

    # Print the scores
    print("Homogeneity: ", hcv[0])
    print("Completeness: ", hcv[1])
    print("V-measure: ", hcv[2])
 ✓  0.5s
Homogeneity:  0.5076435029731041
Completeness:  0.5195214434581532
V-measure:  0.5135137960909583
```

Figure 4.10: K-means measures

4.3.3.1.1 Homogeneity Score

I used three metrics to measure how well the algorithm grouped similar data points and how different groups are from each other. The first one is the homogeneity score. This measure checks each cluster's members and rates how they are alike. It shows how well the algorithm classified data in the same cluster. Vale (2021) notes that it is defined by Shannon's entropy as follows;

$$h = 1 - \frac{H(C|K)}{H(C)}$$

Given that the formula for *H(C|K)* is;

$$H(C|K) = -\sum_{c,k} \frac{n_{ck}}{N} \log\left(\frac{n_{ck}}{n_k}\right)$$

Where nc_k / n_k is the ratio between the number of samples labelled ***c*** in cluster ***k*** and the total number of samples in cluster ***k***.

The ideal homogeneity of 1 is achieved when all samples in cluster k have the same label ***c***. For this measure, K-Means achieved a **0.5076** success rate in classifying similar attacks into the same groups in the CICDDoS2019 dataset.

4.3.3.1.2 Completeness Score

The second measure is the completeness score. This metric ascertains the success of the algorithm in classifying all similar data points. It checks to see how the algorithm fared in assigning each record of data to the right group. The right group will be the one with members with similar characteristics (Vale, 2021). It is given as;

$$c = 1 - \frac{H(K|C)}{H(K)}$$

Where H(K|C) contains nc_k / nc, representing the ratio between samples ***c*** in cluster ***k***. Completeness is achieved when all samples of kind ***c*** have been assigned to the same cluster ***k*** representing a score of 1. The K-Means achieved an average score of **0.5195** in this regard.

4.3.3.1.3 V-Measure

The final metric to assess the performance of the K-Means algorithm with the CICDDOS2019 dataset as a whole is v-measure. The v-measure, also referred to as Normalized Mutual Information (NMI), combines the previous two measures and checks to ascertain how the algorithm managed to classify each data record to its rightful cluster well and does it for all existing datasets. According to Vale (2021), it is given by;

$$NMI = 2 * \frac{h * c}{h + c}$$

Where *c* is the completeness score and *h* is the homogeneity score.

Ultimately, it is a measure of how well the algorithm manages to make all clusters contain similar data with significant distinctions between clusters such that the clusters and significant heterogeneity of the clusters with each other are homogeneous. In this regard, K-Means posted a moderate score of **0.5135**.

4.3.3.2 Performance of Supervised Learning with CART

The assessment of CART was done through the measure of accuracy, precision, recall, and F1-Score. Figure 4.11 shows the confusion matrix of the CART classification of the data.

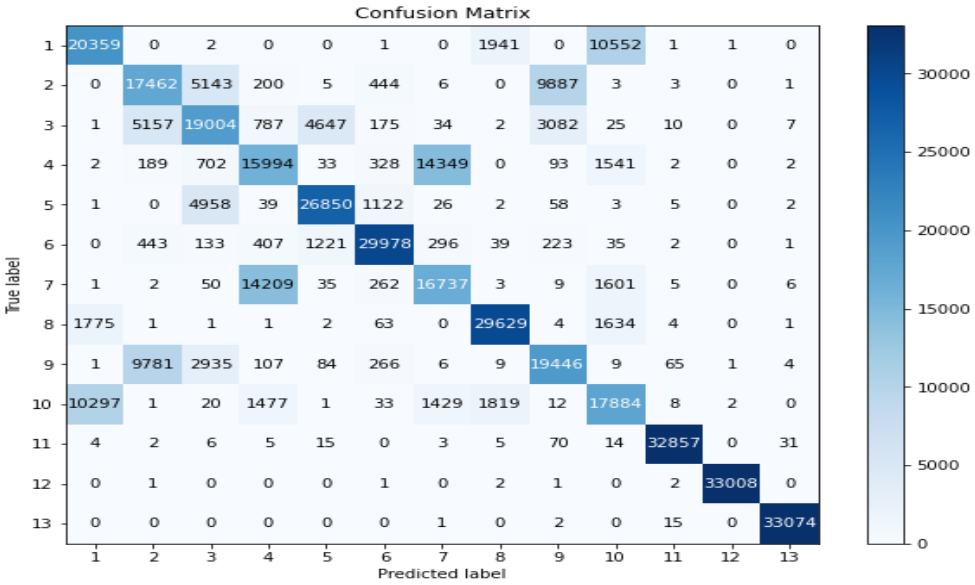


Figure 4.11: Confusion Matrix Showing CART's Predicted against True Labels

The K-Means clustering algorithm was applied to the dataset yielding the results in Table 4.2, where precision, recall, and F1 score were calculated for each cluster against the true labels.

Table 4.2: Performance of the CART algorithm in classifying the data

Class	Precision	Recall	F1 Score	Counts
0	**0.63**	**0.62**	**0.63**	**9814**
1	0.66	0.65	0.66	10010
2	0.73	0.73	0.73	10044
3	0.61	0.61	0.61	9909
4	0.66	0.66	0.66	9906
5	0.80	0.82	0.81	9863
6	0.66	0.65	0.66	9908
7	0.66	0.67	0.66	9738
8	0.71	0.71	0.71	9793
9	0.55	0.55	0.55	9963
10	0.99	0.99	0.99	9786
11	1.00	1.00	1.00	10003
12	1.00	1.00	1.00	9933

These metrics assess the quality of clustering; precision indicates the correctness of clustered instances, recall represents the actuality of true labels captured, while the F1 score gives a harmonic mean of both values. For example, label 10 (DrDoS_NTP) obtained very high scores (0.99) and thus denoted a well-done clustering effort, whereas label 9 (UDP-Lag) scored relatively poorly (0.55) and thus needs future improvement. Counts show the number of instances per label.

The algorithm measures are mapped as shown in Figure 4.11 which highlights the correlation between measuring evaluation metrics that are applied in the study. It demonstrates the dependence of these results on the performance measures such as precision, recall, F1-score and specificity. It points out that the metrics are interdependent, and present one straightforward understanding of the performance evaluation of classification.

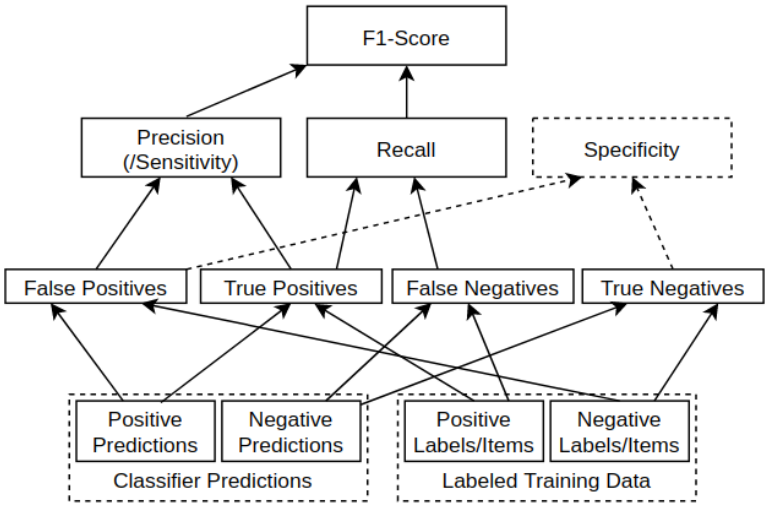


Figure 4.12: Classification Measure

4.3.3.2.1 Accuracy

Accuracy is the ultimate measure of a model's performance. It describes the number of correct predictions over all predictions. According to Kanstrén (2021), it is given as;

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where positive is the classification of an instance to a class that the classifier was trying to classify it to and negative is the classification of a member as being not a member of

the class to which we were trying to associate it. The plain CART model achieved an accuracy score of 0.74 (74%) representing 428872 of the data instances.

4.3.3.2.2 Precision

Precision measures the number of positive predictions made correctly (true positives). Kanstrén (2021) notes its formula as;

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

CART averaged a precision score of 0.74 (74%) in classifying all CICDDOS2019 Dataset attack types.

4.3.3.2.3 Recall/Sensitivity

The recall represents the number of positive cases that the classifier correctly predicted, over all positive cases in the data. Kanstrén (2021) gives its formula as;

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Again, CART posted a recall score of 0.73 (73%) as an average for all attack types in the dataset.

4.3.3.2.4 F1-Score

F1-Score denotes the combination of precision and recall scores mentioned in the preceding sections. It provides a harmonic mean of the values generated from the two measures. Kanstrén (2021) gives its formula as;

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Therefore, the CART averaged an F1-Score of 0.74 (74%) for all attack types.

CART allows the model to be boosted by eXtreme Gradient Boosting (XGBoost) which is relatively fast and does not affect the computational power needs of the CART model. Boosting the model achieves a better score of 75% for all the targeted measures as shown in Figure 4.12. Boosting the model aids in improving its performance while also reducing the risk of overfitting.

	precision	recall	f1-score	support
1	0.63	0.69	0.66	32857
2	0.55	0.53	0.54	33154
3	0.63	0.63	0.63	32931
4	0.50	0.49	0.50	33235
5	0.84	0.87	0.86	33066
6	0.92	0.96	0.94	32778
7	0.52	0.55	0.53	32920
8	0.94	0.89	0.92	33115
9	0.62	0.61	0.62	32714
10	0.59	0.52	0.55	32983
11	1.00	1.00	1.00	33012
12	1.00	1.00	1.00	33015
13	1.00	1.00	1.00	33092
accuracy			0.75	428872
macro avg	0.75	0.75	0.75	428872
weighted avg	0.75	0.75	0.75	428872

Figure 4.13: Performance of XGBoosted CART Model

4.3.4 Development of the Hybrid K-Means and CART Model

This hybrid model is an integration of K-means clustering with CART. The design of this hybrid begins with making K-means operational. The number of clusters was obtained through the Elbow Method using inertia, and K-Means was applied to effectuate the grouping of the network traffic on the basis of similarity. These were converted to labels for clusters and were then added as a new feature into the dataset to provide more context for the classification phase. K-means being unsupervised assists the hybrid model uncover hidden structures in data, which then inform the supervised classification.

The final dataset with the original cluster labels has been used to train the CART implemented with XGBoost. The primary reason for the selection of XGBoost over other implementations of gradient boosting was its working capability with huge data and its improved classification accuracy. The tree is embedded with XGBoost to differentiate between the benign and malicious flows given other variables such as flow duration, average packet size, and protocol type but also clusters added. The incorporation of K-Means allows combining the advantages of both algorithms. K-Means provides patterns and anomalies in data, and CART classifies these patterns accurately. The combination provides a solution to the limitations associated with a single technique, thus providing a robust and flexible solution to application-layer DDoS attack detection.

4.3.5 Effectiveness of the Hybrid K-Means and CART Model

The assessment of the new model was done using the same matrices used in appraising the performance of CART. As such, Accuracy, Precision, Recall, and F1-Score were measured. Table 4.3 shows the individual performances for each class of data analysed.

Table 4.3: Measures of performance for the Hybrid K-Means and CART model for each class of data in the dataset

Class	Precision	Recall	F1-score	Support
0	0.57	0.78	0.65	9814
1	0.65	0.88	0.75	10010
2	0.79	0.75	0.77	10044
3	0.70	0.64	0.67	9909
4	0.70	0.73	0.71	9906
5	0.89	0.83	0.86	9863
6	0.67	0.80	0.73	9908
7	0.66	0.64	0.65	9738
8	0.84	0.66	0.74	9793
9	0.80	0.39	0.53	9963
10	0.99	0.98	0.99	9786
11	1.00	1.00	1.00	10003
12	0.99	1.00	0.99	9933

It can be clearly seen from the results in Table 4.3 that the hybrid model performed very well over most of the classes. Near-perfect precision, recall, and F1-scores (0.99–1.00) were reached with classes 10 (DrDoS_NTP), 11 (BENIGN), and 12 (WebDDoS) indicating high capability in detection. This strongly suggests that the hybrid model is quite good at identifying such types of traffic whether it is benign or malicious. However,

for class 9 (UDP-Lag), recall 0.39 and F1 score 0.53 indicate poor performance as many instances of this class have been misclassified. This may be due to inherent complexity or resemblance of UDP-Lag traffic with that of other classes.

Majority of classes, such as 1 (DrDoS_LDAP), 2 (DrDoS_SNMP), and 5 (DrDoS_MSSQL), have their hybrid model achieving balanced precision with recall by F1 scores that range between 0.71 and 0.86. The model is able to detect and classify these kinds of attack and at the same time exhibiting measurable less false positive and false negative counts. That in support column shows the number of instances per class proving the dataset has indeed been well balanced contributing to the robustness of the model.

On an average basis, hybrid model achieved a precision of 0.79, recall (sensitivity) score of 0.78, and F1-score of 0.78.5 with overall accuracy of 0.78. Such averaged metrics mean the model does well in performing across the dataset and achieving such a good balance in trade-off between precision and recall. There is a needed improvement of the model to detecting certain classes like UDP-Lag even though it excelled in identifying most of the attack types and benign traffic. These results thus put emphasis on further refining one's model to counter these challenges and thus boast overall effectiveness.

An evaluation of hybridisation effects was done by performing a comparative analysis of the standalone CART model and the hybrid K-Means-CART model. Table 4.4 shows that there is uniform performance improvement in most classes when clustering information is introduced into classification process and this manifests the value of hybrid learning in performing complex intrusion detection tasks.

In most class scenarios, the hybrid model shows similar or better precision, recall, and F1-scores. In the case of class 1, DrDoS_LDAP, the hybrid model performed better in the

sense of improved recall 0.65 to 0.88 and F1-score 0.66 to 0.75 accords. For class 6, DrDoS_UDP, recall was improved from 0.65 to 0.80, and F1-score from 0.66 to 0.73. For class 9, UDP-lag, upon improvement in precision from 0.55 to 0.80, a drop in recall from 0.55 to 0.39 showed the conflicting characteristics. All in all, the hybrid models perform better, especially when detecting complex attack types with a reasonable level of accuracy on benign traffic.

Table 4.4: Comparison between a pure CART model and the Hybrid with K-means

Code	Label	Pr(o)	Pr(Hm)	Rc(Or)	Rc(Hm)	F1(Or)	F1(Hm)
0	Syn	0.63	0.57	0.62	0.78	0.63	0.65
1	DrDoS_LDAP	0.66	0.65	0.65	0.88	0.66	0.75
2	DrDoS_SNMP	0.73	0.79	0.73	0.75	0.73	0.77
3	DrDoS_SSDP	0.61	0.70	0.61	0.64	0.61	0.67
4	DrDoS_NetBIOS	0.66	0.70	0.66	0.73	0.66	0.71
5	DrDoS_MSSQL	0.80	0.89	0.82	0.83	0.81	0.86
6	DrDoS_UDP	0.66	0.67	0.65	0.80	0.66	0.73
7	TFTP	0.66	0.66	0.67	0.64	0.66	0.65
8	DrDoS_DNS	0.71	0.84	0.71	0.66	0.71	0.74
9	UDP-lag	0.55	0.80	0.55	0.39	0.55	0.53
10	DrDoS_NTP	0.99	0.99	0.99	0.98	0.99	0.99
11	BENIGN	1.00	1.00	1.00	1.00	1.00	1.00
12	WebDDoS	1.00	0.99	1.00	1.00	1.00	0.99

Where, Pr(0), Rc(o), and F1(o) stand for precision, recall and F1 scores of the original CART model, Pr(Hm), Rc(Hm) and F1(Hm) denote precision, recall, and F1 scores of the hybrid model.

4.4 Chapter Summary

This chapter introduced and presented the results of the proposed hybrid K-Means-CART intrusion detection model. The dimensionality reduction and preliminary data preparation was done to reduce the high dimensionality and overlapping of DDoS application-layer traffic, which was not easy to process using unsupervised clustering. The K-Means algorithm performed moderately as a way of showing the limitation it has, when used on its own, on the complex and overlapping data sets. The CART classifier had a better classification performance, with an overall accuracy of 74, and a high accuracy of 75 after XGBoost boosting.

Combining K-Means clustering with CART created a hybrid model that performed better than the separate classifiers reaching an overall accuracy of 78 with equal precision and recall in the majority of attack classes. Substantial gains were made in several classes of DDoS, but some were harder to overcome like UDP-Lag due to the similarity of the traffic. Generally, the results attest that the combination of unsupervised and supervised learning methods is more effective in the identification of application-layer DDoS attacks and directly responds to the aims of the study.

CHAPTER FIVE

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of Findings

The study aimed to analyse the available intrusion detection methodologies in application-layer DDoS attacks, create a hybrid model of intrusion detection and evaluate it on the CICDDoS2019 dataset.

Comparing them to Objective 1, which aimed at analysing the strengths and weaknesses of the currently existing methods of intrusion detection, it is shown that standard IDS systems, like ID3, Random Forest, Naive Bayes, and Logistic Regression, do not perform consistently in instances involving high-dimensional and complex application-layer traffic. Whereas ID3 has attained a fairly good F1-score of 69, Naive Bayes and Logistic Regression scored dismally, at 5% and 4% respectively. This outcome points at the weakness of models whose assumptions are based on the independence of features or the linearity of decision boundaries, which was aimed at testing unsupervised learning in terms of application-layer DDoS detection, K-Means clustering with PCA scored moderately, with a homogeneity score of 50.76, completeness score of 51.95 and a V-measure of 51.35. Despite the compromised accuracy of clustering due to overlapping clusters and outliers, K-Means was useful in the sense that it revealed useful information about the inherent structure and complexity of the data.

In regards to Objective 3, which was to design and test a hybrid intrusion detection model, the performance of K-Means clustering with CART classification showed an improvement that was quantifiable. The CART classifier had a total accuracy of 74% that

went up to 75% with boosting. The hybrid model obtained a total accuracy of 78 when the clustering information was included with significant improvements of DrDoSLDAP (66% to 75%), DrDoSUDP (66% to 73%), and DrDoS_MSSQL (81% to 86%).

The results, on the whole, support the idea in that unsupervised and supervised learning methods can help to increase the detection rate of complex application-layer DDoS attacks, but some issues still exist with certain traffic classes, such as UDP-Lag.

5.2 Contribution of the Study

This study contributes to the research in intrusion detection and cybersecurity in a number of ways. Firstly, it suggests and experimentally validates a hybrid K-Means-CART intrusion detection model that is application-layer-specifically designed to be applied to DDoS attacks in application layer. In contrast to conventional single-algorithm models, the hybrid model exploits the use of clustering-based feature extraction in order to enhance the performance of supervised classification.

Secondly, the study gives a class-level performance analysis based on the CICDDoS2019 dataset, where each category of attacks is most effectively studied with the help of hybrid learning, which has remained challenging for a long time. This would help to make the IDS performance more refined than the aggregate measures of accuracy.

Thirdly, the study identifies the practical drawbacks of popular classifiers including Naive Bayes and Logistic Regression with high dimensional and overlapping traffic scenarios, which supports the necessity of adaptive and hybrid detection methods.

Lastly, the research provides empirical evidence to show that unsupervised and supervised learning are an efficient combination that can be used in enhancing the accuracy of intrusion detection in a contemporary network.

5.3 Conclusions

The study examined how an intrusion detection system can be designed and evaluated using the CICDDoS2019 dataset using K-Means clustering and CART classification. It was found through reviewing the literature that individual and hybrid machine learning methods have been utilized in previous research on IDS with varying levels of success depending on the dataset and the context of the attack.

It was experimentally demonstrated that CICDDoS2019 dataset is a challenge because it is high-dimensional, has overlapping attack patterns, and classes have similarities. Although the K-Means clustering alone performed moderately, it was also useful in the disclosure of latent data structures that were useful in further classification. CART showed good classification performance with an accuracy of 74, a position that improved towards an increase in boosting.

The hybrid K-Means-CART was superior to its component algorithms with a 78 percent accuracy and better recall on some of the complex attack classes. Yet, the fact that some types of traffic, including UDP-Lag, have performance degradation under hybridisation demonstrates that the hybridisation in itself is not the solution to all the detection issues. Altogether, the study verifies that hybrid learning models should be regarded as a more effective and flexible approach to use in detecting application-layer DDoS intrusions as compared to the single-technique.

5.4 Recommendations

This section presents recommendations derived from the findings of the study, focusing on practical and strategic implications.

5.4.1 Policy Recommendations

According to the results of this research, policy-makers and regulatory agencies ought to encourage the use of hybrid methods of intrusion detection, which combine the use of unsupervised and supervised learning. These models have shown higher effectiveness in identifying sophisticated application-layer DDoS attacks and have special application when it comes to securing critical national infrastructure.

The use of modern and high-dimensional data, like the CICDDoS2019, to evaluate and certify the IDS in question should also be encouraged by cybersecurity policies. There would be standardized benchmarking models which would help in making sure that installed IDS solutions are exercised on real and dynamic attacks.

Also, the policy frameworks must help sustain a commitment to invest in data-driven cybersecurity research such as funding towards ongoing retraining of models, cross-sector coordinated efforts, and capacity building to enhance national and organizational cyber resiliency.

5.4.2 Industry practitioners Recommendations.

Network administrators and security operations teams working in the industry should take into consideration implementing hybrid IDS designs using clustering-based preprocessing as well as supervised classification. This has proven to work better at detection of various types of application-layer DDoS attacks.

The use of traditional classifiers like the Naive Bayes or Logistic Regression is discouraged since practitioners are expected to apply it only in simple traffic scenarios. Rather, feature extraction, dimensionality reduction, or clustering methods can happen to be much more effective in improving the performance of detection.

Class imbalance, computational requirements and constant performance monitoring should also be put into consideration when deploying IDS. In order to maintain the effectiveness of detection in real world conditions, three techniques are necessary; resource planning, optimization of features selected and periodic retraining of models.

Lastly, since the hybrid model is less aggressive with respect to some type of attack and limited in terms of evaluation efforts, organizations to verify IDS solutions on multi-datasets or live traffic setup and investigate adaptive learning approaches to enhance resilience to both emerging and zero-day attacks.

REFERENCES

Aamir, M., & Zaidi, S. M. A, (2019). Clustering based semi-supervised machine learning for DDoS attack classification. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2019.02.003

Abushark, Y. B., Khan, A. I., Alsolami, F., Almalawi, A., Alam, M. M., Agrawal, A., Kumar, R., & Khan, R. A. (2022). Cyber Security Analysis and evaluation for Intrusion Detection Systems. *Computers, Materials & Continua*, *72*(1), 1765–1783. https://doi.org/10.32604/cmc.2022.025604

Agate, V., D'Anna, F. M., De Paola, A., Ferraro, P., Re, G. L., & Morana, M. (2022). A behaviour-based intrusion detection system using ensemble learning techniques. *ITASEC*. https://iris.unipa.it/retrieve/342ffb51-84fd-47b9-a5eb-ae8bd10b60be/TOC_preface_paper_ITASEC2022.pdf

Akamai, (2019). State of the Internet - Security: DDoS and Application Attacks Report. 1–24. https://www.akamai.com/us/en/resources/our-thinking/state-of-the-internet-report/global-state-of-the-internet-security-DDoS-attack-reports.jsp

Akkaya, B., & Çolakoğlu, N. (2019). Comparison of multi-class classification algorithms on early diagnosis of heart diseases.

Alsoufi, M., A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-based intrusion detection systems in IOT using Deep Learning: A Systematic Literature Review. *Applied Sciences*, *11*(18), 8383. https://doi.org/10.3390/app11188383

Alturki, R. (2021). Research Onion for smart IOT-enabled mobile applications. *Scientific Programming*, *2021*, 1–9. https://doi.org/10.1155/2021/4270998

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: evolution, applications, and research opportunities. Journal of Internet Services and Applications, 9(1). https://doi.org/10.1186/s13174-018-0087-2

Canadian Institute for Cybersecurity. (2019). *DDoS Evaluation Dataset (CIC-DDoS2019)*. University of New Brunswick est.1785. Retrieved February 24, 2022, from https://www.unb.ca/cic/datasets/DDoS-2019.html

Chen, W., Chen, S., Zhang, H., & Wu, T. (2017, November 1). *A hybrid prediction model for type 2 diabetes using K-means and decision tree*. IEEE Xplore. https://doi.org/10.1109/ICSESS.2017.8342938

Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage Publications, Inc.

Firdaus, D., Munadi, R., & Purwanto, Y. (2020, December 1). *DDoS Attack Detection in Software Defined Network using Ensemble K-means++ and Random Forest*. IEEE Xplore. https://doi.org/10.1109/ISRITI51436.2020.9315521

Ford, V., & Siraj, A. (2014, October). Applications of machine learning in cyber security. In *Proceedings of the 27th international conference on computer applications in industry and engineering* (Vol. 118). Kota Kinabalu, Malaysia: IEEE Xplore.

Fränti, P., & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? Pattern Recognition, 93, 95–112. https://doi.org/10.1016/j.patcog.2019.04.014

Guid, A. G. (n.d.). Writing Res Proposal and Willis Vuko Oso.

Hajj, S., El Sibai, R., Bou Abdo, J., Demerjian, J., Makhoul, A., & Guyeux, C. (2021). Anomaly-based Intrusion Detection Systems: The requirements, methods, measurements, and datasets. *Transactions on Emerging Telecommunications Technologies*, *32*(4). https://doi.org/10.1002/ett.4240

IBM. (n.d.). *Resource utilization and performance*. https://www.ibm.com/docs/en/informix-servers/14.10?topic=basics-resource-utilization-performance

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *Unsupervised Learning*. 503–556. https://doi.org/10.1007/978-3-031-38747-0_12

Jia, B., Huang, X., Liu, R., & Ma, Y. (2017). A DDoS Attack Detection Method Based on Hybrid Heterogeneous Multiclassifier Ensemble Learning. Journal of Electrical and Computer Engineering, 2017. https://doi.org/10.1155/2017/4975343

Kanstrén, T. (2021, May 19). *A look at precision, recall, and F1-score*. Medium. Retrieved May 17, 2022, from https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*, *2*(1). https://doi.org/10.1186/s42400-019-0038-7

Krzywinksi, M., & Altman, N. (2017). Correction: Corrigendum: Classification and regression trees. *Nature Methods*, *14*(9), 928-928.

Krzywinski, M., & Altman, N. (2017). Classification and regression trees. Nature Methods, 14(8), 757–758. https://doi.org/10.1038/nmeth.4370

Kumar, G. (2016). Denial of service attacks – an updated perspective. *Systems Science & Control Engineering*, *4*(1), 285–294. https://doi.org/10.1080/21642583.2016.1241193

Kumar, V., Kumar, A., Garg, S., & Payyavula, S. R. (2022). Boosting algorithms to identify distributed denial-of-service attacks. *Journal of Physics: Conference Series*, *2312*(1), 012082. https://doi.org/10.1088/1742-6596/2312/1/012082

Laftah Al-Yaseen, W., Ali Othman, Z., Nazri, A., & Zakree, M. (2015). Hybrid modified-means with C4. 5 for intrusion detection systems in Multiagent Systems. *The Scientific World Journal*, *2015*.

Lee, K., Kim, J., Kwon, K. H., Han, Y., & Kim, S. (2008). DDoS attack detection method using cluster analysis. *Expert Systems with Applications*, *34*(3), 1659–1665. https://doi.org/10.1016/j.eswa.2007.01.040

Malek, Z. S., Trivedi, B., & Shah, A. (2020). User behavior pattern -signature based intrusion detection. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. https://doi.org/10.1109/worlds450073.2020.9210368

Obaid, H. S., & Abeed, E. H. (2020). DoS and DDoS attacks at OSI layers. *International Journal of Multidisciplinary Research and Publications*, *2*(8), 1-9.

Radev, G. (2019). Cyber Threats for Modern Economy. In *Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE)* (pp. 192-197). International Conference on Application of Information and Communication Technology and Statistics and Economy and Education (ICAICTSEE).

Radoglou-Grammatikis, P. I., & Sarigiannidis, P. G. (2019). An Anomaly-Based Intrusion Detection System for the Smart Grid Based on CART Decision Tree. 2018 Global Information Infrastructure and Networking Symposium, GIIS 2018, October, 1–5. https://doi.org/10.1109/GIIS.2018.8635743

Rashid, A., Gardiner, J., Green, B., & Craggs, B. (2019). Everything is awesome! or is it? cyber security risks in critical infrastructure. *Critical Information Infrastructures Security*, 3–17. https://doi.org/10.1007/978-3-030-37670-3_1

Raval, U. R., & Jani, C. (2016). Implementing & improvisation of K-means clustering algorithm. *International Journal of Computer Science and Mobile Computing*, *5*(5), 191-203.

Reddy, G. S., & Chittineni, S. (2021). Entropy based C4.5-SHO algorithm with information gain optimization in Data Mining. *PeerJ Computer Science*, *7*. https://doi.org/10.7717/peerj-cs.424

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. da, & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLOS ONE*, *14*(1). https://doi.org/10.1371/journal.pone.0210236

Salih, A. A., & Abdulazeez, A. M. (2021). Evaluation of classification algorithms for Intrusion Detection System: A Review. *Journal of Soft Computing and Data Mining*, *2*(1), 31–40. https://doi.org/10.30880/jscdm.2021.02.01.004

Saunders, M., Lewis, P., & Thornhill, A. (2023). *Research Methods for Business Students* (9th ed.). Pearson.

Scikit Learn. (n.d., a). *2.3. clustering*. scikit. Retrieved May 14, 2022, from https://scikit-learn.org/stable/modules/clustering.html#k-means

Scikit-Learn. (n.d., b). *1.10. decision trees*. scikit-lean. Retrieved September 12, 2022, from https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

Semantic Scholar. (n.d.). *Computational learning theory*. Engati. https://www.engati.com/glossary/computational-learning-theory

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. *2019 International Carnahan Conference on Security Technology (ICCST)*. https://doi.org/10.1109/ccst.2019.8888419

Sharma, R., Sharma, K., & Khanna, A. (2020). Study of supervised learning and unsupervised learning. *International Journal for Research in Applied Science and Engineering Technology*, *8*(6), 588-593. http://doi.org/10.22214/ijraset.2020.6095

She, C., Wen, W., Zheng, K., & Lyu, Y. (2016). Application-Layer DDoS Detection by K-means Algorithm. 50(Iceeecs), 75–78. https://doi.org/10.2991/iceeecs-16.2016.16

Singh, A., & Jain, A. (2018). Study of cyber attacks on cyber-physical system. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3170288

Singhal, S., Medeira, P. A., Singhal, P., & Khorajiya, M. (2020). Detection of app-DDOS attacks using big data technologies. Journal of Discrete Mathematical Sciences and Cryptography, 23(2), 563–571. https://doi.org/10.1080/09720529.2020.1729505

Smys, S., Basar, A., & Wang, H. (2020). Hybrid intrusion detection system for internet of things (IOT). *December 2020*, *2*(4), 190–199. https://doi.org/10.36548/jismac.2020.4.002

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2020). Efficient processing of deep neural networks. *Synthesis Lectures on Computer Architecture*, *15*(2), 1–341. https://doi.org/10.2200/s01004ed1v01y202004cac050

Thakkar, A., & Lohiya, R. (2021). A survey on Intrusion Detection System: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, *55*(1), 453–563. https://doi.org/10.1007/s10462-021-10037-9

Torabi, M., Hashemi, S., Saybani, M. R., Shamshirband, S., & Mosavi, A. (2018). A Hybrid clustering and classification technique for forecasting short-term energy consumption. *Environmental Progress & Sustainable Energy*, *38*(1), 66–76. https://doi.org/10.1002/ep.12934

Tripathi, N., & Hubballi, N. (2021). Application layer denial-of-service attacks and Defense Mechanisms. *ACM Computing Surveys*, *54*(4), 1–33. https://doi.org/10.1145/3448291

Valle, F. (2021, April 8). *V measure: An homogeneous and complete clustering*. Medium. Retrieved May 17, 2022, from https://towardsdatascience.com/v-measure-an-homogeneous-and-complete-clustering-ab5b1823d0ad

Varpio, L., Paradis, E., Uijtdehaage, S., & Young, M. (2019). The distinctions between theory, theoretical framework, and Conceptual Framework. *Academic Medicine*, *95*(7), 989–994. https://doi.org/10.1097/acm.0000000000003075

Vishwakarma, R., & Jain, A. K. (2019). A survey of DDoS attacking techniques and defence mechanisms in the IOT network. *Telecommunication Systems*, *73*(1), 3–25. https://doi.org/10.1007/s11235-019-00599-z

Xiao, J., Tian, Y., Xie, L., Jiang, X., & Huang, J. (2020). A Hybrid Classification Framework Based on Clustering. *IEEE Transactions on Industrial Informatics*, *16*(4), 2177–2188. https://doi.org/10.1109/tii.2019.2933675

Yerriswamy T, & Gururaj Murtugudde. (2020). Study of Evolutionary Techniques in the field of Network Security. *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*. https://doi.org/10.1109/icstcee49637.2020.9277082

Yi Yi Aung, & Myat Myat Min. (2018). *Hybrid Intrusion Detection System Using K-Means and Classification and Regression Trees Algorithms*. https://doi.org/10.1109/sera.2018.8477203

Yzzogh, H., & Benaboud, H. (2025). Flooding distributed denial of service detection in software-defined networking using k-means and naïve Bayes. *International Journal of Electrical and Computer Engineering (IJECE)*, *15*(1), 817. https://doi.org/10.11591/ijece.v15i1.pp817-826

Zhou, Z. H. (2021). *Machine learning*. Springer Nature.

APPENDICES

Appendix 1: NACOSTI Research Permit

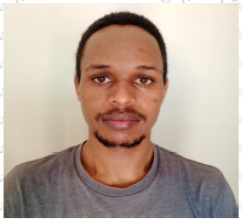
REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 676260

Date of Issue: 21/March/2024

RESEARCH LICENSE



This is to Certify that Mr.. Victor Kipngetich Cheruiyot of Moi University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Uasin-Gishu on the topic: A HYBRID INTRUSION DETECTION MODEL FOR APPLICATION LAYER DDOS ATTACKS BASED ON K-MEANS AND CART ALGORITHMS for the period ending : 21/March/2025.

License No: NACOSTI/P/24/34117

676260

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix 2: Plagiarism Check

SR867

ISO 9001:2019 Certified Institution

THESIS WRITING COURSE

PLAGIARISM AWARENESS CERTIFICATE

This certificate is awarded to

CHERUIYOT K. VICTOR

MS/IT/7784/20

In recognition for passing the University's plagiarism

Awareness test for Thesis **entitled: A HYBRID INTRUSION DETECTION MODEL FOR APPLICATION LAYER DDOS ATTACKS BASED ON K-MEANS AND CART ALGORITHMS with** similarity index of 5% and striving to maintain academic integrity.

Word count:21200

Awarded by

Prof. Anne Syomwene Kisilu
CERM-ESA Project Leader Date: 15/08/2025