Arab American University

Faculty of Graduate Studies

Department of Natural, Engineering and Technology Sciences

Master Program in Cybercrimes and Digital Evidence Analysis

Detecting and Permitting Legitimate Traffic from IPs with Malicious Reputation

Bashar Sadi Mustafa Jaber

201920307

Supervision Committee:
Dr. Osama Mansour
Dr. Majdi Owda
Dr. Chatrine Qwaider

This Thesis Was Submitted in Partial Fulfillment of the Requirements for the Master Degree in Cybercrime and Digital Evidence Analysis.

Palestine, September / 2024

Arab American University
Faculty of Graduate Studies
Department of Natural, Engineering and Technology Sciences
Master Program in Cybercrimes and Digital Evidence Analysis

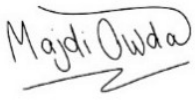
Thesis Approval

Detecting and Permitting Legitimate Traffic from IPs with Malicious Reputation

Bashar Sadi Mustafa Jaber
201920307

This thesis was defended successfully on September 21, 2024, and approved by:

Thesis Committee Members:

Name	Title	Signature
1- Dr. Osama Mansour	Main Supervisor	
2- Dr. Majdi Owda	Members of Supervision Committee	
3- Dr. Chatrine Qwaider	Members of Supervision Committee	

Palestine, September / 2024

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

Student Name: Bashar Sadi Mustafa Jaber
Student ID: 201920307
Signature: Bashar Sadi Mustafa Jaber

Date of Submitting the Final Version of the Thesis: 4 . 2 - 2025

Dedication

To the late yet present, who devoted his life for us and passed away before witnessing the fruits of his labor in us... my dear father.

To the jewel of the crown and the essence of life, who carried me through weakness and exerted every effort to nurture me with love and support... my beloved mother.

To the companion in struggle, who walked alongside me in life's journey and supported me with unwavering love and steadfastness... my dear wife.

To the source of my joy and hope for the future, who has been my light and accompanied me with their support and innocence... my beloved children: Salma, Saadi, Ahmed, and Tayma.

To the pillar and support, the source of strength and encouragement, who have always been my backbone and supporters through every stage of my life... my siblings: Bilal, Ammar, Amer, Bayan, and Serin.

To the one who devoted his time and valuable guidance, playing a significant role in directing and supporting me throughout my academic journey... my mentor and supervisor, Dr. Osama Mansour.

To the martyrs, the brave prisoners, and the heroic wounded,

To you, Gaza,

I offer this dedication.

I dedicate this message to you, my mother, and to all of you, hoping that it will be a source of pride for you as you are a source of pride to me.

Bashar Sadi Mustafa Jaber

Acknowledgments

I want to express my deepest gratitude to everyone who supported me throughout the development of this thesis. First and foremost, I want to thank Dr. Osama Mansour, my supervisor, for his invaluable guidance, encouragement, and constructive feedback. His expertise and insights have been instrumental in shaping this work and provided clarity and direction during the research process. I extend heartfelt thanks to my colleagues and peers for their support, shared knowledge, and engaging discussions, which greatly enriched my understanding.

To my family, thank you for your unwavering support, patience, and encouragement throughout this journey. Your belief in me has been a constant source of strength. Finally, I am grateful to all those whose work I have cited and who contributed indirectly to my research. Also, I extend gratitude to the Arab American University, acknowledging the role of the faculty, department, research facilities, and program coordinator and committee for their guidance and support in completing this work. This thesis would not have been possible without the collective effort of this academic community. Thank you all for being part of this significant milestone in my life.

Bashar Sadi Mustafa Jaber

Detecting and Permitting Legitimate Traffic

from IPs with Malicious Reputation

Bashar Sadi Mustafa Jaber

Supervision Committee:
Dr. Osama Mansour
Dr. Majdi Owda
Dr. Chatrine Qwaider

Abstract

In the current digital environment, detecting malicious activities within network traffic has become paramount for ensuring cybersecurity. This thesis introduced the Normal Traffic Detection (NTD) model, which differentiates between normal and abnormal IP address traffic. Drawing upon the collaborative strengths of Support Vector Machines (SVM), Sequential Artificial Neural Networks (ANN), and Decision Trees, it stands for Network Traffic as a beacon of innovation in anomaly detection. The methodology of the NTD model lies in its sophisticated approach to analyzing incoming traffic data. Initial scrutiny involves capturing and analyzing the nuances of the traffic, with particular emphasis on behavior. This data is scrutinized meticulously against an Anomaly Behavior Database (ABD), a repository teeming with previously identified aberrations in network behavior. Any matches with entries within the ABD are promptly flagged as malicious, warranting further investigation. However, not all traffic bears the hallmark of known anomalies. For those instances that evade identification within the ABD, NTD embarks on a journey of sequential classification. The traffic is subjected to the discerning scrutiny of SVM, Decision Trees, and ANN, each algorithm meticulously parsing through the data in pursuit of anomalous patterns. Upon detection of malicious intent, the traffic is promptly logged into the ABD, enriching its repository with newfound insights. The efficacy of NTD transcends mere theoretical conjecture; empirical validation using real-world cybersecurity datasets serves as a litmus test for its prowess. Comparative analyses against traditional single-algorithm methods reveal a resounding victory for NTD, boasting superior metrics. Whether measured by the F1 score, precision, or recall, NTD emerges as the undisputed champion, heralding a new era in network traffic anomaly detection. Beyond its immediate applications in cybersecurity, NTD's implications extend far and wide. Its robust performance underscores its potential to fortify defenses across various domains, from financial institutions safeguarding sensitive transactions to governmental agencies protecting critical infrastructure. The ripple effects of NTD's deployment resonate throughout the digital ecosystem, engendering a newfound sense of confidence in the face of ever-evolving cyber threats.

Keywords: Network Traffic Detection (NTD), Anomaly Detection, Support Vector Machines (SVM), Machine Learning, Cybersecurity.

Table of Contents

#	Title	Page
Declaration		I
Dedication		II
Acknowledgments		….III
Abstract		IV
List of Tables		vi
List of Figures		vii
List of Definitions of Abbreviations		viii
Chapter One: Introduction		1
1.1 Background		1
1.2 Problem Statement		3
1.3 Cyberspace		6
1.4 Cyberattack		8
1.5 Cybersecurity		9
1.6 attack techniques and cybersecurity		10
1.6.1 Passive And Active Attacks		11
1.6.2 Targeted and Untargeted Attacks		11
1.7 Summary		13
Chapter Two: Literature Review		15
2.1 Mechanisms To Detect Attacks		15
2.2 Signature-Based Detection (SBD):		15
2.3 Machine Learning Techniques		16
2.4 IPS And IDS		18
2.5 Intrusion Detection System Ids		18
2.6 Intrusion Prevention System Ips		22
2.7 False Positive Detection		24
2.8 Hypothesis		26
2.9 Summary		26
Chapter Three: Methodology		28
3.1 Description Of Proposed Model Architecture		29
3.1.1 Supervised Vector Machine (Svm)		31
3.1.2 Decision Tree		32

3.1.3 Artificial Neural Networks (Ann) 34
3.2 Data Sampling: Collection And Gathering 35
3.2.1 Features Description 37
3.3 Data Preprocessing 39
3.3.1 Data Cleaning 40
3.3.2 Data Transformation 41
3.3.2.1 Label Encoding 41
3.3.2.2 Scaling Of Numerical Values 41
3.3.2.3 Feature Engineering 41
3.3.3 Splitting Data 41
3.3.4 Encoding Categorical Variables 41
3.4 Model Building 42
3.5 Using Ensemble Learning In Model Development 42
3.5.1 Improve Accuracy 43
3.5.2 Reducing Bias 43
3.5.3 More Powerful 43
3.5.4 Robustness And Reliability 43
3.6 Model Evaluation 43
3.7 Summary 44
Chapter Four: Exploratory Data Analysis 46
4.1 Data Preprocessing 46
4.2 Dataset Properties 46
4.3 Traffic Distribution 46
4.3.1 Distribution Of Traffic Type(Class) 47
4.3.2 Classification Of Data Into Two Categories 47
4.3.3 Feature Distribution In Dataset 48
4.4 Summary 51
Chapter Five: Results 52
5.1 Processing Of Data In The Dataset 52
5.2 Applying Each Algorithm Separately 52
5.2.1 Svm Algorithm 52
5.2.2 Decision Tree Algorithm. 54
5.3 Normal Traffic Detection Model (Ntd) 58
5.4 Evaluation Of Proposed Solution 59

5.5 Evaluation Metrics For Ntd Model Over Dataset 1 60
5.5.1 Confusion Matrix 60
5.5.2 Comparison Between Algorithms And Ntd Over Dataset 1 60
5.6 Summary 64
5.7 Future Work 65
5.8 Conclusion 66
References 67
ملخص 75

List of Tables

Table #	Title of Table	Page
Table 1.1 Cyberspace Domains		7
Table 1.2 List Of Attack Methods		13
Table 2.1: Result Comparison (Lansley, 2020)		17
Table 2.2: Evaluation And Time Elapsed Of The Proposed Model.		24
Table 3.1: Data Files Of Nsl-Kdd		37
Table 3.2 Features Description Of The Nls-Kdd Dataset		37
Table 4.1 Attack Type Distribution		47
Table 4.2 Classification Of Traffic Normal/Attack		47
Table 5.1 Confusion Matrix For Svm		53
Table 5.2 Evaluation Metrics For Svm Algorithm		53
Table 5.3 Confusion Matrix For Decision Tree		54
Table 5.4: Evaluation Metrics For Decision Tree		55
Table 5.5 Confusion Matrix For Ann Algorithm		56
Table 5.6 Evaluation Metrics For The Ann Algorithm		56
Table 5.7 Confusion Matrix Of Ntd		60
Table 5.8 Comparison Of Results Of All Algorithms Over Dataset 1		60
Table 5.9 Comparison Of Results Of All Algorithms Over Dataset 1		60
Table 5.10: Confusion Matrix For Svm Over Dataset2		61
Table 5.11 Evaluation Metrics Of Svm Over Dataset2		62
Table 5.12 Confusion Matrix For Decision Tree		62
Table 5.13 Evaluation Metric Of Decision Tree Over Dataset 2		62
Table 5.14 Confusion Matrix For Ann Over Dataset2		63
Table 5.15 Evaluation Metrics Of Ann Over Dataset 2		63
Table 5.16 Confusion Matrix Of Ntd Over Dataset 2		63
Table 5.17 Results Of Ntd Over Dataset2		64
Table 5.18 Comparison Table Of The Evaluation Metrics Over Dataset 2		64

List of Figures

Figure #	Title of Figure	Page
Figure 1.1:	Internet Access Through ISP	4
Figure 1.2:	Attack Categories And Types	11
Figure 2.1:	Basic Intrusion Detection System Design	19
Figure 2.2:	Proposed Solution Design	20
Figure 2.3:	Wsn Model Of Ids	21
Figure 2.4:	Design Of Ids Ips System	23
Figure 2.5:	Framework Of The Proposed Model Soinn	24
Figure 3.1:	Machine-Learning Methodology Of The Proposed Solution	28
Figure 3.2:	Ntd Model Flowchart	30
Figure 3.3	Linear Svm	31
Figure 3.4:	Decision Tree	34
Figure 4.1:	Label Distribution Of Traffic	46
Figure 4.2	Distribution Of Attack Types	48
Figure 4.3	Features Distribution In Dataset	49
Figure 4.4	Class Distribution On Srv_Rerror_Rate And Labels	50
Figure 4.5	Traffic Label In Terms Of Protocol Type And Volume	50
Figure 5.1	Scores Of Evaluation Metrics Of The Decision Tree.	55
Figure 5.2	Accuracy In Ann Algorithm For Training And Validation	57
Figure 5.3	Percentage Loss For Ann Algorithm	58
Figure 5.4	Ntd Model Algorithm	59

List of Definitions of Abbreviations

Abbreviations Title

No.	Term	Abbreviation / Definition
1	3G	Third Generation
2	4G	Fourth Generation
3	5G	Fifth Generation
4	ABD	Abnormal Behavior Database
5	ANN	Artificial Neural Network
6	API	Application Programming Interface
7	APT	Advanced Persistent Attack
8	AIBD	Artificial Intelligence-Based detection
9	CNN	Convolutional Neural Networks
10	CSCA	Compact Sine Cosine Algorithm
11	DDOS	Distributed Denial of Service attack
12	DNN	Deep Neural Networks DNN
13	DNS	Domain Names System
14	DOS	Denial Of Service Attack
15	DT	Decision Tree
16	EDR	Endpoint Detection and Reporting
17	F1	Field of information retrieval and statistical analysis
18	FN	False Negative
19	FP	False Positive
20	GRU	Gated Recurrent Unit
21	HTTP	Hyper Text Transfer Protocol
22	HTTPS	Secure Hyper Text Transfer Protocol
23	IDES	Intrusion Detection Expert System
24	IDS	Intrusion Detection System
25	IPS	Intrusion Prevention System
26	ISACA	Information Systems Audit and Control Association
26	ISP	Internet Service Provider
27	LTE	Long Term Evolution
28	NAT	Network Adress Translate
29	NERD	Network Entity Reputation Database
30	NIST	National Institute of Standards and Technology
31	NTD	Normal Traffic Detection
32	PMCSCA	Polymorphic Mutation Compact Sine Cosine Algorithm
33	OTX	Open Threat Exchange
34	Precision	Quality of positive predictions that model executes
35	PM	Polymorphic Mutation
36	Recall	Evaluates the model's capacity to recognize all relevant instances
37	SBD	Signature Based Database
38	SRI	Stanford Research Institute
39	SVM	Supervised Vector Machine
40	TCP	Transmission Control Protocol
41	TN	True Negative
42	TP	True Positive

43	UBA	User Behavior Analysis
44	UDP	User Datagram Protocol

Chapter One: Introduction

1.1 Background

In recent years, significant developments in communications and information technology have occurred. Technology used widely into all aspects of life, especially in the financial sector. Methods of conducting financial operations have diversified, and people have become more dependent on financial wallets and mobile banking. With the development of communication methods through mobile devices, particularly with the advancement of 3G, 4G, and 5G networks in cellular technology, the use of electronic financial wallets in financial operations and commercial exchanges have spread more widely. These online operations varied, such as withdrawals and deposits, money transfers, online purchases, stock trading, etc.

This development led to a significant improvement in efficiency and methods of hacking techniques executed by hackers to hack networks and electronic accounts of individuals and companies (Alenezi et al., 2020).

Moreover, reactively, it led to a development in the performance of protection measures against cyber-attacks. This development increased the efficiency and effectiveness of protection tools like Intrusion Detection Systems IDS, Firewalls, and Intrusion Prevention Systems IPS. IDS is a mechanism to detect attacks and alert technical administrators about attack attempts with no intervention to mitigate attacks. IPS, which stands for Intrusion Prevention System, acts as the detection and prevention of suspicious traffic (Fazal et al., 2022).

In the past, all these tools have been built on the concept of signature-based classification, which means “IP Reputation” where any of these security systems has its repository of malicious IP addresses that have a history of suspicious behavior such as malware, Scanning, spoofing, phishing(Usman et al., 2021).

IP Reputation means in cyberspace refers to the reliability or credibility associated with entities such as IP addresses, domains, files, or even individuals and organizations. It is a key concept used to assess potential risks and ensure security in the digital realm (Sucuri, 2022)

In addition to lousy reputation IP addresses, these tools have been built repositories based on malicious patterns and behaviors against networks; this repository contains identified behaviors that indicate hacking and abnormal traffic.

Nevertheless, this method did not solve the whole problem; it did not have the optimal degree of protection against malicious traffic targeting the networks since there were attacks from IPs and behaviors that still needed to be added to the repository, a zero-day attack. This type of attack exploits unidentified vulnerabilities that cybersecurity tools cannot detect. (Guo, 2023).

Machine learning contributed greatly to overcoming such a problem. All new security systems include dynamic detection based on Machine Learning (ML) techniques and take suitable actions to allow or deny traffic.

In this chapter, the description of the problem is presented and demonstrated through the problem statement section; Chapter 2 discusses the literature reviews, demonstrates the meaning of (Intrusion Detection Systems) IDS and (Intrusion Prevention Systems) IPS, and the differences between both systems; it also discussed some of proposed IDS and IPS solution presented thorough previous research. Chapter 3 discussed the methodology and proposed solution for Normal Traffic Detection (NTD) to enhance detection and resolve the problem statement, as demonstrated in section 1.2. The experiment discussed in Chapter3 showed the use of three classification algorithms separately: Supervised Vector Machine SVM, Artificial Neural Networks ANN, and Decision Tree, as well as compared the evaluation metrics for each algorithm independently to judge the efficiency and effectiveness of NTD, evaluation metrics such as F1 score, Recall, Precision, and elapsed time were used for evaluation, this is because it focuses on resolution time and accuracy. Then, it presents the proposed solution, how it will work, and how it will improve the detection process, then applies the required actions on the firewall to block or permit access based on the prediction results.

The experiments and results analyses discussed in Chapters 4 and 5 showed that the proposed NTD model achieved the highest values making it a good choice to be used in cyber security traffic classification. It showed very high accuracy compared to the algorithm above, F1 0.975, Precision score 0.983, Recall 0.970, accuracy 0.972, and low processing time measurements of 6.155 seconds. The F1 score for using SVM 0.948 separately has been calculated. The result of calculating the F1 score for Decision Tree was 0.823, and for ANN 0.953.

This indicates that the proposed solution NTD can be considered as a perfect choice for data classification. It will act in case of wrong traffic classifications hitting the networks and resolve the problem statement.

1.2 Problem Statement

Most detection and prevention systems have built their solutions on two technologies, traditional signature-based detection and pattern-based analysis, using machine learning techniques. Machine learning methods have improved the efficiency and effectiveness of the protection process.(Sen & Mehtab, 2020)

IPS and IDS

IDS and IPS systems are responsible for detecting anomaly attacks against networks, and there are three types of systems based on mechanism (Singh & Khare, 2022) :

(1) Signature-based IDS: This type stores signatures of known attack types to find similar attacks targeting the network. (Sowmya & Mary Anita, 2023)

(2) Anomaly-based IDS: Detects anomalies in traffic behavior or pattern. This mechanism applies to match the pattern detected with stored normal system behavior (S. , Kumar et al., 2023).

(3) **Hybrid IDS**: This type uses both mechanisms together. (Qazi et al., 2023),

Assigning a public or private IP address is determined by the availability of IP addresses in a pool of IPs from Internet Service Providers (ISPs) and cellular operators, in the case of 3G, 4G, or 5G. These ISP providers assign private IP addresses to users and then apply a Network Address Translate (NAT), where a public IP address represents all private IP addresses of internal machines. (Ebbers, 2016) which is a mechanism by which traffic is handed over to the internet. Below figure 1.1illustrates the process:

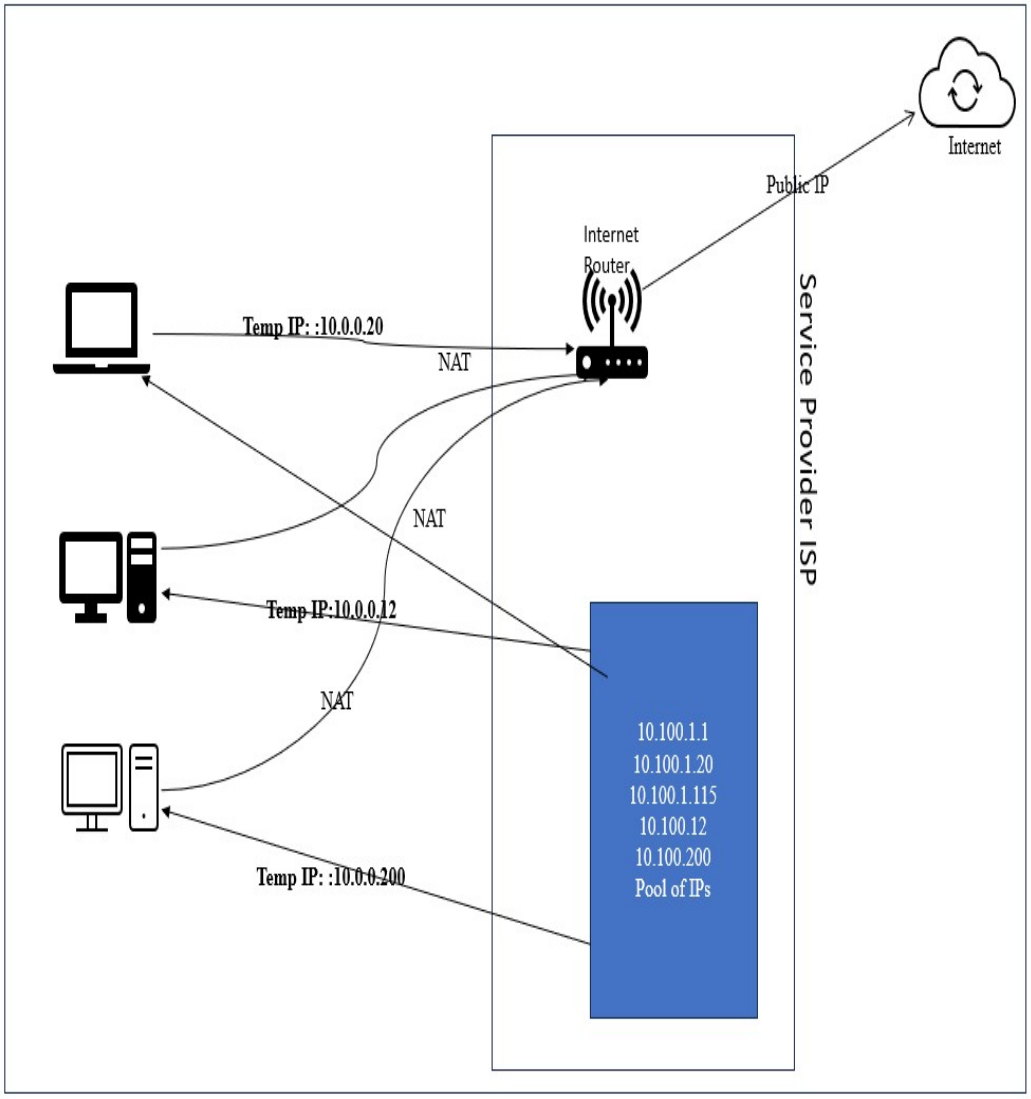


Figure 1.1: Internet Access Through ISP

According to the above flow, the problem is that a normal traffic which does not include suspicious and malicious traffic behavior will be blocked and prevented from accessing the destination network because of a misdiagnosis of the IP address, which considers that IP addresses are malicious where the traffic is normal. Due to the very high traffic sourced from many of these users, a high rate of this traffic contains malicious and suspicious behavior; this will lead to classifying NATed IPS belonging to internet service providers (ISP) as malicious. Therefore, Security systems such as firewalls, IPS, IDS, and WAF will register the repository and then block it.

As a result, all customers and users who access the Internet through this NATed IP will be blocked once they try to access destination networks such as banks, wallets, mobile service providers, etc., even if they have normal traffic.

This problem was raised and discovered when IPs, especially for ISPs (Internet Service Providers) and Cellular technology users Third generation of connective/network technology (3G), Fourth generation of connective/network technology (4G), Fifth generation of connective/network (5G), and Long-Term Evolution (LTE), are blocked from destination networks (Salman et al., 2023).

The high probability of obtaining public Nated IP address that has been used for malicious activities by another user who may apply abnormal behavior will occur, which will lead to adding this IP to the database signature and then being blacklisted in the security systems, therefore, this IP address will be blocked and prevented from accessing destination networks all over the world where any security system such as firewalls or web application firewalls WAFs, which is meant by abnormal behavior such as hacking attempts, viruses, malware, and scanning,

As a result, the security system in the destinations will prevent access to its networks. This will cause interruption of the service provided by the destination company, since preventing access means that users or clients who expect to get service from various sectors through online services like wallets, mobile banking, and trading cannot execute their transactions.

In this thesis, NTD mechanism is built on four phases:

Phase I: Pattern analysis of traffic originating from bad reputation IPs to decide whether traffic is normal or anomalous. In this phase, to get the most optimal accuracy three classification algorithms where used:

First: SVM: Supervised Vector Machine.

Second: ANN: Artificial Neural Networks.

Third: Decision Tree.

Phase II: Reconfiguring Security devices such as IPS, firewalls, to permit only all normal traffic from the source IP will enable users to handle these transactions, regardless of source IP address reputation. In the case of malicious traffic, the model will do nothing with no intervention and keep the action taken from the destination network which is to block the traffic.

Phase III: Evaluating NTD efficiency and effectiveness. The F1 score, a combination of two metrics calculated by compensating recall and precession, has been used to evaluate algorithm performance.

Phase IV: Comparing three algorithms separately with the proposed solution.

Two datasets were used in the model training and testing, as well as for comparison and analysis of the results generated; one data set will be created from scratch to help the model perform a successful prediction, and another data set will be built from a collection of traffic data from various sources as mentioned in section 4.2. Also, the model will be fine-tuned to achieve the intended result. Therefore, due to the lack of a specific-domain dataset, the following research questions have been formulated:

Question 1: Can the proposed solution (NDT) accurately detect regular traffic originating from IPs identified as malicious?

Question 2: Does the proposed model achieve higher accuracy than other protection system models?

Question 3: Does the proposed solution achieve high performance in response time?

Question 4: Regardless of the IP address's reputation, can the proposed solution be considered a good choice for permitting normal behavior?

1.3 Cyberspace

Most of the users' activities in cyberspace, at all levels, are carried out in cyberspace; these activities include all real-life transaction domains such as financial, economic, social, commercial, and governmental activities; these activities are practiced in all fields of users including, non-governmental, governmental institutions, and individuals (Li & Liu, 2021)

Chalupniks, K.((Chałubińska - Jentkiewicz, 2022) defined Cyberspace as a term consisting of a combination of the two words Cybernetics and Space, which means cybernetic space, where Cybernetic refers to the science of communications and automatic control systems in both machines and living, things. In 1982, a science fiction author, William Gibson (Gibson, 1982) , created the word cyberspace in his book, **Burning Chrome**. Even if this was fictional, this term has become used in professional and academic contexts.

William described cyberspace in this book as a

"*Consensual hallucination experienced daily by billions of legitimate operators, in every nation, by children being taught mathematical concepts... A graphic representation of*

data abstracted from the banks of every computer in the human system. Unthinkable complexity" (Gibson, 1982)

This definition focused on how people perceive a new environment, but it is still very applicable because it shows how fully interactive cyberspace knowledge could be created. National Institute of Standards and Technology NIST defined cyberspace as:

"*A global domain within the information environment consisting of the interdependent network of information systems infrastructures including the Internet, telecommunications networks, computer systems, and embedded processors and controllers."* (National Institute of Standards & Technology, 2016)

Cyberspace Domains

Cyberspace can be considered a mixture of multiple fields, such as the physical, digital, network, and social domains while the cyber domain can be considered part of cyberspace; the sections below describe various types of cyber domains, the following table describes each type of cyberspace domains.

Table 1.1 Cyberspace Domains

Domain Name	Description
Physical Domain	It consists of physical security information in cyberspace, such as rooms, buildings, doors, and hardware devices like computers, printers, scanners, and server terminals. In this domain, both attacker and security defender may enter the room, get out of the room, and control computers in addition to other actions to use these devices (L. Zhang et al., 2022)
Digital Domain	The digital domain includes information that is digitally stored in cyberspace, which means digital data like usernames, passwords, documents, applications, and databases (L. Zhang et al., 2022); the digital layer called the logical layer also, as it contains the logical data software, data packets (Tsagourias, 2021).
Network Domain	The Network domain plays a major role in cybersecurity according to dictionary.com (Dictionary.com, 2021), the network domain includes a system containing combinations of hardware devices like personal computers, computer terminals, servers, multifunction machines, display devices, and telephones connected through telecommunication media like cables and wireless devices used as an intermediate for data transmission. The main objective of networks

	is to share resources; therefore, in cyberspace, computer networks should be protected to guarantee the protection of communication through these networks. Network security means that a network system does not contain threats and can understand the function of resource sharing. To achieve this goal, both the network's hardware and software should work normally; in addition, data security should be maintained during information transmission (Zheng, 2021).
Social Network Domain	This domain allows users, individuals, groups, and governmental and nongovernmental organizations to connect to share ideas, beliefs, and knowledge. This exchange will provide a very large and rich pool of knowledge; therefore, this knowledge and information may have a critical role for corporations and governments. In addition, the majority of social media posts are written in text format, enabling text processing and applying data mining techniques to people's sentiments, feelings, and beliefs (Malizan et al., 2022). Bishop, M (M. Bishop, 2019) assured that social media are effective and efficient interactive communication methods in the health sector; this effect has appeared very clearly in developing countries, which still face constraints of limited access to healthcare systems.

1.4 Cyberattack

An assault carried out by attackers using more than one computer or network is referred to as a Cyberattack. Such attacks have the potential to steal data and maliciously disrupt the system. Utilizing infected computers allows for this (Hasan & Al-Ramadan, 2021).

In addition to the above definition, and according to Bebeshko (Bebeshko et al., 2021)a Cyberattack can be defined as the use of technical weakness and shortcomings of security techniques in modern cyberspace to interrupt the services introduced and steal valuable information from the organization .

The National Institute of Standards and Technology (NIST) (National Institute of Standards and Technology, 2011), also has defined cyber-attack as

"*Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself*. "

According to Miao (Miao, 2021), a Cyber-Attack is malicious behavior that aims to disable, destroy, or control the environment/infrastructure and information to destroy one of three pillars of information security: Confidentiality, Integrity, and Availability of the data." CIA is defined in the section below.

1.5 Cybersecurity

Many literatures and dictionaries have defined cybersecurity as tools, policies, processes, and procedures that aim to preserve and maintain data confidentiality, integrity, and availability, where Confidentiality aims to preserve authorized access and disclosure, including measures and tactics to protect privacy and proprietary information, Integrity protects data from modification or deletion by unauthorized users, which means maintaining data accuracy, and Availability means Ensuring that systems, services, and data are available and accessible when needed (H. Azam et al., 2023). Accountability is the ability to trace activities on the system to a specific user or person; it aims to map an activity to the responsible person or user (ISACA, 2008)

American Dictionary Merriam–Webster dictionary, which is considered one of the publishers of language references, defines cyber security as "*measures taken to protect a computer or computer systems (as on the Internet) against unauthorized access or attack*" (Cains et al., 2022)

International Telecommunications Union (ITU) (Study Group, 2008)published another definition for cyber security that states: "*The collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets" within the cyber security foci of confidentiality, availability, and integrity (CIA) objectives*".

The Department of Homeland Security's National Initiative for Cybersecurity Careers and Studies glossary defines cybersecurity as an activity or process that carries out the defense of data and systems against various attack objectives, including damage, unauthorized use, modification, and exploitation (National Initiative for Cybersecurity Careers & (NICCS), 2020).

1.6 Attack Techniques and Cybersecurity.

Many literatures and dictionaries have defined cybersecurity as tools, policies, processes, and procedures for preserving and maintaining data confidentiality, integrity, and availability. Confidentiality aims to preserve authorized access and disclosure, including measures.

With the development of technology and telecommunications, Cyber criminals have developed various techniques of attack; these techniques caused information security problems for the users of technology, such as individuals, private sectors, and public sector, which makes them more exposure to vulnerabilities related to security problems. Attacks and types have a massive impact that stimulated many researchers to discuss different types, techniques, and methods of attacks on all layers and the technology field. Attacks can be classified according to the objective of attacks, which means the objective of interrupting communication. Two types of this classification were identified: Passive attack and Active attack.

Johan Note and Maaruf Ali (Note & Ali, 2022) have introduced a diagram containing attack types, categories, and examples of these categories Figure 2-1 illustrates the diagram. These categories include Misuse of resources such as Man in the middle, User Access Compromise, Root Access Compromise, Web Access Compromises including SQL Injection and Cross-site Scripting, Malware like Viruses, Trojan, Spyware, and Ransomware, and Denial of Service DOS like Host-Based, Network-Based, and Distributed (DDOS) and tactics to protect privacy and proprietary information.

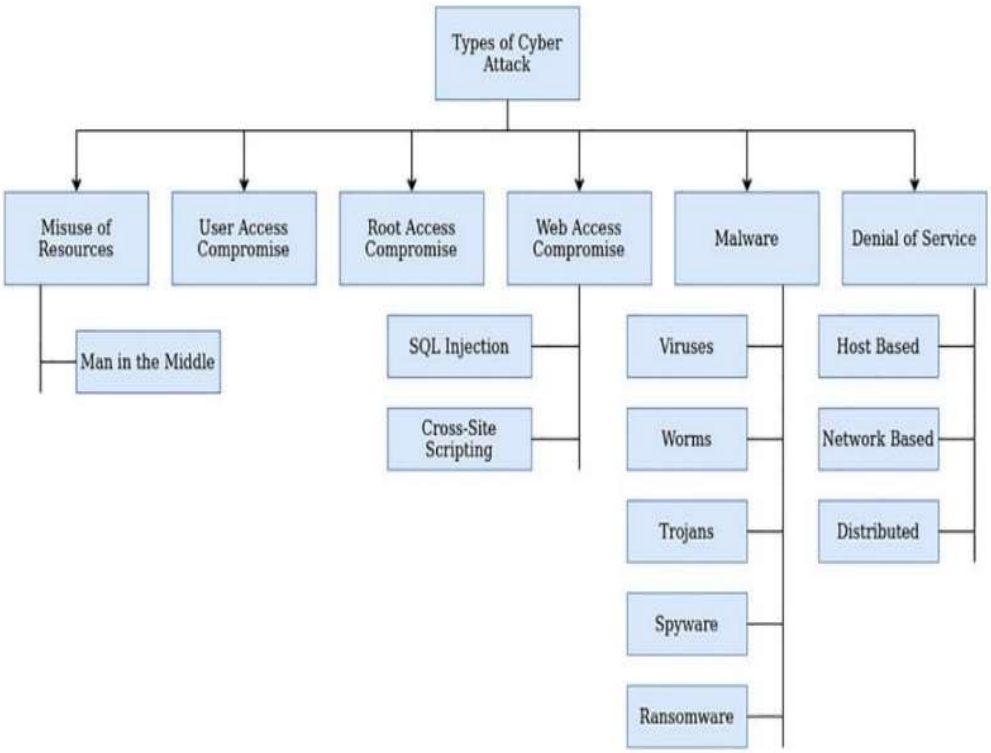


Figure 1.2: Attack Categories and Types

1.6.1 Passive and Active Attacks

Hackers have divided the attacks into several ways; two major types of attacks have been defined in cyberspace and cybersecurity: Passive attack and Active attack. In a passive attack, the attacker listens without interfering or affecting data flow or changing the data transferred between clients, users, and network devices; this type of attack does not affect data integrity or availability and has a massive effect on data confidentiality; examples of this type of attack include Man in the Middle (Mohapatra, 2020) .

In an active Attack, the attackers interfere, modify, and alter the content of data packets or systems; in addition to affecting efficiency, effectiveness, continuity, or denying the service, this attack affects the integrity, confidentiality, and availability. (Hadi, 2022), Examples of active attacks include Modification, Spoofing, Replaying, Repudiation, and Denial of Service (DoS). (Canto et al., 2023).

1.6.2 Targeted and Untargeted Attacks

Another way to classify attacks is through targeted and untargeted attacks. Jibi Mariam Biju, Neethu Gopal, and Anju J Prakash (Biju, 2019)have defined two main types

of attacks: targeted attacks and untargeted attacks. In a targeted attack, the attacker has a particular interest in the specific target group, and the perpetuation of this type of attack may take a long time as the attacker puts much effort into piercing the system. The attacker uses spear phishing attacks, botnet deployment, and subverting the supply chain.

In an untargeted attack, attackers target many users or devices. They use techniques that include phishing, ransomware, and scanning, but they are not limited to these.

Cybercriminals use many attack methods (Yohanandhan et al., 2020). In their survey have listed sixty-three methods of these attacks, including data tampering attacks, man in the middle attack, DOS attacks, SQL injection attacks, …,

According to Amin & Rahman Attacks also can be categorized according to the network OSI model, along with network layers: Application, Presentation Session, Transport, Network, Data Link, and Physical (Amin & Rahman, 2023)

Position on the network:

Attacks can also be classified based on their position in the network in two ways, Internal or external: This clarifies whether the attack source comes from an internal network as an internal threat or if the source comes from outside the network. Attacks on the OSI model: This means the attacks on each layer of the network OSI model (physical, data link, network, transport, and application layer); in this classification, some attacks can target more than one network layer (Bengag et al., 2021)

Common Types of Attacks

The National Institute of Standards and Technology NIST (National Institute of Standards & Technology, 2016), a US agency responsible for setting standards and measures, has identified common types of attacks: Identity Theft, SPAM, Web Attacks, and Ransomware. These types are listed below

Identity Theft: Illegal use of another user's identity information, like username, without his or her approval. (Wyre et al., 2020)

SPAM: Any undesired, intrusive digital communication transmitted in large volume, frequently transmitted by email, or shared through text messages, voice calls, and social media. (Malwarebytes, 2020)

Web Attack: Malicious activities search for any kind of vulnerability in websites to get unauthorized access, in order to steal confidential information, introduce malicious content, or modify and alter the website's content (Dawadi et al., 2023).

Ransomware: A special type of malware that encrypts a targeted sensitive file for the victim; these files include financial data, business data, databases, and personal information, with a decryption key to restore encrypted files. (Ali, 2017)

(Cisco, n.d.) Identified the top seven cyber-attack types: Malware, Denial of Service DOS, SQL injection, Cross Site Scripting, DNS Tunneling, Zero-Day-Exploit, Man in the Middle MITM, and Phishing. Cisco did not mention other important types of attacks, like social media attacks, advanced persistent threat (APT), Khaleefa and abdullah have identified the APT (Khaleefa & Abdulah, 2022). which is a sophisticated method with main objective of the attack is to steal sensitive information from targeted victims

Coursera (Coursera Staff, 2024) , a learning platform identified eight basic types of attacks: Malware, Phishing, Spoofing, Backdoor Trojan, Ransomware, Password attacks, Internet of Things attack, Cryptojacking, Drive-by download, Denial-of-service attack

(Matzelle, 2019)in thier work missed other important types of attack like SQL injection, Brute Force, Man in the Middle, and APT.

In general, there are many types, and techniques of attacks, which target the organizations' assets, table 2.1 contains some of the well-known methods and types of attacks.

Table 1.2 List of Attack Methods

Worm	Data Tampering	False Negatives
Password Cracking	Man-in-Middle	False Alarms
Repudiation Attack	Replay Attack	Stealthy Attack
Ransomware	Denial of Service (DoS)	Masquerade Data
GPS Spoofing	False Data Injection	Time Delay
Payload Attack	Switching	Disordered Data
Mail BOMB	Data Integrity Attack	Information Disclosure
Smurf	Data Availability Attack	Confidentiality Attack

1.7 Summary

This chapter studies the prompt advances in communication technologies and their effect on economic operations, such as mobile banking and digital wallets.

While these developments have enhanced transaction efficiency, they have also led to advanced cyber threats targeting networks. Traditional security tools like IDS and IPS depend on signature-based methods, concentrating on known malicious IP addresses and patterns. However, these methods struggle against zero-day attacks, which utilized

unexplored vulnerabilities. The next chapter will discuss the existing research related to the study's topic. This literature review helps set the stage, points out what's missing in current knowledge, and shows how the thesis adds to the academic conversation.

Chapter Two: Literature Review

2.1 Mechanisms to Detect Attacks

This chapter covers previous works on solutions to detect and prevent anomalies and abnormal behavior in network traffic. Most research discussed this subject, including analysis and evaluating detection and prevention techniques.(Qaddoura et al., 2021).

While searching for related work small amount of research discussed permitting normal pattern detection to access networks. Also, a few papers proposed solutions built on three layers of detection, i.e., three classification algorithms.

(Al Jallad et al., 2020)

Diaba, S. Y., & Elmusrati, M. (Diaba & Elmusrati, 2023) presented an algorithm to detect Distributed Denial of Service DDOS attacks through a hybrid Convolutional Neural network and Gated Recurrent Unit (GRU) in a physical smart grid. Alshehri, A., Khan, N., Alowayr, A., & Alghamdi, M. Y (Alshehri et al., 2023)employed machine learning techniques with User Behavior Analytics (UBA) to present a proposed attack detection framework; this framework used a user activity over a network to determine normal behavior.

Hybrid models play a crucial role in identifying network attacks. Hongyu Liu and Bo Lang (H. Liu & Lang, 2019) combined support vector machines (SVMs) with k-means clustering to detect intrusions. They first used k-means to cluster network traffic data, identifying potential anomalies. Then, they applied SVM to assess these clusters and classify them as malicious or benign. This hybrid approach significantly improved detection accuracy and reduced false positives with percentage of 4.1% compared to standalone methods, highlighting the effectiveness of combining supervised and unsupervised learning in anomaly detection. Tekerek A, Bay O(Tekerek & Bay, 2019). presented two mechanisms for detecting attacks: Signature-Based Detection (SBD) and Artificial Intelligence-Based detection (AIBD). SBD detects known attacks, and AIBD detects anomaly traffic.

2.2 Signature-Based Detection (SBD):

Events and behavioral activities of traffic are used to generate the signature; this signature is matched with a signature database to detect attacks, like Intrusion Prevention System IPS, firewall, Web Application Firewall WAF, or Intrusion Detection System IDS,

alerts and blocks such malicious traffic found in these signatures. (Sihag Vikasand Swami, 2020).

A traditional signature-based detection, which depends on Signature Detection, is a method that identifies specific patterns in network traffic that match pre-defined attack signatures registered in the database (Markevych & Dawson, 2023).

The main shortcoming of this method is that it is effective and efficient in detecting only pre-defined known attacks registered on its database. However, in the case of new attacks or anomaly behavior that does not exist in the database, or the last updates are not installed on the signature, or in case of zero-day attacks, it has a severe problem, as it cannot detect such attacks because the dependency of catching malicious traffic depends only on the signature database it has (Elshafie et al., 2019) . Therefore, the SBD method has severe problems with zero-day attacks, which are new and unknown. (Meddeb et al., 2023) . The seriousness of this problem lies in the raising rate of Zero Day Attack, which has made this technique less effective and perform poorly (Spadaccino & Cuomo, 2020).

2.3 Machine Learning Techniques

With the increase of zero-day attacks and the development of attack methods, traditional cyber detection systems have become unable to detect newly developed attacks. Over the last few years, machine learning has been developed to improve cyber-attack detection methods, and traditional signature-based detection. Machine learning has been used to improve the effectiveness and efficiency of attack detection, such as malware, breach recognition, and other types of attacks (Fraley & Cannady, 2017).

Machine Learning is a subfield of Artificial intelligence, which is the ability of machines to simulate humans in solving problems. It is a branch of computer science that trains computers without being programmed (Bi et al., 2019). It is a type of artificial intelligence technique that can automatically discover information from huge datasets (Michie et al., 1995).

Several research studies have been developed to use machine learning in data science to enhance attack detection in different areas, such as the Internet of Things (IoT), networks, and telecommunications. For instance, Dr. R, Valanarasus and A, Christy (V. R & A, 2019) has used a combination of the neural network NN and support vector machine SVM for detecting and classifying the distributed denial of service attacks DDOS in telecommunications networks to enhance detection accuracy. He used features such as

packet type (UDP, TCP), window size, time window, traffic flow up to 1000, payload, and packet size 1024 as simulation parameters. Vinayakumar R, Alazab M, Soman K, Poornachandran P, and Venkatraman S (Vinayakumar et al., 2019) Proposed a new scalable and hybrid framework named ScaleMalNet; this model helps collect malware samples from various sources. In addition, the research presented a processing technique for malware classification; this framework follows an approach of two stages: the first stage was built on classifying the executable files into malware or normal using both dynamic and static analysis. Then, in the second stage, the framework has classified the malware file into the corresponding malware family.

Merton Lansley, Francois Mouton, Stelios Kapetanakis & Nikolaos Polatidis (Lansley et al., 2020) developed a mechanism that detects attacks on social media, known as social engineering attacks. They built a method on neural network algorithms and natural language processing. Both offline and online texts can be examined using the method. First parsing is applied to the text, checking the grammar of the text using natural language processing. Then artificial neural networks classify the text if it contains normal traffic or abnormal attacks. They used the decision tree algorithm, random forest algorithm, and multi-layer perception algorithms. They measured the accuracy of the method over both real and semi-synthetic datasets, The results of their experiment are shown below Table 02-2:

Table 2.1: Result Comparison (Lansley, 2020)

Real Dataset (a)			**Semi-Synthetic Datasets (b)**		
Algorithm	Results		Algorithm		Results
Decision tree	0.681		Decision tree		0.918
Random forest	0.683		Random forest		0.9107
Multi-Layer perception	0.691		Multi-Layer perception		0.925

The above table shows unsatisfactory results, as the percentage does not exceed 92%, meaning that there is about 8% inaccuracy in the results, where 8% is high compared with massive traffic in the network and the criticality in cyberspace, which means every 100-network packet we have a probability of eight malicious packets.

Many machine-learning methods were used to detect abnormal traffic behavior. The critical point in these techniques is the efficiency and effectiveness of the proposed detection approaches that use ML algorithms and datasets.

2.4 IPS and IDS

Intrusion Detection System IDS is a system that detects and monitors traffic for potential malicious and abnormal behavior and attack attempts; it is passive, meaning that it does not make any intervention in traffic flow; it is just used for alerting, whereas Intrusion Prevention System (IPS), works actively, it detects malicious traffic and prevents such traffic from reaching the targeted resources (Thapa & Mailewa, 2020) . The terms intrusion detection system (IDS) and intrusion prevention system (IPS) were initially used in an academic work titled "An Intrusion Detection Model" by Dorothy E. Denning in (Denning, 1987)The Stanford Research Institute (SRI) took this chance and, created the Intrusion Detection Expert System (IDES). The IDS system used statistical anomaly detection, user signature profiles, and host systems to identify malicious network behaviors .(Radoglou-Grammatikis et al., 2020)

2.5 Intrusion Detection System IDS

The first intrusion detection system was proposed in 1980 by Anderson J (Anderson, 1980), and many IDS products have been introduced since then. These systems differ in two ways: design and structure, as well as data collection and monitoring mechanisms.

However, most of these proposed systems depend on the design shown in Figure 1 below. Which consists of three components:

Sensor device: a gathering device that collects data from systems and components integrated into IDS.

Analysis: responsible for data processing, analyzing received data, and deciding anomaly activity.

- **Knowledgebase Component:** a Database that contains collected and processed data; this component includes information on attack patterns and data profiles.

Finally, the configuration device provides information about the IDS system's status and is responsible for initiating all procedures and components; Figure 2-2 below illustrates the mechanism of the basic IDS system (Faker & Dogdu, 2019)

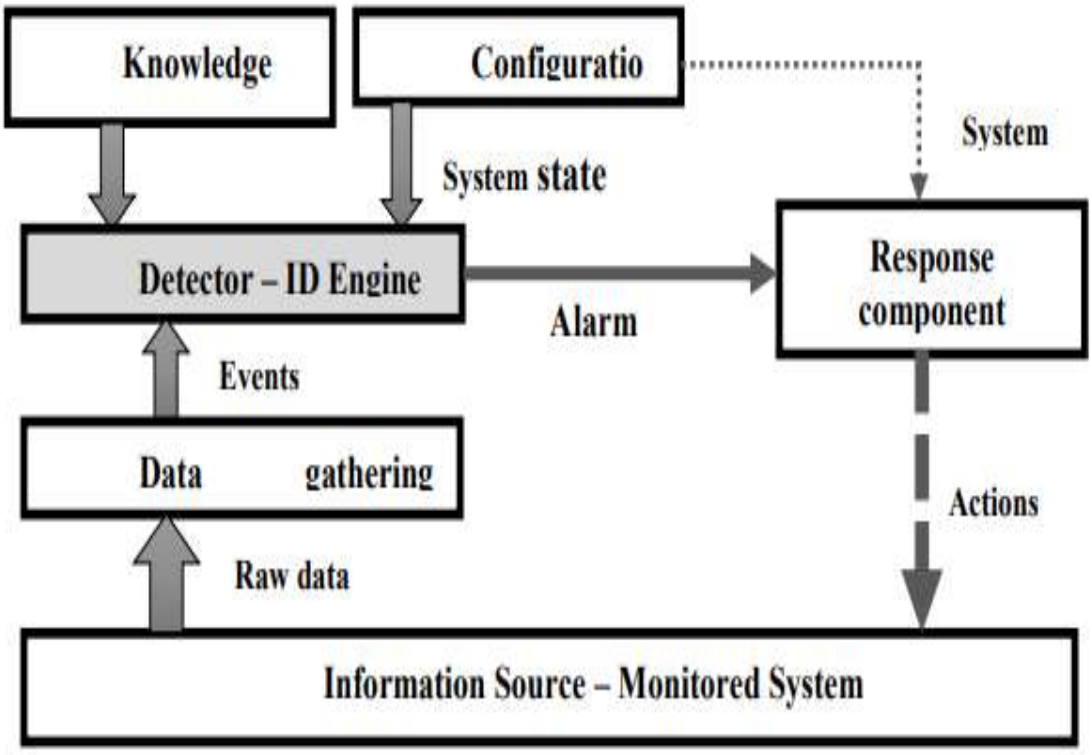


Figure 2.1: Basic Intrusion Detection System design

However, many proposed intrusion detection systems still face a high rate of false positive alarms as well, and they generate many alerts for low risk while they are high risk; this approach will lead to reduced trust of security analysts and, as a result of that it will lead to ignoring severe security alerts. Many researchers have focused on developing IDSs with higher accuracy and efficiency to reduce false favorable rates. For instance, ftikhar Ahmad , Qazi Emad Ul Haq, Muhammad Imran, Madini O. Alassafi, and Rayed A. AlGhamdi(Ahmad et al., 2022) proposed a network detection system based on artificial intelligence techniques; they used **Ad boost** techniques for their proposed system.

Ad boost, also called the meta-learning method, is designed to increase the efficiency of binary classifiers in machine learning (Kurama & Vihar, 2020) . They selected features from the known UNSW-NB15 dataset, a hybrid dataset containing two types of behaviors: real normal behaviors and attack activities (Moustafa & Slay, 2015); the proposed solution used three techniques. Artificial neural network (ANN): to match targets, Support vector machine (SVM), and Ad boost. The solution design shown in Figure 2-3 is based on different components: Training data, testing data, data processing,

feature selection, applying the Ad boost model, which contains the training and testing phase, and classification.

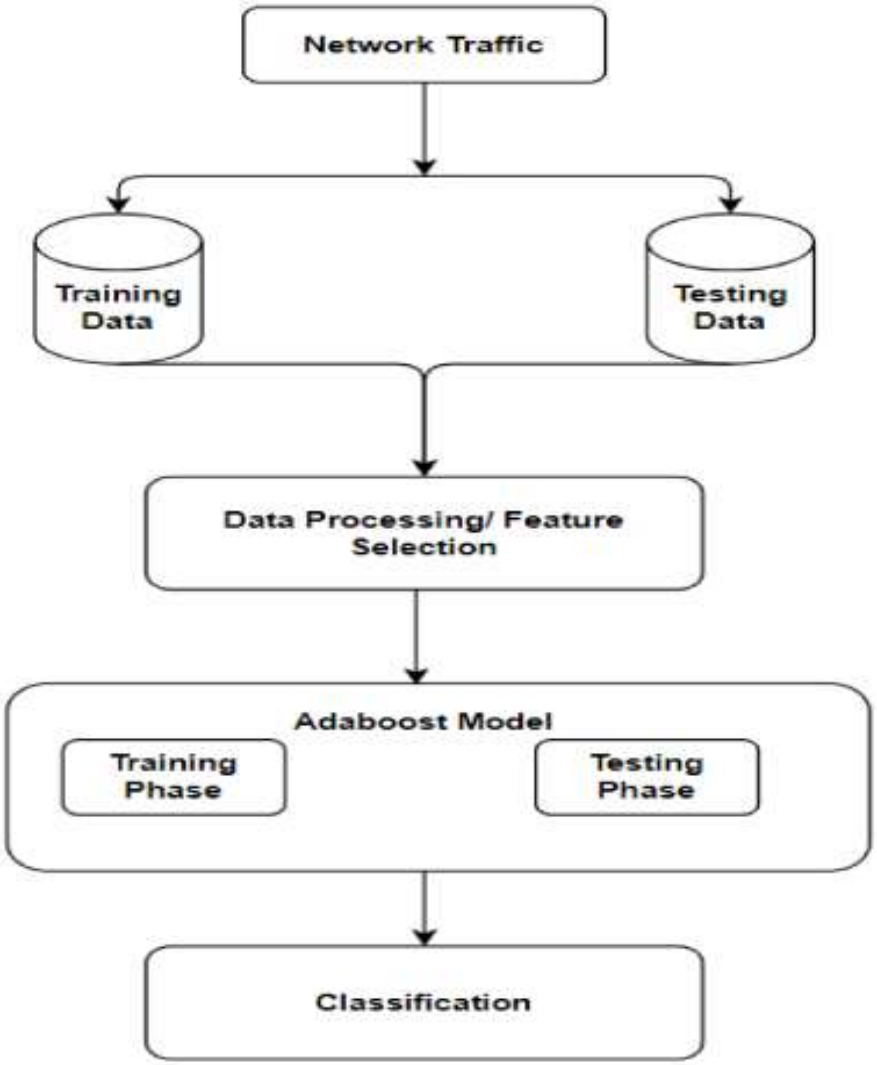


Figure 2.2: Proposed Solution Design

Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (G. Liu et al., 2022), Presented an enhanced intrusion Detection Model based on an improved K- Nearest Neighbor (KNN) technique to detect anomaly behavior and attacks, especially DDOS, and various attacks in wireless sensor networks (WSNs).

The WSN solution used a combined Arithmetic Optimization Algorithm (AOA) and KNN algorithm to detect numerous DDOS attacks. Figure 2-4 below describes the proposed architecture and illustrates the model.

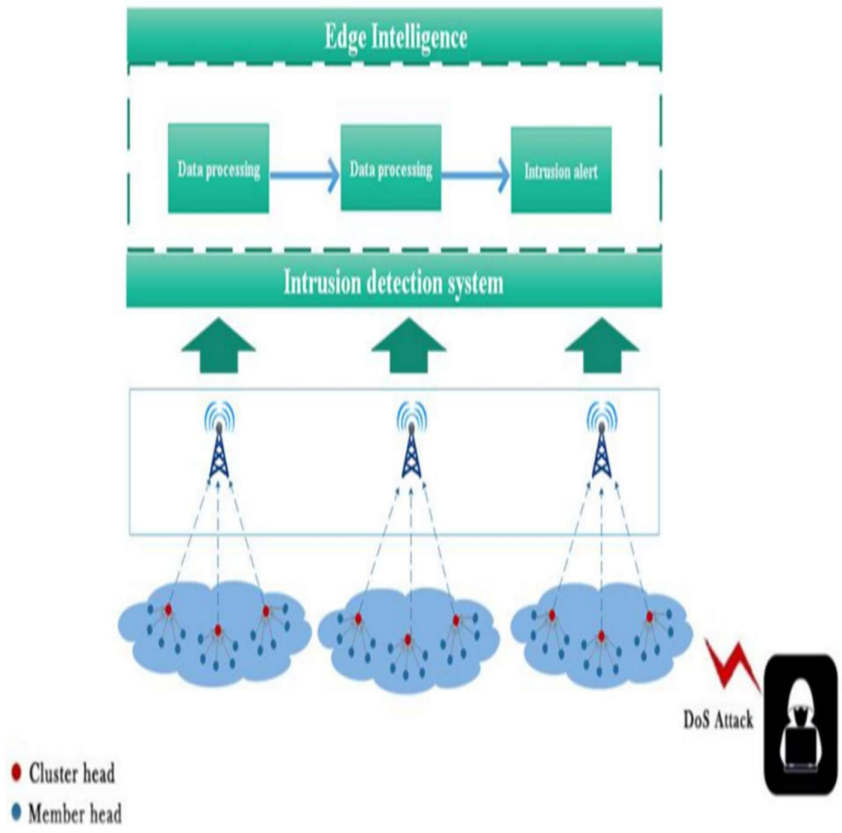


Figure 2.3: WSN Model of IDS

Current intrusion detection systems have several problems and challenges, summarized into three areas (Khan et al., 2019) The first problem is that the existing intrusion detection techniques are often ineffective in achieving high detection rates or lowering false alarms because they cannot keep up with the ongoing evolution of the cyber threat landscape and the emergence of new threats.

The second problem is that traditional machine learning techniques, utilized in intrusion detection systems, have several drawbacks, including overfitting and excessive bias caused by redundant or irrelevant information and unbalanced class distribution of network traffic.

The third problem is labeling the network data set for intrusion detection systems development, which is a severe problem. Over time, extensive work is needed to develop labeled datasets.

2.6 Intrusion Prevention System IPS

(Anggraeni et al., 2022) in their work have focused on the implementation of intrusion prevention systems (IPS) and their role in protecting and strengthening the networks from being hacked, also the illustrated intrusion prevention systems (IPS) discover and prevents suspicious traffic and attack attempts This research did not address allowing natural traffic to pass through the net, but the focus was only on preventing entry. Many proposed IPS solutions have been presented last period; for example, Bocu R, Iavich M (Bocu & Iavich, 2022)proposed a real-time intrusion detection and prevention system based on machine learning techniques on 5G networks. This research used a convolutional neural network (CNN) to train the model. Researchers proposed a design of a detection and prevention system that contains four stages or layers:

Data Forwarding Layer: This is the first stage in the model design. It is responsible for monitoring and collecting traffic, it detects suspicious pattern in real time and sends information to the data management and control layer to block it.

Data and Intelligence Layer: This layer detects suspicious traffic and identifies the anomalies based on the analysis.

Data Management and Control Layer: Receives data security measures from data intelligence. The architecture and design for the proposed solution are illustrated in Figure 2-5 below:

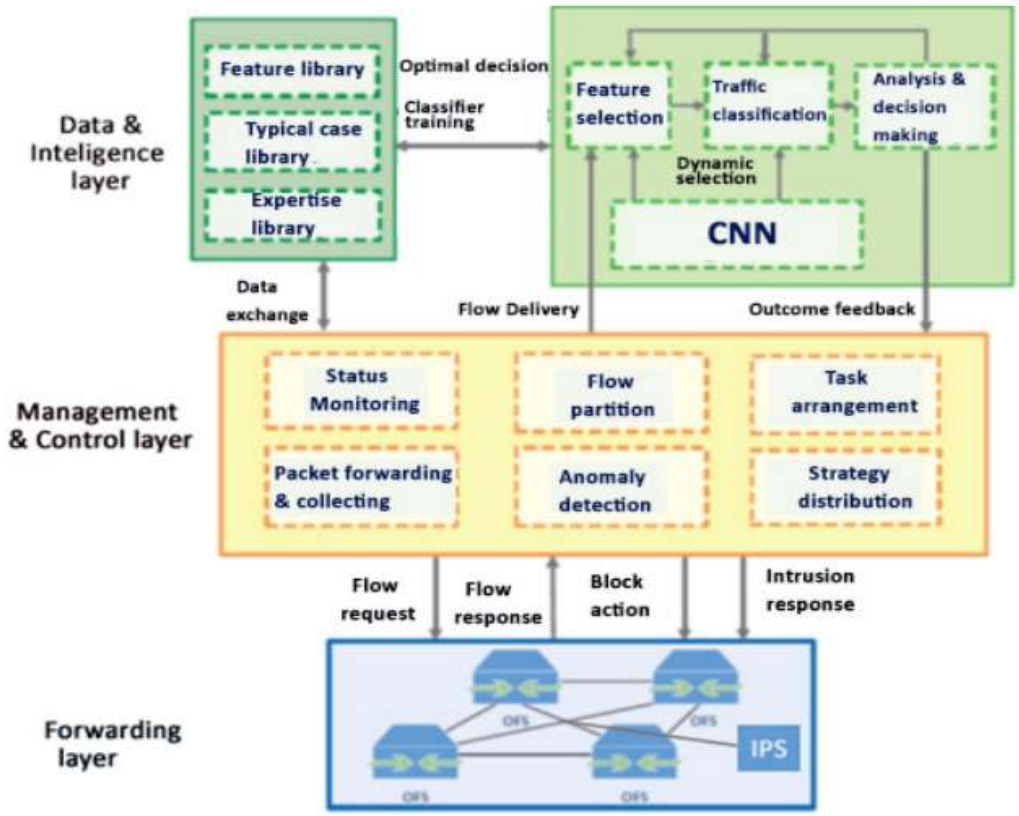


Figure 2.4: Design of IDS IPS System

This system's maximum accuracy was 94.1%, which needs improvement and will be considered insufficient in security.

Constantinides, C., Shiaeles, S., Ghita, B., & Kolokotronis, (Constantinides et al., 2019) proposed an online Incremental Learning intrusion prevention system that combines SVM, and a Self-Organizing Incremental Neural Network SOINN. Their solution does not depend on signatures, and according to experimental findings using the NSL KDD dataset.

The framework, as shown in Figure 2-6 consists of a detection engine, a preprocessing module for the incoming traffic, a validation module, and an update module that feeds the detection engine's failed results.

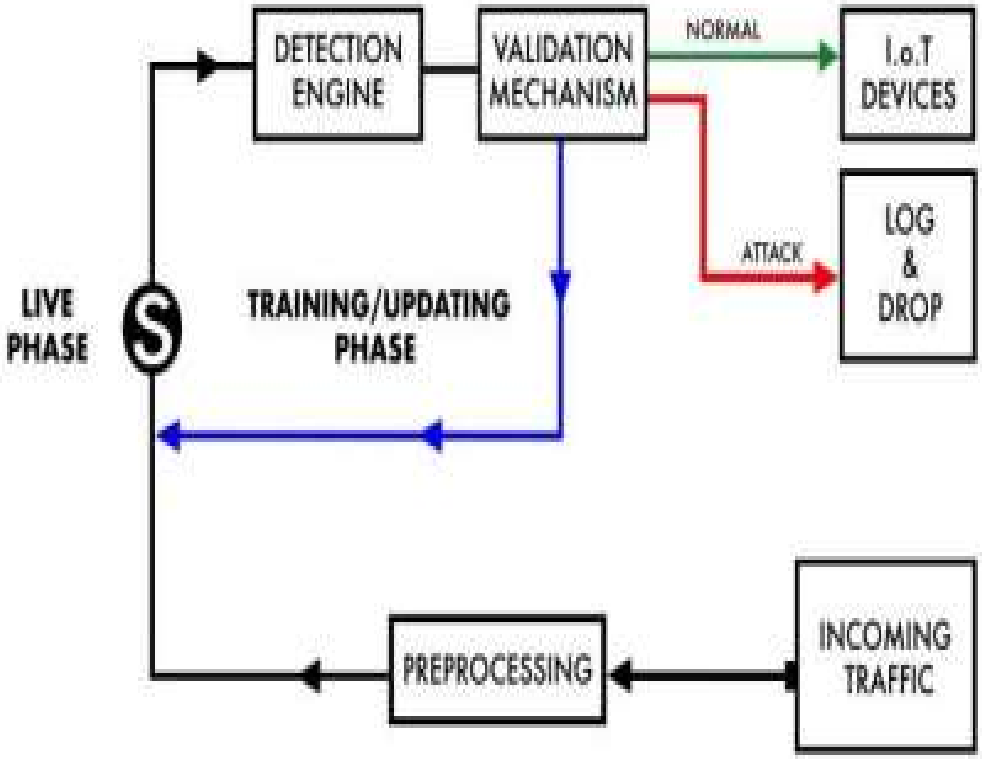


Figure 2.5: Framework of the Proposed Model SOINN

The results in Table 2-3 showed that the best accuracy metric achieved in this proposed solution is 89.67% for all five classes is not very high since the accuracy should be higher, also the time elapsed for all five classes showed long time.

Table 2.2: Evaluation and Time Elapsed of the Proposed Model.

Round	Accuracy [%]	Time [s]	# samples
Initial training	78.23	986	22544
Round 1	84.44	1065	28028
Round 2	87.98	1647	31948
Round 3	88.88	2328	34975
Round 4	88.96	2801	37776
Round 5	89.67	3285	40557

2.7 False Positive Detection

False positive detection means the classification of normal traffic from IP addresses classified as malicious by the signatures of any security component, such as firewalls, WAFs, IPS, or IDS; at the same time, these malicious IP addresses generate normal traffic.

Signature-based intrusion detection and prevention systems will not achieve accurate results with the highest efficiency (Malek, 2020).

Kuang, Xu, Suo, & Yang, (Chen et al., 2020) proposed an IPS based on Convolutional Neural Networks (CNN). This solution comprises two parts: Part one contains offline training using CNN, which includes an input layer of 9×9. It then reduces it through successive convolutional layers with a maximum pooling layer to reach an output layer of 1×1.

The second part is the online detection phase, using an open-source IDS, Suricata. The packets are pre-processed, and the trained model is used on the network traffic to produce the detection outcome. The CICIDS2017 dataset is used to test the model. Their model achieved an accuracy of 96.55% and 99.56%.

Despite the proposed system achieved very high evaluation metrics, but it did not address the problem of false positive detection of normal traffic based on signature.

Kumar Singh Gautam & Doegar (Kumar Singh Gautam & Doegar, 2018) performed three tests to show how their approach proposed better results. First, they applied normalization on KDD Cup9e9 dataset. Then, they performed feature selection using a correlation method. The feature selection used information gain as a decision factor; finally, they used three algorithms: Naïve Bayes, PART, and Adaptive Boost. They obtained an accuracy of 99.9732% on the KDD Cup99 dataset only.

Pan, Fan, Chu, Zhao, & Liu (Pan et al., 2021) Proposed a solution to detect intrusion patterns in wireless networks. The solution has been built in the cloud platform to achieve maximum computational power efficiency. Moreover, to minimize the load on cloud computing, the model employed sink nodes based on the fog. They combined Polymorphic Mutation (PM) and Compact SCA (CSCA) to create a solution that was as light as possible. CSCA reduced the computational load by leveraging probability to lower the density of the data. In order to lessen the loss of precision when utilizing CSCA, polymorphic mutation was incorporated. They employed PMCSCA to improve the KNN algorithmic parameters for the optimum configuration. They tested their solution on the NSL-KDD dataset and obtained an accuracy of 99.327%, and on the UNSW-NB15 dataset, and obtained 98.27%.

Yu & Bian (Yu & Bian, 2020) They proposed a Few-Shot Learning-based IDS model (FSL). This model is a deep learning technique that can be used to learn from minor to no data. In the proposed solution, they extracted features using two embedding models, Convolutional Neural Network (CNN) and Deep Neural Networks DNN, which have at

least two layers of complexity. These models assist in reducing the input data's dimension without wrapping significant data. They used the UNSW-NB15 and NSL-KDD datasets to test the model. Their solutions achieved an accuracy of 92.34% and 92%, respectively.

2.8 Hypothesis

First: Many proposed systems and solutions failed to address the issue of allowing normal traffic from IP addresses that are incorrectly classified as malicious.

Second: The proposed model will effectively predict and permit normal traffic utilizing the dataset.

Third: The proposed (NTD) model, presents advanced solutions using multiple machine learning algorithms, and will successfully recognize and permit potential normal traffic, illustrating high reliability and strength.

Fourth: The proposed model will accomplish higher accuracy than existing models and algorithms.

Fifth: This hypothesis recommends that the proposed model, through its inventive approach and design, will beat existing security frameworks across various performance metrics such as accuracy, precision, recall, and overall performance metrics.

Sixth: The proposed model will display high performance in response time, ensuring fast detection and response.

Seventh: This hypothesis is built on the premise that the model's optimized architecture will allow it to operate productively and swiftly without affecting system confidentiality, integrity, and availability and permitting potential legitimate traffic in real-time scenarios.

2.9 Summary This Chapter investigates the evolution of cyber-attack detection methodologies. It begins with an overview of Signature-Based Detection (SBD), which is effective for identifying known threats but encounters significant challenges when dealing with new or zero-day attacks. To address these limitations, Artificial Intelligence-Based Detection (AIBD), particularly through machine learning techniques, has emerged as a promising avenue for detecting emerging threats, including malware and denial-of-service attacks. Nevertheless, these approaches still face obstacles, such as a high rate of false positives and challenges related to dataset labeling. The chapter further explores Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), highlighting their critical roles in identifying and mitigating cyber-attacks. While these systems are

valuable, they are often impeded by elevated false positive rates. In response, researchers are investigating the integration of machine learning to enhance both the accuracy and efficiency of IDS and IPS solutions. The chapter concludes by positing that a novel model that combines machine learning with advanced detection mechanisms has the potential to outperform existing systems, thereby offering improved detection capabilities and real-time response.

Chapter Three: Methodology

This chapter describes the methodology and approach to building NTD model; this methodology depends on the following steps demonstrated in Figure 3-1: Data Collection and gathering, Data Processing, Model Building, Model Evaluation, and Model Deployment, these steps.

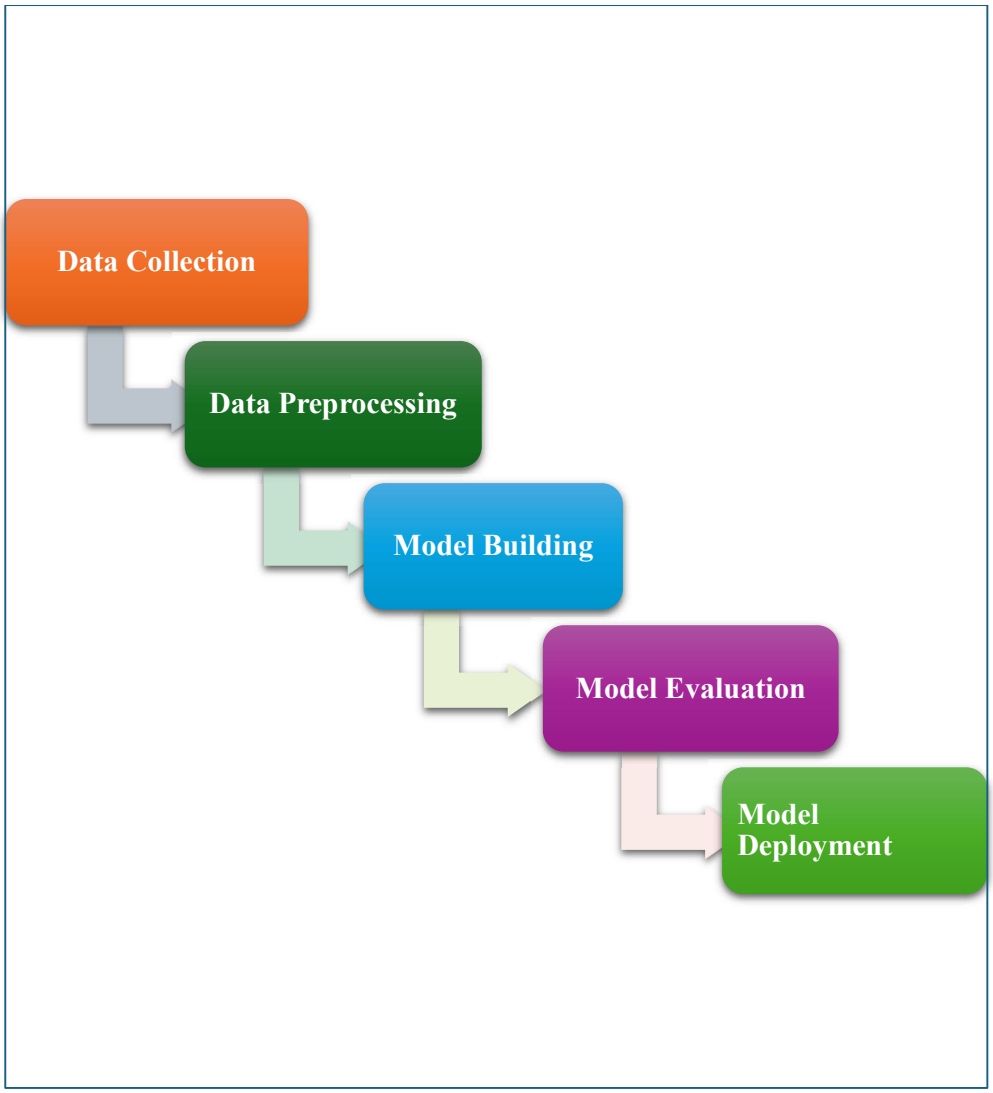


Figure 3.1: Machine-Learning Methodology of the Proposed Solution

As discussed in the literature review in Chapter 2, many previous works have proposed machine learning techniques to improve efficiency and effectiveness in detecting and preventing malicious activities against networks. These techniques use three approaches: signature-based, Pattern analysis, and Hybrid, which employ both signature-based and pattern analysis to detect traffic.

However, most of them aimed to deny the malicious traffic, but they did not reach the intended results, especially in eliminating the false positives detection; as a result, this led us to propose a solution based on the use of multiple algorithms consequently, in order to achieve optimal accuracy, effectiveness, and efficiency of detection, as well as taking action in permitting normal network traffic by building APIs that apply a whitelisting mechanism on the firewalls and other security devices, conjugated with continuous monitoring of all activities of the same source IP addresses which generated such traffic.

In short, the main objective of the proposed solution has been described as:

to detect and permit normal patterns from malicious IP addresses. To achieve this objective, the proposed model NTD is designed based on the process that uses multiple algorithms; SVM, Decision Tree, and ANN.

3.1 Description of Proposed Model Architecture

The proposed solution is NTD, which stands for Normal Traffic Detection. The architecture of this model consists of the following steps:

First: Implement data preprocessing techniques on a dataset, including data gathering and cleaning.

Second: Database called ABD, which stands for Abnormal Behavior Database; this database is built from any traffic that was classified as abnormal or malicious; this will help in classifying traffic so that if the incoming traffic behavior exists in the ADB, there is no need to proceed in the classification, which improves the efficiency and effectiveness of the proposed model "NTD" at the first run of the model NTD, ADB database will be empty, then as soon the model started " NTD " will store the malicious packet information and data on ADB database.

Third: dataset is being loaded into the model and stored in a CSV file.

Fourth: Network Packets from the data set will be checked to see if such a malicious pattern exists in the ADB database, which means that it was classified before and registered in the database.

Fifth: If such a pattern does not exist in ADB, traffic will be classified by three algorithms as the following sequence:

Classification using SVM algorithm, If the classification result is Normal, the process will continue to another **Decision Tree** algorithm. In the case of an anomalous classification result, the ADB will be updated by adding packet information. The same procedure is implemented with both decision trees and ANN.

If the classification result indicates that the traffic is “Normal,” the source IP address is waitlisted to permit traffic to the destination networks. This is done in conjunction with continuous monitoring of the source IP address and behavior.

Figure 3.2 includes the flowchart of the Detection of the normal traffic model (NTD):

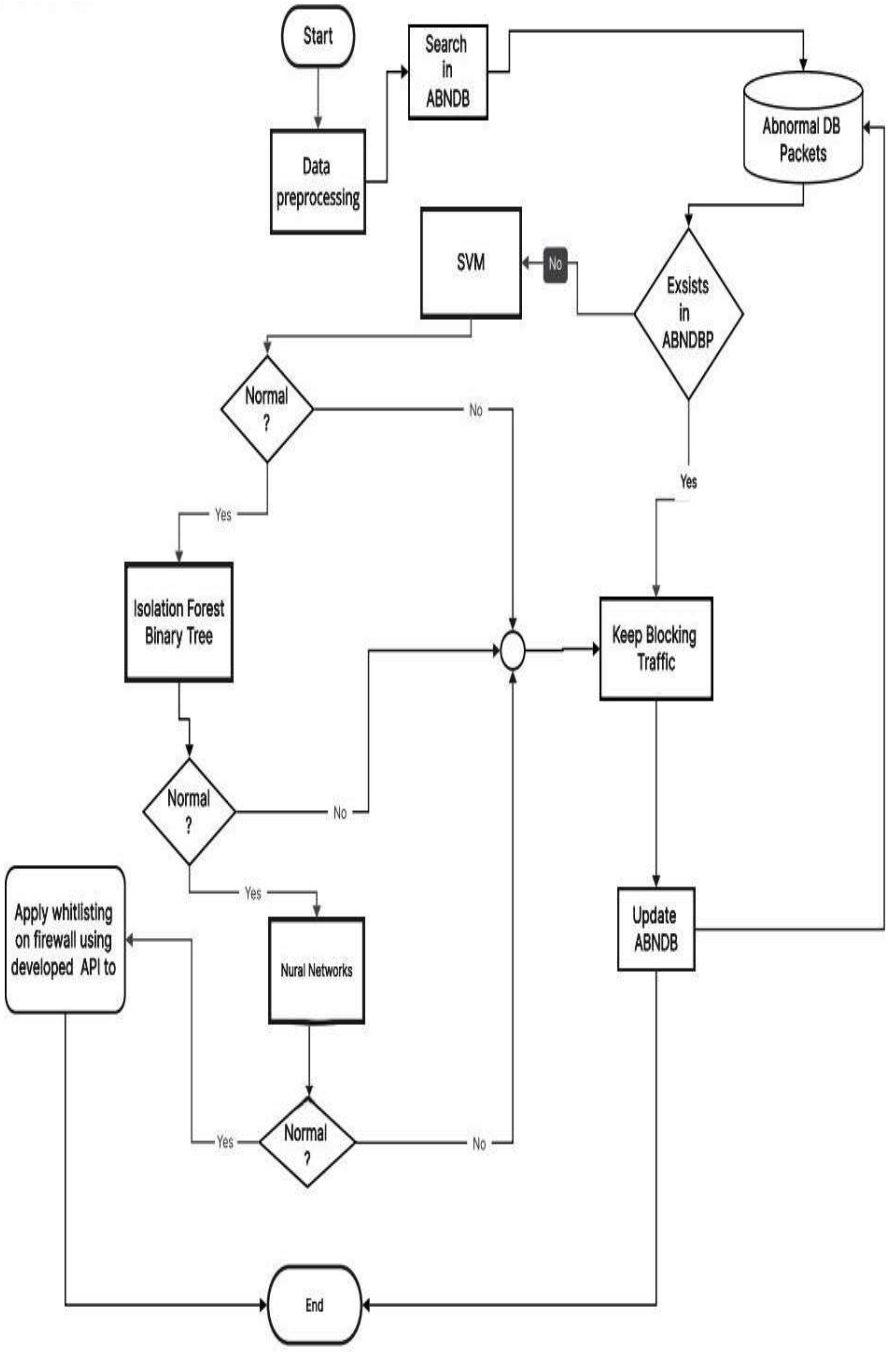


Figure 3.2: NTD Model Flowchart

3.1.1 Supervised Vector Machine (SVM)

Support Vector Machine (SVM) is a learning model that employs classification techniques to classify data into two groups (Valkenborg et al., 2023). The linear SVM model seeks out the hyperplane with the most significant decision boundary (Z. Liu et al., 2022); Figure 3-3 below illustrates the linear SVM. It looks for hyperplane **wx+b=0, where w is the weight vector, and x is the input vector.**

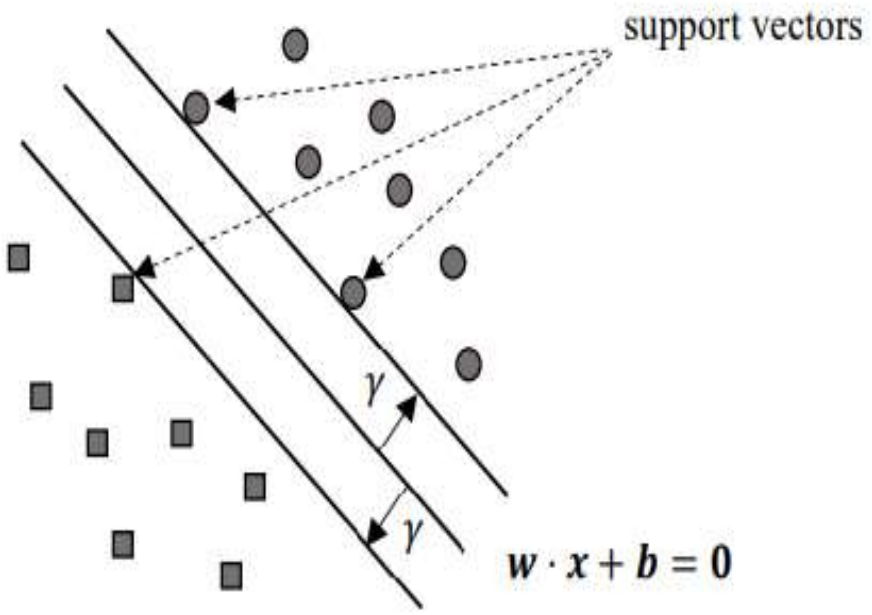


Figure 3.3 Linear SVM

SVM is widely used in high-dimensional classification due to its high accuracy in the evaluation. (Palanivinayagam & Damaševičius, 2023), also according to (Butt et al., 2023). SVM has perfect accuracy in detecting attacks, achieving high accuracy rates across various classification experiments. After SVM classifies data as normal or abnormal, another classification technique, the decision tree, is applied to normal traffic.

RBF is a popular choice for SVMs because of its high flexibility and ability to detect relationships between variables (S. Zhang et al., 2023).

Most of the research proved that the Redial Basis Function RFB is Effective for non-linearly separable data; it maps features into an infinite-dimensional space using Gaussian radial basis functions. It is the best kernel used in SVM Umaporn Yokkampon, Sakmongkon Chumkamon, Abbe Mowshowitz, Ryusuke Fujisawa, Eiji Hayashi (Yokkampon et al., 2021).in research Anomaly Detection Using Support Vector Machines for Time Series Data." Also, this research shows that using the RBF kernel improved the

validity and accuracy of anomaly detection. Also, Krishnaveni S. and Vigneshwar (Krishnaveni S.and Vigneshwar, 2020) showed that the RBF kernel has achieved the highest accuracy. The RBF kernel has been used in the experiments to execute the Supervised Vector Machine algorithm. Other research showed that RBF achieved high accuracy, more than 95%, on the KDD cup99 dataset, which makes RBF a good choice for SVM in anomaly detection (Almaiah et al., 2022).

The RBF kernel has two essential parameters: gamma and C (regularization parameter). The choice of these parameters can significantly impact the performance of an SVM model, making it necessary to tune them carefully.

Gamma: A Gaussian Kernel's parameter, gamma, determines the width of the kernel function; it is used to handle non-linear categorization ensemble learning. Gamma value calculation was used in previous experiments and research, such as research titled Prediction of phases in high entropy alloys using machine learning (Bobbili, 2023), which showed the enhancement in reduction of overfitting, the accuracy of predictions, and the generalizability of models.

Regularization parameter (C): Frequently represented by the letter C, a regularization parameter controls the trade-off between achieving a good fit to the training data and a simple decision boundary. It manages the compromise between reducing the classification error and maximizing the margin. While a larger C number reduces misclassification but may result in a less margin, a smaller C value permits a larger margin but may misclassify some points (Dehlaghi-Ghadim et al., 2023).

In this research, the C value will be selected first at value 1, then the performance will be evaluated. After that, the value will be tuned to reach the optimal value of C. The gamma ensemble method will be used to calculate the C value.

3.1.2 Decision Tree

A decision tree is a binary tree structure used to isolate instances; in detection, this binary tree method isolates the normal behavior traffic from abnormal instances (Munir et al., 2019) , this technique is based on outlier detection; Every tree is made by recursively splitting the instances, choosing an attribute at random, and splitting the value between the attribute's maximum and minimum values (Meira et al., 2020). It is recognized for its speed and user-friendliness; consequently, it has become widely utilized in the development of classification models, also it was proposed as a model for detecting anomalies in much research.

Machine learning decision trees provide an efficient method of making decisions by methodically describing the issue and considering all possible outcomes. The algorithm becomes more adept at predicting results for upcoming data as it processes more data (Z. Azam et al., 2023).

Decision Tree (DT) has shown better performance in previous studies such as Geeta Singh & Neelu Khare (Singh & Khare, 2022), Mahshid Helali Moghadam; Ali Balador; Hans Hansso (Dehlaghi-Ghadim et al., 2023).

Chauhan, Nagesh Singh has described a Decision Tree in his paper as a popular machine-learning approach for classification and regression tasks, it classifies instances by arranging them from the root to the leaf or terminal node, where the leaf or terminal node shows the instance's classification. Then, the nodes in the tree act as a test case for a specific attribute, and every edge dropping from the node represents potential solutions to the test case. This recursive procedure is applied to every subtree (Chauhan, 2022)

Also (Myles et al., 2004) described how they used a Decision Tree to classify network traffic into normal or anomaly. The characteristics of the supplied data create a hierarchical structure of judgments or rules. The tree's leaf nodes reflect the ultimate forecasts or results, whereas each interior node represents a judgment call or test on a particular attribute. Figure 3-4 illustrates the working principle of the decision tree:

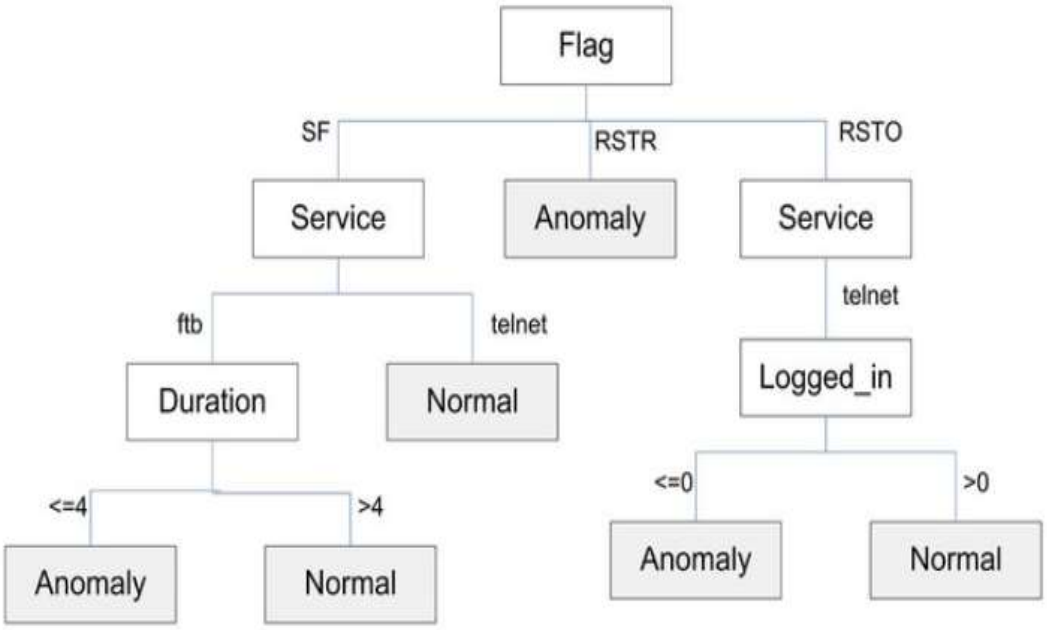


Figure 3.4: Decision Tree

3.1.3 Artificial Neural Networks (ANN)

The third level of classification applied in this proposal is Artificial Neural Networks NTD, so after SVM and DC have classified the traffic as normal, another level of classification takes place using Sequential ANN, it is another technique that classifies traffic into two classes. The traffic that was classified as normal from a binary tree will also be tested using neural networks. Sequential ANN has achieved excellent results in different cybersecurity solutions, for example vulnerability detection (Bilot et al., 2023), (Nguyen et al., 2022) , (Chen et al., 2020; Wei et al., 2021), and threat intelligence (Wei et al., 2021) . Moreover, malware detection has been widely and successfully applied to intrusion detection and prevention systems.

ANN Neural networks are strong models that can extract complex relationships and patterns from data. It can learn on its own and complete tasks that a linear program cannot do, also, its parallel architecture allows neural networks to function normally even if one of its components fails. (Revanesh et al., 2024). Described Neural Networks as a self-learning and do not require reprogramming. The fact that ANN learns from sample data sets is one of its important advantages (Shah & H Trivedi, 2012).

Architecture Artificial neurons, nodes, or units are arranged in layers and connected to form neural networks. The input layer, hidden layer(s), and output layer are the three primary types of layers into which these nodes are arranged. The input layer collects the input data, the hidden layers process it, and the output layer generates the results or predictions that are ultimately produced.

Weights and Bias: Each connection between two nodes is given a weight indicating the connection is strength. The associated bias of every node (aside from the input nodes) can change the node's activation threshold.

Feedforward Pass: The input data is transmitted through the network from the input layer to the output layer in a forward direction. The input values are multiplied at each node by the appropriate weights, added with the bias term, and then passed through an activation function. The network can learn intricate relationships to the non-linearity introduced by this activation function.

Backpropagation: The algorithm compares the predictions to the true values (in supervised learning scenarios) and determines an error or loss after the feedforward pass is finished. Then, the network generates predictions. The weights and biases in the network are then modified using a method known as backpropagation using this error. Backpropagation updates the weights and biases in a way that minimizes error by calculating the gradient of the loss concerning those parameters.

Repeating the backpropagation-based iterative adjustment of the weights and biases is called epochs or iterations. In order to reduce error and raise prediction accuracy, the neural network learns from the data and updates its parameters (C. M. Bishop & Nasrabadi, 2006).

3.2 Data Sampling: Collection and Gathering

An attempt was made to collect a dataset containing real traffic records; however, this effort failed because acquiring the necessary data required approval from management. Management denied the request due to an internal data classification policy that prohibits the use of such information by external parties, as this type of data is classified as restricted. To overcome this problem, datasets were collected from multiple sources, and then lists were merged to into a unified dataset that contains a list of malicious IP addresses and network traffic, where the dataset consisted of the following:

First: Part of the dataset that contains network activities and network flows collected from IP Network Traffic Flows Labeled with 75 Apps from Kaggle.com (Kaggle).

Second: Part of the data was collected from the Network Entity Reputation Database (NERD)

Third: Part of the data was collected from **NSL-KDD**Network Entity Reputation Database (NERD)

This dataset contains the Network Entity Reputation Database (NERD) , a database containing a list of malicious IP addresses worldwide. The NERD system collects data about the sources of cyber threats and creates an updated database of known malicious IP addresses. NERD includes hostnames, geolocation data, and information regarding malicious IP addresses, such as date and time of reporting, hostname, and geographical location (Bartoš, 2020). This database is dependent on several sources, such as:

First: Warden- CESNET: A database system sharing threat information; it collects data from reports from multiple security systems, such as NetFlow, honeypots, and other sources(CESNET, 2017) .

Second: DShield: A firewall log correlation system run by SANS institute (SANS Internet Storm Center, 2023).

Third: Blocklists: IT uses around fifty public "blacklists" from nearly twenty providers; these lists contain lists of suspicious and malicious IP addresses in a plain text format.

Fourth: AlienVault Open Threat Exchange (OTX) (Project, 2011): an open portal where security researchers and experts share millions of daily threat indicators (AlienVault, 2019).

Sixth: MISP is an open-source threat intelligence and sharing platform that includes IP addresses from events. Events tagged with "top: (Wagner et al., 2016)

NSL-KDD

A data set was used to solve some problems of the KDD'99 data set. It is a newer version of the KDD data, and it can be applied as an adequate benchmark dataset that helps compare various intrusion detection methods. In addition to that, the reasonable size of this dataset, which is around four million records, makes it affordable to run the experiments on the complete set without randomly selecting a small portion. Data files of NSL-KDD, used in this research, are described in the following Table 3-1

Table 3.1: Data Files of NSL-KDD

Data Files	Description
KDDTrain+.ARF	Full NSL-KDD train set with ARF format binary labels in ARF
KDDTrain+_20Percent.ARF	A 20% subset of the KDDTrain+.ARF file.
KDDTrain+_20Percent.ARF	A 20% subset of the KDDTrain+.txt file.
KDDTest+.ARF	Full NSL-KDD test set with binary labels in ARF format
KDDTest+.TXT	Full NSL-KDD test set including attack type and difficulty level in CSV format
KDDTest-21Percent+.ARF	A subset of the KDDTest+.arf file which does not include records with difficulty of 21 out of 21
KDDTest-21Percent+.ARF	A subset of the KDDTest+.txt file which does not include records with difficulty of 21 out of 21

3.2.1 Features Description

Table 3-2 describes all features in the NLS-KDD data set, and because the all the used datasets are extracted from network traffic, same features will be used when the experiment deals with other datasets such as:

Table 3.2 Features Description of the NLS-KDD Dataset

No.	Feature Name	Type	Description
1	duration	Continuous	Length (in seconds) of the connection.
2	protocol_type	Categorical	Protocol used (e.g., TCP, UDP, ICMP).
3	service	Categorical	Network service e.g., HTTP, FTP
4	flag	Categorical	Status flag of the connection
5	src_bytes	Continuous	Number of bytes sent (source to dest.)
6	dst_bytes	Continuous	Number of bytes sent (dest to source.)
7	land	Binary 0,1	IF connection is on the same host/port
8	wrong_fragment	Continuous	No. of wrong fragments

9	urgent	Continuous	No. of urgent packets
10	hot	Continuous	No. of accesses to sensitive data
11	num_failed_logins	Continuous	No. of failed login attempts.
12	logged_in	Binary 0,1	If connection is from a successful login
13	num_compromised	Continuous	No. of compromised conditions.
14	root_shell	Binary 0,1	If root shell was obtained
15	su_attempted	Binary 0,1	If "su root" command was attempted
16	num_root	Continuous	No. of root accesses.
17	num_file_creations	Continuous	No. of file creation operations.
18	num_shells	Continuous	No. of shell prompts invoked.
19	num_access_files	Continuous	No. of operations on access control files.
20	num_outbound_cmds	Continuous	No. of outbound commands in an FTP .
21	is_host_login	Binary 0,1	If the login belongs to a host
22	is_guest_login	Binary 0,1	If the login is a guest login
23	count	Continuous	No. of connections to the same host.
24	srv_count	Continuous	No of connections to the same service
25	serror_rate	Continuous	% of connections that have SYN errors.
26	srv_serror_rate	Continuous	SYN errors to the same
27	rerror_rate	Continuous	% of connections that have REJ errors.
28	srv_rerror_rate	Continuous	REJ errorsthe same service .
29	same_srv_rate	Continuous	% of connections to the same service.
30	diff_srv_rate	Continuous	% of connections to different services.
31	srv_diff_host_rate	Continuous	% of connections to different hosts.

32	dst_host_count	Continuous	No. of connections to the same dest.
33	dst_host_srv_count	Continuous	NO. of connections to the same dest
34	dst_host_same_srv_rate	Continuous	% of connections to the same dest.
35	dst_host_diff_srv_rate	Continuous	% of connections to different dest.
36	dst_host_same_src_port_rate	Continuous	% of connections with the same source port.
37	dst_host_srv_diff_host_rate	Continuous	% of connections to different destination
38	dst_host_serror_rate	Continuous	% of connections of SYN errors on dest.
39	dst_host_srv_serror_rate	Continuous	% of connections of SYN errors on dest.
40	dst_host_rerror_rate	Continuous	% of connections with REJ errors on dest.
41	dst_host_srv_rerror_rate	Continuous	% of connections with REJ errors on dest.

3.3 Data Preprocessing

Data is available in many forms and formats that the machine cannot understand as raw data extracted from devices, so it is essential to convert data into a form that is clear and readable by the machine

Data processing is an essential machine-learning step involving converting raw data into a format suitable for training and testing a machine-learning model. The quality and accuracy of the model is highly dependent on the quality and accuracy of the dataset used for training. The overall process of data processing plays a critical role in the success of machine learning models. The goal is to ensure that the data used for training is accurate, relevant, and representative of the real-world problem the model is designed to solve. Also, it is necessary to solve problems that affect consistency and may prevent data analysis (Maharana et al., 2022).

The preparation process starts after collecting and aggregating data; therefore, data should be prepared before building models. Data preparation is the basic stage in data processing; it is important to ensure high-quality data accuracy to have effective and high-accuracy results from data analysis. Data processing includes Data Cleaning, Data transformation, Feature Engineering, Handling data imbalance, Dimensions Reduction,

splitting data into training and testing, and Normalization and scaling (Raina & Krishnamurthy, 2022).

3.3.1 Data Cleaning

This process involves detecting and resolving errors or inaccuracies in the dataset. This includes handling missing fields, error correction, and removing duplicates. Regarding outliers' detection, we will not solve them here because, in traffic activities, the outliers indicate suspicious traffic or behavior (Tableau, 2016).

Python script was developed to apply data cleaning on the dataset, this includes removing duplicate records from the dataset. In addition, a manual inspection review technique was used for data validation. A script using Python was developed to achieve this mission.

After script execution, forty-two columns (features) were adopted, which is the same number of actual fields in the dataset.

3.3.1.1 Missing Values

As for the Identification of missing values, the method must be used to treat null values, the missing data has been corrected using the Mean/Median/Mode Imputation method, which replaces missing values with the mean, median, or mode of the observed values in the variable (Li D. Z., 2021) .

3.3.1.2 Error Correction

The error correction process was then executed, and the script used several functions to correct the errors.

3.3.1.3 Removing Duplicates

Because duplicate data negatively affects the evaluation metrics and analysis, the removing duplicates process aims to remove all duplicates in the dataset. Therefore, this process is considered one of the essential steps in the data cleaning process. since when there is a high rate of repetition of data records in the dataset, it will reduce the generality of the results, as the selected features may be overfit to classes or instances with more repetitions. As a result, this will lead to poor accuracy of the results at the evaluation stage (Yin et al., 2023). After this step, all duplicated data was removed, and the number of duplicates removed was 30000.

3.3.2 Data Transformation

Data Transformation involves converting raw data into an appropriate format for analysis. In this process, a Python script executes all its components: standardizing units of measurement, scaling numerical values, Encoding, and Normalization. Standardizing units of measurement.

Label Encoding

The Label Encoding technique was applied to non-numerical values that include protocol type (TCP, UDP) and Service (HTTP, HTTPS, …) in addition to label, which implies whether the traffic is normal or abnormal.

3.3.2.1 Scaling of Numerical Values

Z-score normalization was used to scale numerical values. This technique preserves the same shape of the distribution of values and puts features to a similar scale (zero mean and unit variance). Also, it is suitable when there are different scales of feature values. In addition, considering that outliers may be anomalies, the outliers will not be affected, and it is suitable for spotting outliers (Chikodili et al., 2020)**.**

3.3.2.2 Feature Engineering

This step focuses on creating new features from the existing features in the dataset to provide additional information. This process is useful for training the model (Patel, 2021) Because the solution is to investigate traffic from malicious IP addresses, a blocked source IP address feature was added to the KDD cup dataset; this list of IPs was collected from intelligence tools, as mentioned in the data gathering and collection section.

3.3.3 Splitting Data

This process involves dividing the dataset into two subsets, training to train the model, and testing to test it. An 80-20 ratio was used, with 80% of the data used as a training dataset and 20% as a test dataset.

3.3.4 Encoding Categorical Variables

Categorical variables are defined as variables that can assume a finite number of discrete values. As machine learning algorithms necessitate numerical data for processing, it is essential to encode these variables into numerical formats before their use in training. In this context, the Label Encoding technique was employed to convert non-numerical values, including protocol types (such as TCP and UDP), service types

(such as HTTP and HTTPS), and the categorical label indicating whether the traffic is classified as normal or abnormal. This encoding process facilitates the transformation of categorical values into numerical representations, effectively enabling their integration into machine learning models.

3.4 Model Building

After the dataset has been collected and processed to make it ready to be fit into the suggested module NTD, the model-building phase starts; the model-building approach contains four main steps:

First: Design the module which consists of:

- Import the required libraries, classes, and modules.
- Development of the three functions used in the NTD model: SVM, Decision Binary Tree, and ANN.

Second: Evaluate the implemented algorithm separately,

Third: Evaluate the NTD model to find out its efficiency and effectiveness

Fourth: Compare the evaluation metrics between each algorithm and the NTD model.

3.5 Using Ensemble Learning in Model Development

Algorithms have different strengths and weaknesses, so using multiple algorithms in the model can help understand patterns, behaviors, and relationships that might not exist using a single algorithm.(Cvitić et al., 2021).

Combining multiple algorithms into a hybrid detection model effectively leverages the unique strengths of each technique, leading to significant improvements in detection accuracy and a substantial reduction in false positive rates. This approach not only merges various algorithms and models to create a powerful system capable of addressing a wide range of data patterns but also demonstrates an exceptional ability to adapt to emerging threats in real-time. The synergy achieved through this integration eliminates the weaknesses of individual methods, resulting in a comprehensive and robust detection solution. This ultimately enhances trustworthiness and effectiveness in identifying unusual behaviors within data sets.(Olateju et al., 2024)

Ensemble learning is a technique that combines multiple algorithms to overcome their limitations and leverage their strengths. For instance, DT may perform well in certain situations and combining them with SVM or ANN can improve overall performance, especially in complex classification tasks (Acito, 2023).

NTD utilized a stacking ensemble method using SVM, DT, and ANN algorithms. It involves training these algorithms on the same dataset and using their predictions as inputs to make the final prediction. In addition to the above Ensemble Method has many advantages:

3.5.1 Improve Accuracy

Various algorithms have their own strengths and weaknesses. By using multiple algorithms in a model can help overcome the limitations of any single approach. For example, while decision trees may be effective in certain scenarios, combining them with other algorithms like SVM or ANN can enhance overall accuracy, especially in complex tasks such as classification.

3.5.2 Reducing Bias

Depending on a single algorithm when building a model can result in bias. Sometimes the algorithm may overfit or underfit the data. Using multiple algorithms reduces this risk and achieves better generalization of new data (Doganer, 2021).

3.5.3 More powerful

Combination of several models into a single one is more powerful, and accurate prediction .

3.5.4 Robustness and Reliability

In scenarios where the data is noisy or incomplete, multiple algorithms can provide more reliable results by cross validating each other's predictions, leading to more robust conclusions.

3.6 Model Evaluation

A confusion matrix was used to evaluate each algorithm separately and the proposed solution NTD. The evaluation process consists of three stages:

First, evaluating each algorithm alone,

Second: Evaluating the NTD model

Third: comparing the results.

Performance metrics have been used which will effectively measure four values:

model’s precision, accuracy, recall, and F1-Score (Bohutska, 2021).

Accuracy = $\frac{TP+TN}{\mathrm{T\,P + F\,P + T\,N + F\,N}}$

Recall: It measures actual anomalies that were detected by the application of the solution and algorithms; it is calculated below equation:

Recall = $\frac{\boldsymbol{TP}}{\mathbf{T\,P + F\,N}}$

The precision determines cases that are identified as anomalies are true anomalies; it is calculated using the below equation:

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

F1 Score calculates the overall performance of the model by depending on both Recall and Precision; the below equation calculates the F1 score:

$\mathrm{F1 - Score} = \frac{2}{\frac{1}{\mathrm{P\,recision}} + \frac{1}{\mathrm{Recall}}}$

Where FP = False Positive.

FN = False Negative.

TP=True Positive.

TN=True Negative (Muntean & Militaru, 2023)

True Positives (TP) refers to the total number of samples correctly expected to be positive.

True Negative (TN): Refers to the count of instances accurately predicted as negative.

False Positives (FP) refers to the number of samples incorrectly expected to be positive.

False Negatives (FN) represent the number of samples that were incorrectly predicted as negative.

3.7 Summary

The NTD (Network Traffic Detection) model utilizes a systematic approach encompassing various stages, including data collection, preprocessing, model construction, evaluation, and deployment. This methodology is designed to overcome the limitations of earlier techniques, notably the incidence of false positives, by incorporating advanced algorithms such as Support Vector Machine (SVM), Decision Tree, and Artificial Neural Network (ANN). To facilitate efficient classification, abnormal traffic data is systematically logged in an Abnormal Behavior Database (ABD). In instances where a traffic pattern does not exist within the ABD, the data undergoes a sequential classification process utilizing the three algorithms. Furthermore, normal traffic is

subjected to a whitelisting procedure for continuous monitoring. The NTD model draws upon diverse datasets from reputable sources, including Kaggle's network flows, NERD, and NSL-KDD. The preprocessing phase is meticulous, featuring essential steps such as data cleaning, encoding, and feature engineering to enhance data quality. Ensemble learning techniques are employed to synergize the different algorithms, thereby boosting both accuracy and robustness, which is crucial for the reliable classification of normal versus malicious traffic. The forthcoming chapter, titled "Experimental," will provide an in-depth examination of the implementation and evaluation of the NTD model.

Chapter Four: Exploratory Data Analysis

4.1 Data Preprocessing

The processes mentioned above have been applied to data before starting to execute the algorithms and building the model.

4.2 Dataset Properties

The experiment in this research used a subset of the main NLS KDD dataset which contains 494021 records, the subset size was 125972 records, 25.5% of the main dataset, the reason of using part of dataset was due to limitation of resources, since many failed trials have been done to obtain results from main dataset, and to address the limitation of resources issue, the approach was to reduce the size of the dataset gradually until results were obtained. with the number of features 41 columns as they represent attributes.

4.3 Traffic Distribution

Distribution of Traffic in Terms of Normal or Abnormal

The selected data set contains 125972 with 42 features, 67342 records with 53.46 % of data classified as Normal traffic, while 58630 records with a percentage of 46.54% are classified as attacks. Figure 4.1 below shows the distribution of traffic in terms of attack type, i.e., normal or attack.

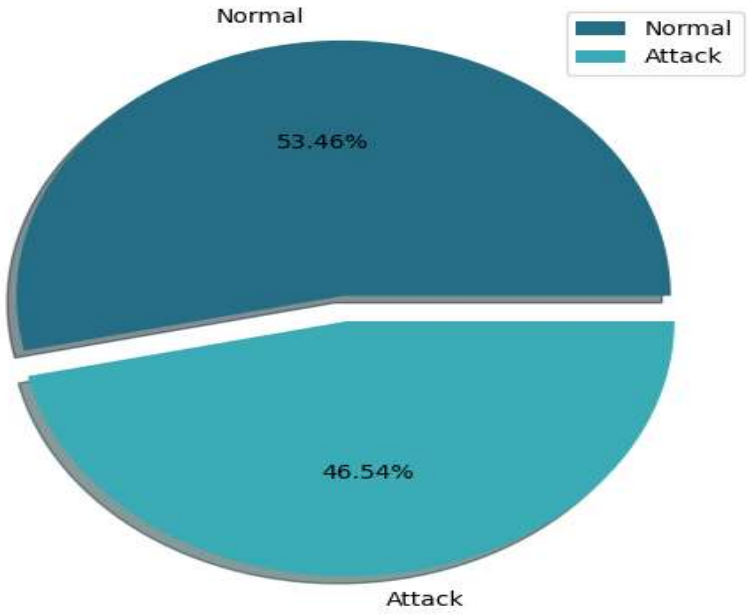


Figure 4.1: Label Distribution of Traffic

4.3.1 Distribution of Traffic Type(Class)

Table 4-1 below shows the traffic distribution according to the traffic's nature. Normal or abnormal over the main dataset of records:

Table 4.1 Attack Type Distribution

Label	Count	Label	Count
normal	67342	warez master	20
neptune	41214	land	18
satan	3633	imap	11
ipsweep	3599	rootkit	10
portsweep	2931	loadmodule	9
smurf	2646	ftp_write	8
nmap	1493	multihop	7
back	956	phf	4
teardrop	892	perl	3
warezclient	890	spy	2
pod	201	buffer overflow	30
guess_passwd	53		
Total			125972

4.3.2 Classification of Data into Two Categories

Because the module focuses on only two classes, data labels in the Labels column have been changed into two classes, Normal for normal behavior, and Attack for not normal, so that data has been classified into two categories (Normal, Attack), any labels do not equal “Normal” will be considered as label of attack, then we obtained the result in below table 4-2:

Table 4.2 Classification of Traffic Normal/Attack

Labels	**Count**
Attack	58630
Normal	67342
Total	125,972

Figure 4.2 shows the distribution of attack type and count:

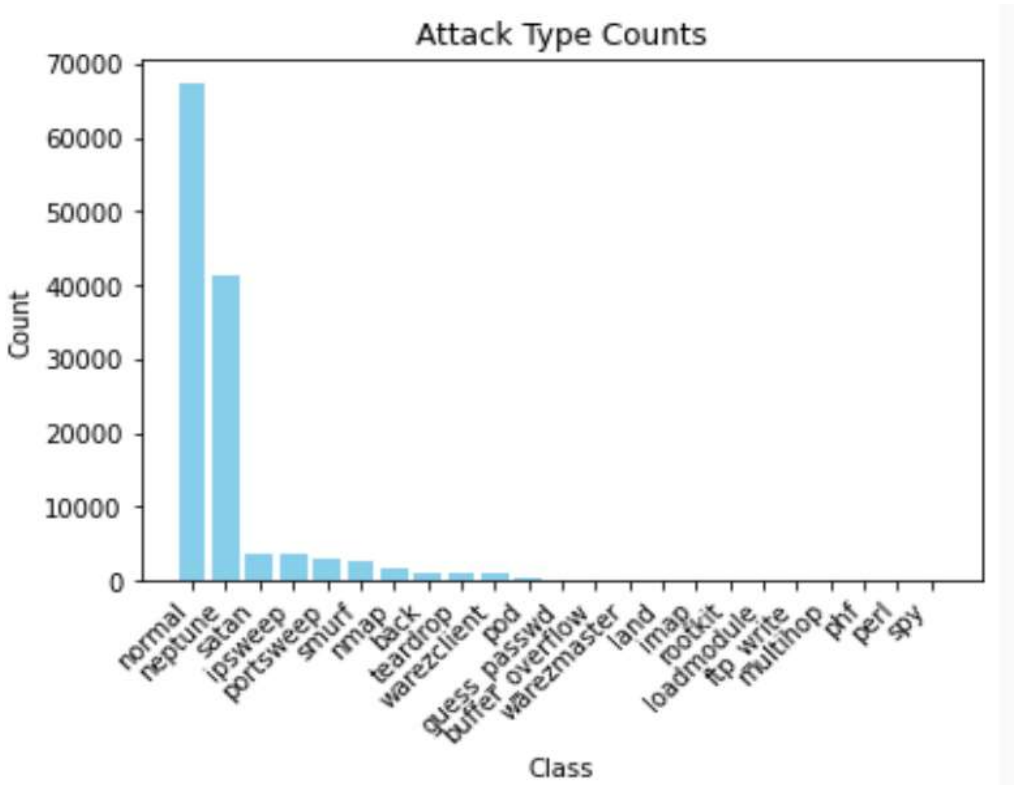


Figure 4.2 Distribution of Attack Types

4.3.3 Feature Distribution in Dataset

For the purposes of data analysis and understanding the relationships between features in the dataset, a statistical analysis method that studies dependencies and relationships between features in the dataset (Kneusel, 2021) was implemented. In other words, any change of one value will affect the value of the other.

The below Figures shows some samples of features distribution, each figure has pair of features, such as :

dst_host_srv_rerror_rate' and Frequency, 'srv_serror_rate' and frequency, Labels (Class) and Srv_rerror_rate.

Figure 4-3 below shows a visualization of these features:

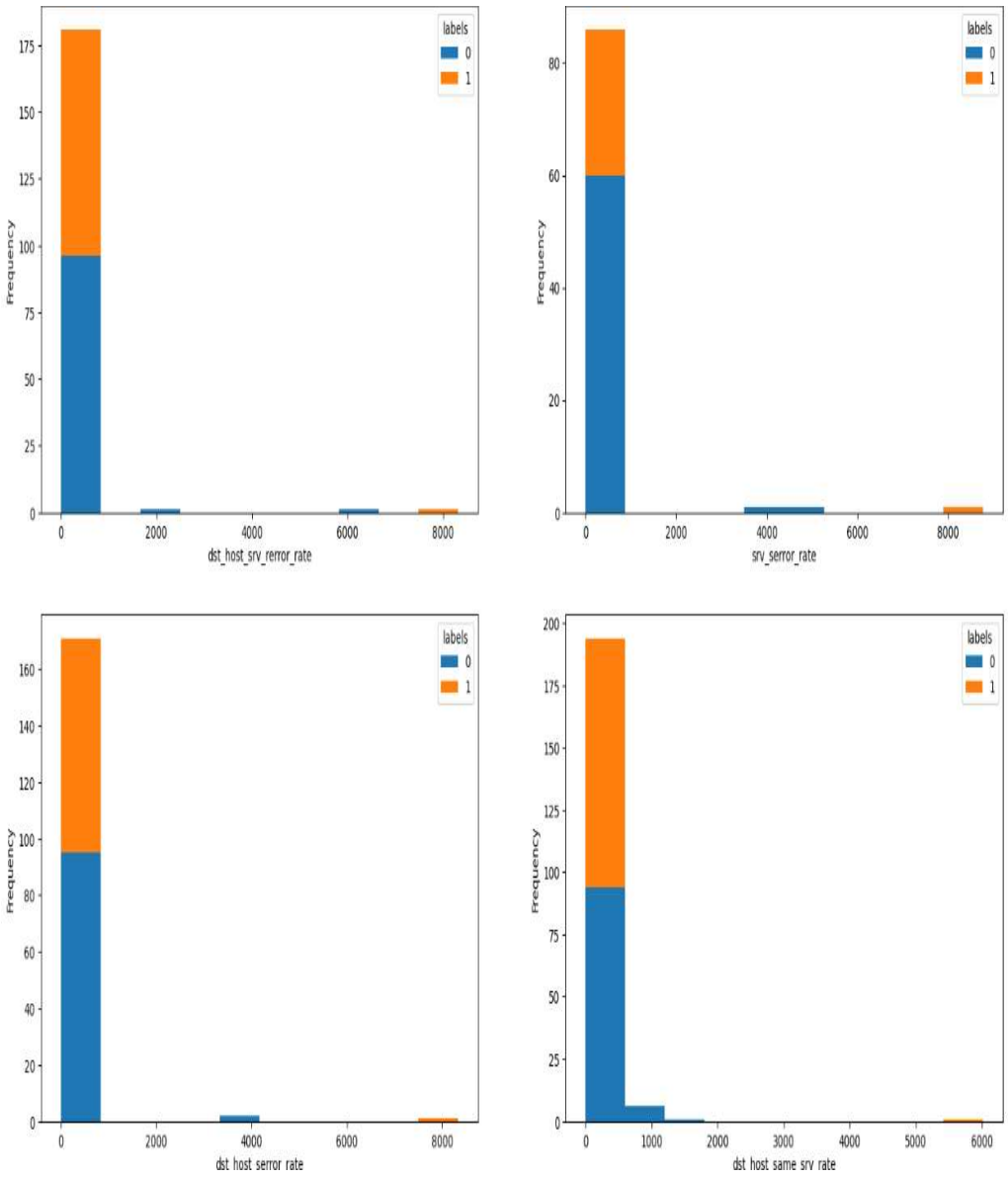


Figure 4.3 Features Distribution in Dataset

Other examples of feature's distribution are shown in below graph in figure 4-4 shows the correlation between **Labels (Class)** and **Srv_rerror_rate** which indicates normality of traffic and **Srv_rerror_rate** feature.

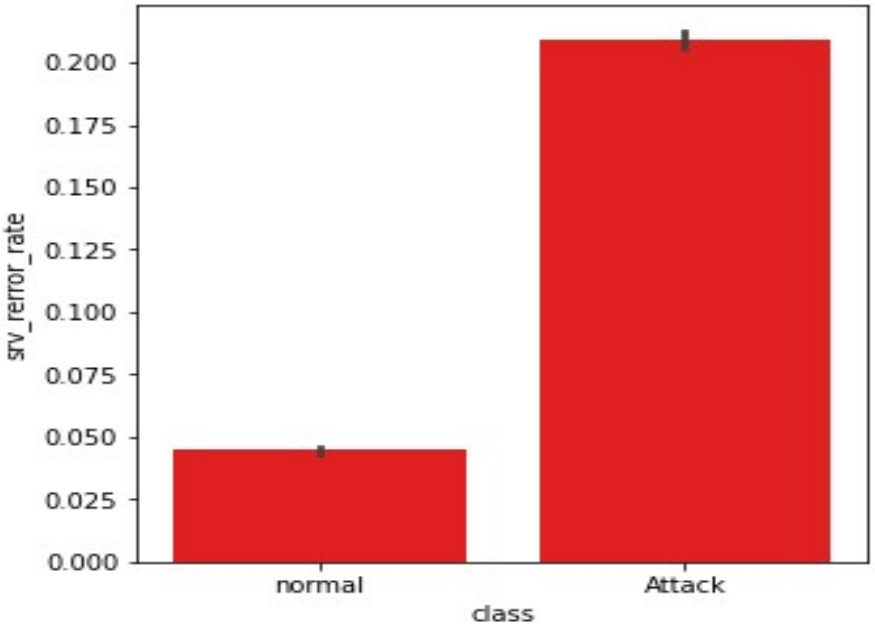


Figure 4.4 Class Distribution on Srv_Rerror_Rate and Labels

Distribution of protocol type and volume is illustrated in the below figure 4-5

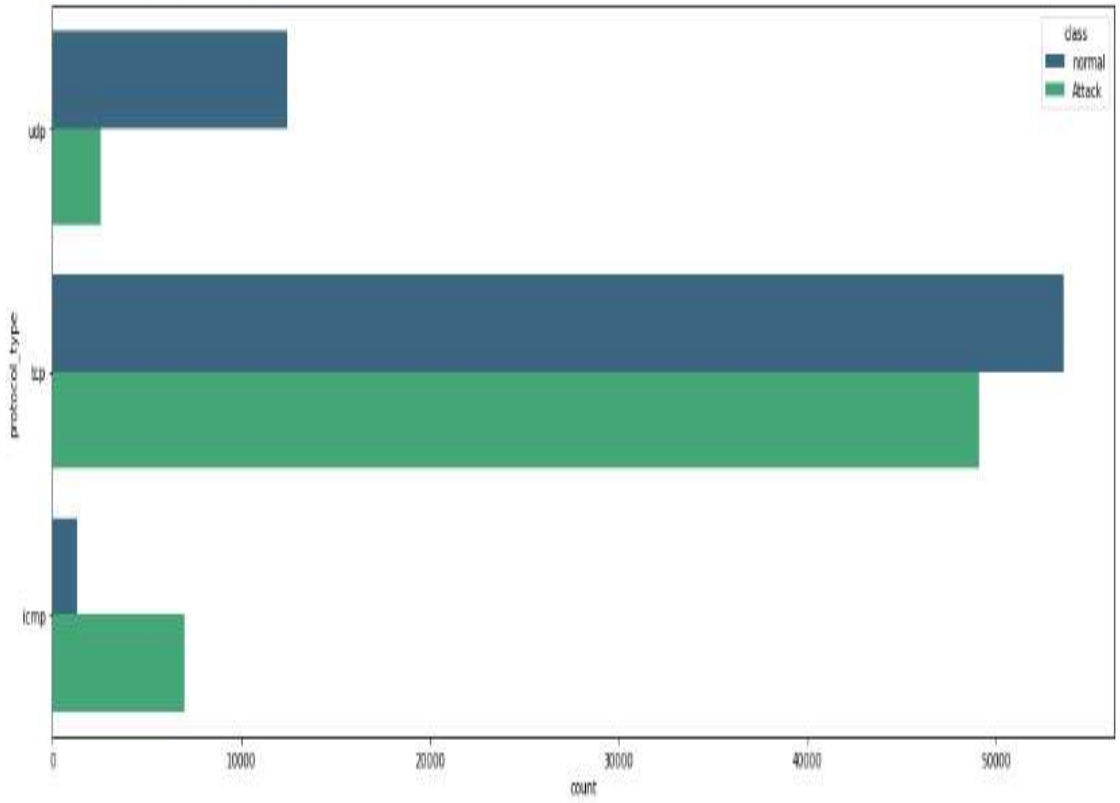


Figure 4.5 Traffic Label in Terms of Protocol Type and Volume

4.4 Summary

This chapter presented a comprehensive overview of the experimental setup, detailing the datasets employed, the preprocessing methods applied, and the procedures for model training and testing. It explicitly defines the performance metrics utilized to assess accuracy, precision, recall, and the overall effectiveness in differentiating between normal and malicious network traffic. The subsequent chapter discussed the results in depth and compared the performance of each algorithm when applied independently with that of the NTD model, evaluating the performance enhancements achieved through the ensemble approach.

Chapter Five: Results

5.1 Processing of Data in the Dataset

The dataset underwent a data preprocessing phase. This includes removing duplicates and applying Mean imputation, a process that compensates null values with the Mean value of the same feature (Rosenthal, 2017). Then, to compare the evaluation metrics for each algorithm separately with the proposed NTD model, the methodology is to apply SVM, Decision Tree, and ANN separately on the same dataset.

This researcher has adopted Accuracy, F1, Recall, Precision, and elapsed time as evaluation metrics. However, the main evaluation metric in the research experiment focused on is the F1 score as the main; this is because the F1 score is a combination of other tow metrics, Recall and Precision, as mentioned in Equation:

F1= 2* (Precision * Recall)/ (Precision + Recall) (Ismail & Wediawati, 2023), this equation shows that F1 score represents the harmonic mean of precision and recall, balancing the tradeoff between them.(M. Mahmud et al., 2023).

5.2 Applying Each Algorithm Separately

The study applied three algorithms separately. It calculated the number of malicious packets, elapsed time, and all values of evaluation metrics, including accuracy, perception, recall, and F1 score.

The first step in the experiment is to apply SVM to the main dataset. The section below will discuss SVM and the metrics' results analysis.

The initial attempt at the experiment using the main dataset was unsuccessful due to hardware limitations. The primary dataset contains approximately 500,000 records, and to address this problem a smaller subset of the main dataset was extracted, which included 125972 records with 41 out of 42 features as IP address feature was removed .

5.2.1 SVM Algorithm

The implementation of the SVM algorithm contains the following steps:

Splitting datasets into training and testing datasets with a ratio of 80:20 with 100777 records training dataset and 25195 testing dataset, this ratio is set based on a review of previous papers and books that shows that this ratio will be a good choice for dataset splitting and enough for training and testing modules for such as (A.- Mahmud & Shimada, 2023)and (Alharbi et al., 2023).

The table shows the confusion matrix for the SVM algorithm over the first dataset; below table 5-1shows the confusion matrix for the SVM algorithm:

First Confusion Matrix SVM:

Table 5.1 Confusion Matrix for SVM

Actual Negative	**49128**	**708**
Actual Positive	**622**	**49309**
	Predicted Negative	**Predicted Positive**

- True Positives (TP): 49309
- False Negatives (FN): 708
- False Positives (FP): 622
- True Negatives (TN): 49128
- Accuracy = 98.9%

The result in the table below showed that SVM achieved very high evaluation metrics on the customized dataset; Table 5-2 illustrates the results:

Table 5.2 Evaluation Metrics for SVM Algorithm

Evaluation Metric	**Value**
F1 Score	0.990
Recall	0.986
Precision	0.9879
Accuracy	0.989
Total Elapsed Time	23.8165 seconds

The above values of evaluation metrics F1, Recall, Precession, Accuracy, and total elapsed time indicate that SVM is a very good choice for data classification and anomaly

detection in network traffic, especially for F1 score, but it did not achieve optimal elapsed time.

The reason for the high elapsed time value is that SVM tries to find the hyperplane that maximizes the margin between different classes. This involves solving a complex quadratic optimization problem, especially when there are many features and data points. For large datasets, this optimization becomes computationally expensive.

5.2.2 Decision Tree algorithm.

A decision tree is considered one of the good algorithms used for anomaly detection. Various papers concluded that decision trees achieved high detection accuracy and better processing time (S. R et al., 2023).the execution of the decision indicated a high F1 score.

The results also showed high accuracy. Table 5-3 below shows the confusion matrix for the Decision Tree algorithm over the dataset.

Confusion Matrix:

Table 5.3 Confusion Matrix for Decision Tree

		Actual Value
Predicted Value	**50000**	**5874**
	505	**36,841**

Evaluation metrics: As shown in the results in Table 5-4 below, the decision tree achieved very high evaluation metrics over the dataset:

Table 5.4 : Evaluation Metrics for Decision Tree

Evaluation Metric	Value
F1 Score	0.920
Recall	0.991
Precision	0.851
Accuracy	92.5%
Elapsed time	0.78 seconds

First, a high F1 score demonstrates that the decision tree is highly precise and recall, which means it balances false positives and false negatives well. In this case, the F1 score of 94.1% is very high.

Second: Recall, which is sensitivity or true positive rate, measures the percentage of the actual positive cases identified correctly by the Decision tree. A high recall value indicates the decision tree's effectiveness in capturing most of the positive cases. For example, a Recall value of 99.4% indicates that the model correctly identified about 99.4% of the actual positive cases.

Third: Precision measures the proportion of correct identifications. A high precision value indicates that the decision tree has few false positive errors. The value of 0.89.3% indicates that about 0.89.3% of the positive predictions made by the model are correct. Figure 5-1 below demonstrates the scores of evaluation metrics:

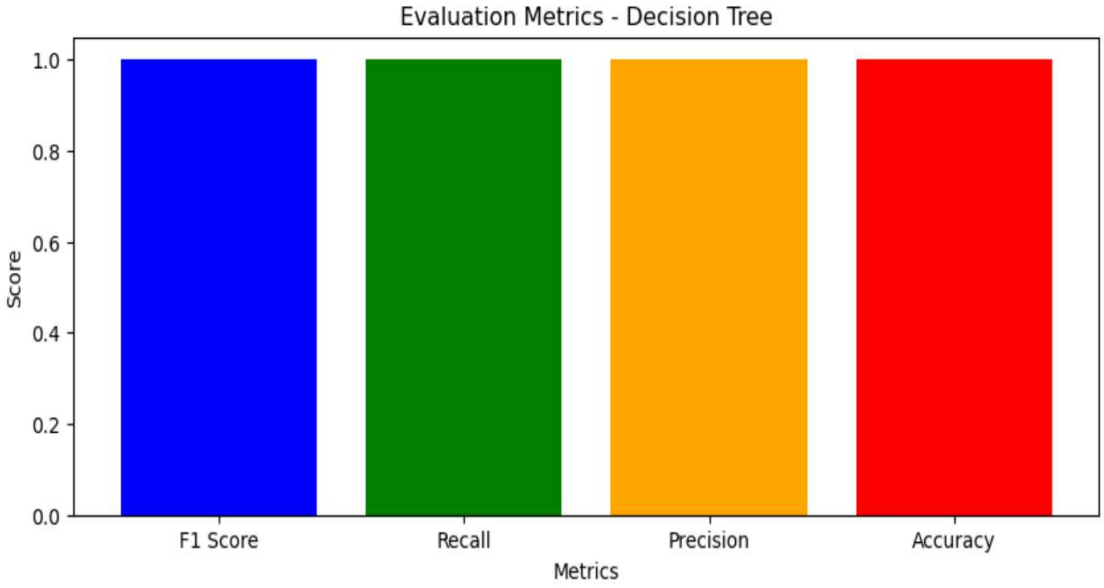


Figure 5.1 Scores of Evaluation Metrics of the Decision Tree.

5.2.3 Artificial Neural Networks (ANN)

Like SVM and decision trees, ANN is considered a good algorithm for anomaly detection. The results showed very high evaluation metrics, as illustrated in the confusion matrix in table 5-5:

Confusion Matrix:

Table 5.5 Confusion Matrix for ANN Algorithm

	Actual Value	
Predicted Value	**60000**	**241**
	120	**38707**

As shown in the below table 5-6 the evaluation metrics achieved very high values, which indicates nonlogical results.

Table 5.6 Evaluation Metrics for the ANN Algorithm

Evaluation Metric	**Value**
F1 Score	0.997
Precision	0.996
Recall	0.998
Accuracy	0.987
Total Elapsed Time	47.37649965286255 seconds

These results indicate that the sequential ANN algorithm is a good choice for data classification solutions. In addition to that, many papers concluded and recommended the use of ANN as a classifier of anomaly detection such as Nebrase Elmrabit; Feixiang Zhou; Fengyin Li; Huiyu Zhou in their paper Evaluation of Machine Learning Algorithms for Anomaly Detection (Elmrabit et al., 2020) and Mimoun Lamrini, Mohamed Yassin Chkouri, and Abdellah Touhafi in the paper Evaluating the Performance of Pre-Trained Convolutional Neural Network for Audio Classification on Embedded Systems for Anomaly Detection in Smart Cities (Lamrini et al., 2023)

Figures 5-3 visualizes the evaluation metric, which shows high values of these metrics, which indicates that it can be considered a good choice for classification. It shows a high rate of model Accuracy in both Training and validation, as shown in Figure 5.3

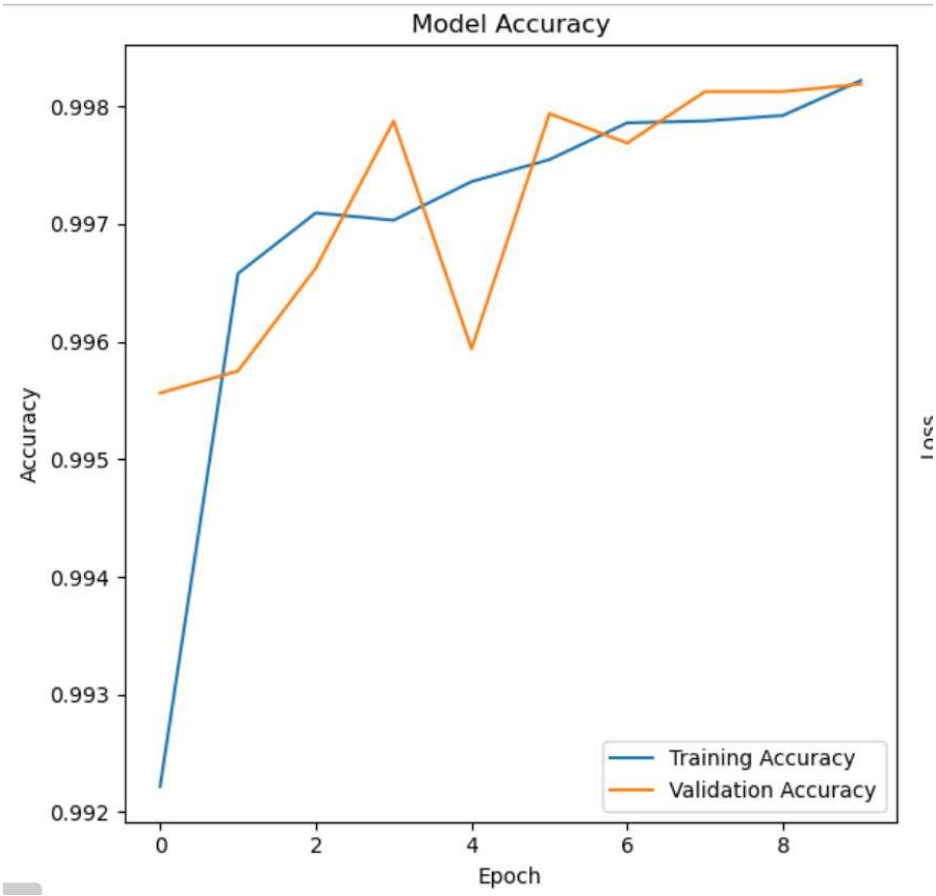


Figure 5.2 Accuracy in ANN Algorithm for Training and Validation

The result of loss indicates that the algorithm achieved a very high evaluation, which supports that it can be considered a good choice to classify the data; below, Figure 5-4 illustrates the low loss values in both training and validation.

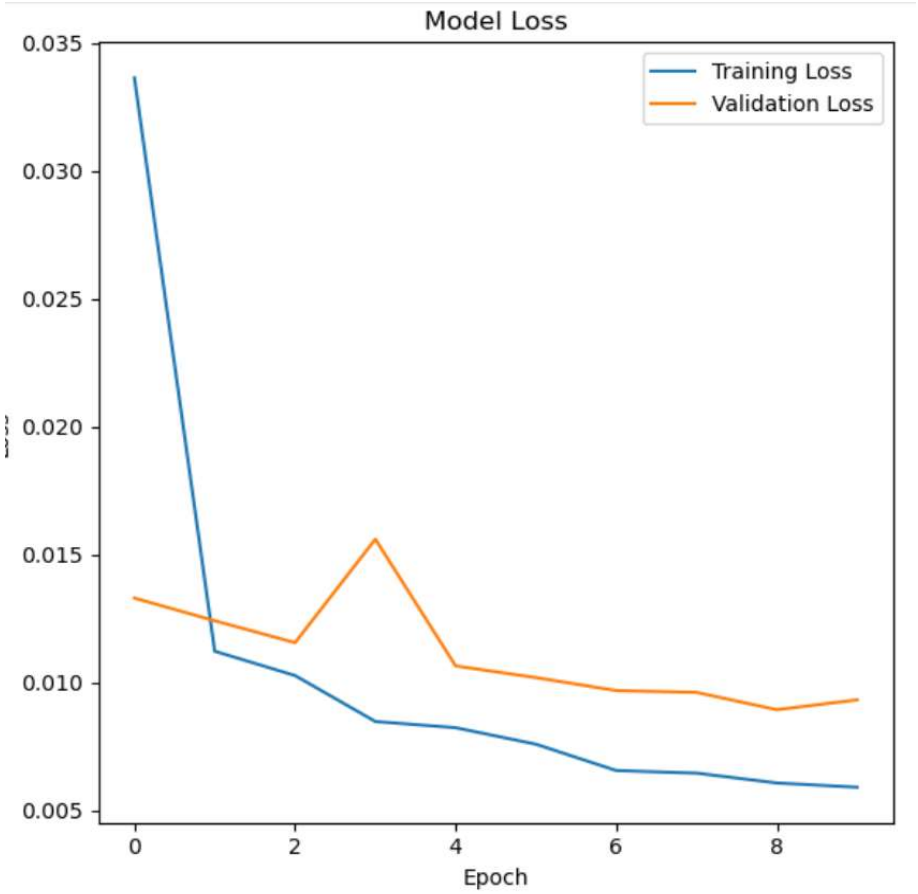


Figure 5.3 Percentage Loss for ANN Algorithm

5.3 Normal Traffic Detection Model (NTD)

The research has proposed a solution called the Normal Traffic Detection Model, abbreviated by (NTD) to address the problem mentioned in the problem statement. It consists of several steps; the below paragraph describes the algorithm that applies triple classifications after data preparation for classification:

First, Prepare the data for classification; this includes preprocessing and dividing the dataset into training and testing datasets.

Second, a knowledge base database called Abnormal Database Packets (ADP) contains a repository of traffic that has been detected before and classified as malicious (abnormal) traffic. By reducing detection and classification time, this database helps improve the NTD performance, efficiency, and effectiveness.

Third: steps on how the solution works, below figure 5-5 shows the model algorithm.

Step 1: Read traffic.

Step 2: Search for the behavior in the ABD database.

Step 3: If behavior exists in ABD, then.

Classify the traffic as malicious, do nothing, and exit

Else, classify traffic using **the SVM** algorithm.

If the is classified as malicious, then.

classify the traffic as malicious

Add traffic details to the ABD database, do nothing, and exit

Else classification is Normal:

classify traffic using a **Decision Tree**

If the classification is Malicious, then.

Add traffic details to the ABD database, do nothing - exit

Else classification is Normal, then:

Classify traffic using neural networks.

If the Classification is Malicious, then:

Add traffic details to the ABD database; do nothing-exit

Else classification is Normal: do the following:

Grant access to source IP

Build an API to add the source IP in allowlisting.

Keep Monitoring the traffic.

Figure 5.4 NTD Model Algorithm

5.4 Evaluation of Proposed Solution

First Dataset:

The main dataset source is a collection of aggregated datasets described in the above section 4.2. first dataset was extracted randomly from the main dataset, it contains 125972 out of 494021 records of normal and malicious traffic. Then dataset was splatted into training and testing. A common split ratio 80% training and 20% testing, was used, therefore training dataset contains 100777 records and testing dataset contains 25195 records the result showed a very high metrics that proved the hypothesis and answer the questions, and the model achieved the heist values of evaluation. The results in Table 5-7 below show very high metrics' values, proposed NTD.

5.5 Evaluation Metrics for NTD Model over Dataset 1

5.5.1 Confusion Matrix

The confusion matrix of the model over training is shown in table 5-7 below:

Table 5.7 Confusion Matrix of NTD

	Actual Value	
Predicted Value	**50000**	**44**
	38	**50620**

Table 5.8 Comparison of Results of all Algorithms Over Dataset 1

Evaluation metrics	**Results**
F1 Score	0.99912
Precision	0.99913
Recall	0.9992
Accuracy	0.9992
Total Elapsed Time	9.154 seconds

5.5.2 Comparison between Algorithms and NTD Over Dataset 1

The results shown below, comparison Table 5-8 below clarifies the difference in evaluation metrics between the three used algorithms separately and the NTD proposed solution:

Table 5.9 Comparison of Results of all Algorithms Over Dataset 1

Algorithm	**F1**	**Precision**	**Recall**	**Accuracy**	**Elapsed Time**
Proposed Model (NTD)	0.99912	0.99912	0.99924	0.99924	9.154 s
SVM	0.998	0.987	0.985	0.989	23.816 s
Decision Tree	0.998	0.998	0.998	0.998	0.7768 s
ANN	0.997	0.996	0.998	0.996	47.376 s

Despite the existing problems in the dataset, the above comparison table shows that the proposed solution (NTD) achieved the highest F1 score with 0.999 scores. At the same time, SVM has the lowest score of 0.998; also, in the other metrics, in precision, NTD has

0.999, while ANN has the lowest value of 0.997, in recall with the highest value of 0.999 and SVM with the lowest value of 0.986, and accuracy with 0.999 while SVM is the lowest with 0.989. Regarding the total elapsed NTD, it failed in achieving the optimal elapsed time; while Decision Tree has applied the classification with the lowest time of 0.776 seconds, the proposed solution has the second value with 9.154 seconds. The time is long due to calculations and sequential implementation of more than one algorithm. Also, ANN has the highest elapsed time with 47.376 seconds, which indicates that ANN needs more time to process the classification.

Second Dataset

According to above results, the problem has been raised on the dataset, since is shows illogical results, and to check the generalization and transformation of the model (NTD) and to ensure the results of the study, the same approach was implemented on another dataset of 145000 extracted from the main dataset , 80% of dataset was training dataset with 116000 records and 29000 record testing dataset, considering data preprocessing to prepare the dataset for implementation, taking into consideration the checking existence of overfitting, duplicate records, imbalanced data, empty records, and other actions, the results also showed excellent and logical results, below sections shows results :

SVM Algorithm:

Below, Table 5-9 shows the confusion matrix for SVM over Dataset 2:

Table 5.10: Confusion Matrix for SVM over Dataset2

		Actual Values
Predicted Value	**57,900**	**2,370**
	3,756	**47,062**

The results of evaluation metrics for SVM over dataset2 are shown in figure 5.6:

Table 5.11 Evaluation Metrics of SVM over Dataset2

Metric	Results
F1 Score	0.948
Precision	0.960
Recall	0.939
Accuracy	0.944
Total Elapsed Time	15.963 seconds

Decision Tree:

Confusion Matrix:

Table 5-12 below shows the confusion matrix for the Decision tree

Table 5.12 Confusion Matrix for Decision Tree

	Actual Values	
Predicted Values	**30,000**	**3578**
	9312	**22310**

True Negatives (TN): The results of evaluation metrics are shown in Table 5-13 below shows the results of evaluation metric od Decision Tree:

Table 5.13 Evaluation Metric of Decision Tree over Dataset 2

Metric	Results
F1 Score	0.823
Precision	0.893
Recall	0.763
Accuracy	0.802
Total Elapsed Time	0.428 seconds

ANN Algorithm:

Confusion Matrix:

Table 5-14 below shows the confusion matrix for the ANN algorithm:

Table 5.14 Confusion Matrix for ANN Over Dataset2

Predicted Values	**Actual Values**	
	48000	**1969**
	2467	**44264**

The values of evaluation metrics are shown in Table 5-14 below, which indicates high values of the ANN algorithm over the second dataset:

Table 5.15 Evaluation Metrics of ANN Over Dataset 2

Evaluation Metrics	**Results**
F1 Score	0.953
Precision	0.960
Recall	0.951
Accuracy	0.955
Total Elapsed Time	8.921seconds

NTD Model- Confusion Matrix

Below, table 5-15 shows the results of the confusion matrix of the proposed model over dataset2:

Table 5.16 Confusion Matrix of NTD over Dataset 2

Predicted Values	**Actual Valued**	
	3393	54
	104	2148

Table of Results:

Table 5-16 below shows the evaluation metric results for the proposed NTD model over dataset 2, indicating very high F1, Precision, Recall, and accuracy values.

Table 5.17 Results of NTD over Dataset2

Evaluation Metrics	**Results**
F1 Score	0.975
Precision	0.983
Recall	0.970
Accuracy	0.972

Comparison Tabel:

Below, Table 5-17 shows the comparison table of the evaluation metrics over data set 2:

Table 5.18 Comparison Table of the Evaluation Metrics over Dataset 2

Algorithm	**F1**	**Precision**	**Recall**	**Accuracy**	**Elapsed Time**
Proposed Model (NTD)	0.975	0.983	0.970	0.972	6.155 seconds
SVM	0.948	0.960	0.939	0.944	15.964 seconds
Decision Tree	0.823	0.893	0.763	0.803	0.152 seconds
ANN	0.953	0.960	0.951	0.955	8.922 seconds

5.6 Summary

The results of the evaluation metrics demonstrated that the proposed solution, NTD, is a highly effective and efficient model for addressing the problem, achieving very high accuracy with minimal elapsed time. In contrast, the SVM algorithm recorded the lowest evaluation scores. In the next chapter, "Future Work," we will outline action items aimed at enhancing the effectiveness and efficiency of NTD. These include improving the generalization of the dataset, optimizing real-time implementation, enhancing scalability for large-scale networks, integrating threat intelligence, improving user experience, and adopting a multi-level security approach.

5.7 Future Work

The NTD model has demonstrated strong performance, and there are numerous opportunities to expand its capabilities, enhance its resilience, and widen its range of applications. Future research should focus on the following key areas:

Improving Generalization of the Dataset

The NTD model has shown promising results, but it must be tested on various datasets that include additional traffic patterns and types of threats to support widespread adoption. To assess how effectively the model generalizes to new scenarios—such as different types of cyberattacks, network topologies, and traffic patterns not included in the initial training data—it is essential to evaluate it in diverse network environments.

Optimizing Real-Time Implementation

The model's response time is crucial for efficient real-time threat detection. While the NTD model performs well, optimizing its architecture for quicker detection will enhance its effectiveness.

Enhancing Large-Scale Network Scalability

Future studies could focus on improving the scalability of the NTD model to enable it to process large datasets in real time. This could involve managing the significant amounts of data generated by large, complex networks through edge computing, cloud-based solutions, or distributed processing techniques.

Integrating Threat Intelligence

Currently, the NTD model does not utilize threat intelligence or external data sources. Enhancing detection capabilities could be achieved by incorporating real-time threat intelligence sources, such as open-source threat databases and security feeds.

Improving User Experience

Implementing adaptive learning capabilities could enhance the user experience and reduce false positives. The model could adjust its detection strategies based on changes in the network environment and user behavior.

Multi-Level Security Approach

Future research could focus on integrating the NTD model with other security technologies, such as firewalls, intrusion prevention systems (IPS), Endpoint Detection and Response (EDR), and SIEM solutions. Although the model currently emphasizes behavioral analysis and the detection of normal traffic from malicious IP addresses, a multi-layered security architecture would provide a more comprehensive defense against a broader range of potential attack vectors.

5.8 Conclusion

This thesis set out to answer four critical questions regarding the efficiency of the proposed model in prediction, prevention, and performance within the context of a specific dataset. The results from our extensive analysis provide positive answers to these questions, clarifying the robustness and potential of the proposed solution. The results showed that, when applied to the dataset, the suggested model successfully achieved accurate prediction and permitting of normal traffic regardless of IP address reputation, utilizing sophisticated analytical approaches and machine learning algorithms.

When benchmarked against other protective system models, the suggested model performed better than others. Through testing and cross-validation, our model surpassed other solutions in important performance parameters, such as F1, precision, recall, and overall accuracy. According to the performance evaluation, NTD highlights reaction speed. The model's speed-optimized architecture guarantees fast detection and reaction to threats. Because of its quick response time performance, the model is well-suited for real-time applications by protecting system integrity.

NTD's ability to allow normal activities and patterns regardless of IP address reputation is strong. Rather than depending exclusively on IP reputation, the model ensures actions are not unnecessarily impeded by concentrating on behavioral patterns and anomaly detection. This strategy improves user experience and operational effectiveness, making the solution a sensible option for networks with various dynamic situations. Effective Prediction and Prevention: Within the dataset, the suggested model (NTD) showed a high degree of accuracy in expecting,
permitting, and preventing security risks.

The model proved its robustness and dependability by successfully identifying an d reducing potential risks. In comparative tests, the proposed model NTD accomplished higher accuracy rates than other systems built on one classifier. This was apparent through prevalent performance metrics: F1 score, precision, recall, and overall accuracy, underscoring the model's improved detection capabilities.

The evaluation of response times showed that the proposed solution works with high efficiency but not the highest. The research designed NTD to guarantee quick detection and response to threats, making it well-suited for environments where timely mediation is critical.

References

Acito, F. (2023). Ensemble Models. In *Predictive Analytics with KNIME* (pp. 255–265). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-45630-5_12

Ahmad, I., Ul Haq, Q. E., Imran, M., Alassafi, M. O., & AlGhamdi, R. A. (2022). An Efficient Network Intrusion Detection and Classification System. Mathematics, 10(3), 530. https://doi.org/10.3390/math10030530

Al Jallad, K., Aljnidi, M., & Desouki, M. S. (2020). Anomaly detection optimization using big data and deep learning to reduce false-positive. Journal of Big Data, 7(1), 68. https://doi.org/10.1186/s40537-020-00346-1

Alenezi, M. N., Alabdulrazzaq, H., Alshaher, A. A., & Alkharang, M. M. (2020). Evolution of malware threats and techniques: A review. International Journal of Communication Networks and Information Security, 12(3), 326–337.

Alharbi, R., Alhichri, H., Ouni, R., Bazi, Y., & Alsabaan, M. (2023). Improving remote sensing scene classification using quality-based data augmentation. International Journal of Remote Sensing, 44(6), 1749–1765.

Ali, A. (2017). Ransomware: A research and a personal case study of dealing with this nasty malware. Issues in Informing Science and Information Technology, 14, 87–99.

AlienVault. (2019). Open Threat Exchange (OTX). https://otx.alienvault.com/

Almaiah, M. A., Almomani, O., Alsaaidah, A., Al-Otaibi, S., Bani-Hani, N., Hwaitat, A. K. Al, Al-Zahrani, A., Lutfi, A., Awad, A. B., & Aldhyani, T. H. H. (2022). Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. Electronics, 11(21), 3571.

Alshehri, A., Khan, N., Alowayr, A., & Yahya Alghamdi, M. (2023). Cyberattack Detection Framework Using Machine Learning and User Behavior Analytics. Computer Systems Science and Engineering, 44(2), 1679–1689. https://doi.org/10.32604/csse.2023.026526

Amin, M. S., & Rahman, S. (2023). An Introduction of Open System Interconnection (OSI) Model and its Architecture. Preprints. https://doi.org/10.20944/preprints202305.1858.v1

Anderson, J. P. (1980). Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company.

Anggraeni, A., Ginting, J. G. A., & Ikhwan, S. (2022). Implementation of intrusion prevention system (IPS) to analysis triad cia on network security attacks on web server. JURNAL INFOTEL, 14(4), 277–286. https://doi.org/10.20895/infotel.v14i4.813

Azam, H., Dulloo, M. I., Majeed, M. H., Wan, J. P. H., Xin, L. T., & Sindiramutty, S. R. (2023). Cybercrime Unmasked: Investigating Cases and Digital Evidence. International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence, 2(1). https://doi.org/10.54938/ijemdcsai.2023.02.1.255

Azam, Z., Islam, Md. M., & Huda, M. N. (2023). Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. IEEE Access, 11, 80348–80391. https://doi.org/10.1109/ACCESS.2023.3296444

Bartoš, V. (2020). Network Entity Reputation Database (NERD): Database of malicious entities on the Internet and everything we know about them. https://nerd.cesnet.cz

Bebeshko, B., Khorolska, K., Kotenko, N., Kharchenko, O., & Zhyrova, T. (2021). Use of Neural Networks for Predicting Cyberattacks. CPITS I, 213–223.

Bengag, A., Bengag, A., & Moussaoui, O. (2021). Classification of security attacks in WBAN for medical healthcare. Proceedings of the 4th International Conference on Networking, Information Systems & Security, 1–5. https://doi.org/10.1145/3454127.3456605

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is Machine Learning? A Primer for the Epidemiologist. American Journal of Epidemiology, 2222–2239. https://doi.org/10.1093/aje/kwz189

Biju, J. M. , G. N. , & P. A. J. (2019). Cyber attacks and its different types. International Research Journal of Engineering and Technology, 6, 4849-4852.

Bilot, T., Madhoun, N. El, Agha, K. Al, & Zouaoui, A. (2023). Graph Neural Networks for Intrusion Detection: A Survey. IEEE Access, 11, 49114–49139. https://doi.org/10.1109/ACCESS.2023.3275789

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (1st ed., Vol. 4, Issue 4). Springer.

Bishop, M. (2019). Healthcare Social Media for Consumer Informatics. In C. and H. E. Edmunds Margo and Hass (Ed.), Consumer Informatics and Digital Health: Solutions for Health and Health Care (pp. 61–86). Springer International Publishing. https://doi.org/10.1007/978-3-319-96906-0_4

Bocu, R., & Iavich, M. (2022). Real-Time Intrusion Detection and Prevention System for 5G and beyond Software-Defined Networks. Symmetry, 15(1), 110. https://doi.org/10.3390/sym15010110

Bohutska, J. (2021). Anomaly Detection—How to Tell Good Performance from Bad. Towards Data Science Inc, 17.

Butt, U. A., Amin, R., Aldabbas, H., Mohan, S., Alouffi, B., & Ahmadian, A. (2023). Cloud-based email phishing attack using machine and deep learning algorithm. Complex & Intelligent Systems, 9(3), 3043–3070. https://doi.org/10.1007/s40747-022-00760-3

Cains, M. G., Flora, L., Taber, D., King, Z., & Henshel, D. S. (2022). Defining Cyber Security and Cyber Security Risk within a Multidisciplinary Context using Expert Elicitation. Risk Analysis, 42(8), 1643–1669. https://doi.org/10.1111/risa.13687

Canto, A. C., Kaur, J., Kermani, M. M., & Azarderakhsh, R. (2023). Algorithmic security is insufficient: A comprehensive survey on implementation attacks haunting post-quantum security. ArXiv Preprint ArXiv:2305.13544.

CESNET. (2017, November 7). CESNET: The Czech Educational and Scientific Network . https://warden.cesnet.cz/en/index

Chałubińska-Jentkiewicz, K. (2022). Cyberspace as an Area of Legal Regulation. Cybersecurity in Poland, 23.

Chen, L., Kuang, X., Xu, A., Suo, S., & Yang, Y. (2020). A Novel Network Intrusion Detection System Based on CNN. 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), 243–247. https://doi.org/10.1109/CBD51900.2020.00051

Chikodili, N. B., Abdulmalik, M. D., Abisoye, O. A., & Bashir, S. A. (2020). Outlier detection in multivariate time series data using a fusion of K-medoid, standardized euclidean distance and Z-score. International Conference on Information and Communication Technology and Applications, 259–271.

Cisco. (n.d.). Common cyberattacks. Retrieved June 27, 2024, from https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html

Constantinides, C., Shiaeles, S., Ghita, B., & Kolokotronis, N. (2019). A Novel Online Incremental Learning Intrusion Prevention System. 2019 10th IFIP International

Conference on New Technologies, Mobility and Security (NTMS), 1–6. https://doi.org/10.1109/NTMS.2019.8763842
Coursera Staff. (2024, October 3). 10 Common Types of Cyberattacks and How to Prevent Them. Online Article.
Cvitić, I., Peraković, D., Periša, M., & Gupta, B. (2021). Ensemble machine learning approach for classification of IoT devices in smart home. International Journal of Machine Learning and Cybernetics, 12(11), 3179–3202. https://doi.org/10.1007/s13042-020-01241-0
Dawadi, B., Adhikari, B., & Srivastava, D. (2023). Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. Sensors, 23(4), 2073. https://doi.org/10.3390/s23042073
Dehlaghi-Ghadim, A., Moghadam, M. H., Balador, A., & Hansson, H. (2023). Anomaly Detection Dataset for Industrial Control Systems. IEEE Access, 11, 107982–107996. https://doi.org/10.1109/ACCESS.2023.3320928
Denning, D. E. (1987). An intrusion-detection model. IEEE Transactions on Software Engineering, 2, 222–232.
Diaba, S. Y., & Elmusrati, M. (2023). Proposed algorithm for smart grid DDoS detection based on deep learning. Neural Networks, 159, 175–184. https://doi.org/10.1016/j.neunet.2022.12.011
Dictionary.com. (2021). dictionary.com. Https://Www.Dictionary.Com/Browse/Network. https://www.dictionary.com/browse/network
Doganer, A. (2021). Different Approaches to Reducing Bias in Classification of Medical Data by Ensemble Learning Methods. International Journal of Big Data and Analytics in Healthcare, 6(2), 15–30. https://doi.org/10.4018/IJBDAH.20210701.oa2
Elmrabit, N., Zhou, F., Li, F., & Zhou, H. (2020). Evaluation of Machine Learning Algorithms for Anomaly Detection. 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), 1–8. https://doi.org/10.1109/CyberSecurity49315.2020.9138871
Elshafie, H. M., Mahmoud, T. M., & Ali, A. A. (2019). Improving the Performance of the Snort Intrusion Detection Using Clonal Selection. 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 104–110. https://doi.org/10.1109/ITCE.2019.8646601
Faker, O., & Dogdu, E. (2019). Intrusion Detection Using Big Data and Deep Learning Techniques. Proceedings of the 2019 ACM Southeast Conference, 86–93. https://doi.org/10.1145/3299815.3314439
Fazal, R., Shah, M. A., Khattak, H. A., Rauf, H. T., & Al-Turjman, F. (2022). Achieving data privacy for decision support systems in times of massive data sharing. Cluster Computing, 25(5), 3037–3049. https://doi.org/10.1007/s10586-021-03514-x
Fraley, J. B., & Cannady, J. (2017). The promise of machine learning in cybersecurity. SoutheastCon 2017, 1–6. https://doi.org/10.1109/SECON.2017.7925283
Gibson, W. (1982). Burning Chrome. Omni.
Guo, Y. (2023). A review of Machine Learning-based zero-day attack detection: Challenges and future directions. Computer Communications, 198, 175–185. https://doi.org/https://doi.org/10.1016/j.comcom.2022.11.001
Hadi, T. H. (2022). Types of Attacks in Wireless Communication Networks. Webology, 19(1), 718–728. https://doi.org/10.14704/WEB/V19I1/WEB19051

Hasan, M. F., & Al-Ramadan, N. S. (2021). Cyber-attacks and cyber security readiness: Iraqi private banks case. Social Science and Humanities Journal (SSHJ), 5(8), 2312–2323.
ISACA. (2008). ISACA glossary. https://www.isaca.org/resources/glossary#glossc
Ismail, A., & Wediawati, B. (2023). Understanding the Fundamentals of Machine Learning and AI for Digital Business (1st ed.). Asadel Publisher.
Khaleefa, E., & Abdulah, D. (2022). Concept and difficulties of advanced persistent threats (APT): Survey. International Journal of Nonlinear Analysis and Applications, 13(1), 4037–4052. https://doi.org/10.22075/ijnaa.2022.6230
Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). TSDL: A Two-Stage Deep Learning Model for Efficient Network Intrusion Detection. IEEE Access, 7, 30373–30385. https://doi.org/10.1109/ACCESS.2019.2899721
Kneusel, R. T. (2021). Math for deep learning: what you need to know to understand neural networks. No Starch Press.
Krishnaveni S. and Vigneshwar, P. and K. S. and J. B. and S. S. (2020). Anomaly-Based Intrusion Detection System Using Support Vector Machine. In C. and D. S. and P. B. K. Dash Subhransu Sekhar and Lakshmi (Ed.), Artificial Intelligence and Evolutionary Computations in Engineering Systems (pp. 723–731). Springer Singapore.
Kumar, S. , Setia, R. , & Sigh, K. (2023). Artificial Intelligence and Machine Learning in Satellite Data Processing and Services (S. Kumar, R. Setia, & K. Singh, Eds.; Vol. 970). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-7698-8
Kumar Singh Gautam, R., & Doegar, Er. A. (2018). An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms. 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 14–15. https://doi.org/10.1109/CONFLUENCE.2018.8442693
Kurama, & Vihar. (2020). The meta-learning method. Https://Blog.Paperspace.Com/Adaboost-Optimizer/.
Lamrini, M., Chkouri, M. Y., & Touhafi, A. (2023). Evaluating the Performance of Pre-Trained Convolutional Neural Network for Audio Classification on Embedded Systems for Anomaly Detection in Smart Cities. Sensors, 23(13), 6227. https://doi.org/10.3390/s23136227
Lansley, M., Mouton, F., Kapetanakis, S., & Polatidis, N. (2020). SEADer++: social engineering attack detection in online environments using machine learning. Journal of Information and Telecommunication, 4(3), 346–362. https://doi.org/10.1080/24751839.2020.1747001
Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176–8186. https://doi.org/10.1016/j.egyr.2021.08.126
Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (2022). An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. Sensors, 22(4), 1407. https://doi.org/10.3390/s22041407
Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences, 9(20), 4396. https://doi.org/10.3390/app9204396
Liu, Z., Qian, L., & Tang, S. (2022). The prediction of DDoS attack by machine learning. In M. K. Mohiddin, S. Chen, & S. F. EL-Zoghdy (Eds.), Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021) (p. 55). SPIE. https://doi.org/10.1117/12.2628658

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings, 3(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

Mahmud, A.-, & Shimada, K. (2023). Dataset Construction and Opinion Holder Detection Using Pre-trained Models. International Journal of Service and Knowledge Management, 7(2), 779. https://doi.org/10.52731/ijskm.v7.i2.779

Mahmud, M., Ieracitano, C., Kaiser, M. S., Mammone, N., & Morabito, F. C. (2023). Applied Intelligence and Informatics: Second International Conference, AII 2022, Reggio Calabria, Italy, September 1–3, 2022, Proceedings. Springer Nature.

Malizan, N. A., Razali, N. A. M., Hasbullah, N. A., Wook, M., Zainudin, N. M., & Ramli, S. (2022). Opinion mining hybrid technique to classify people's emotions in text using Kansei and lexicon-based approach for national security domain. AIP Conference Proceedings, 2617(1).

Malwarebytes. (2020). Spam. https://www.malwarebytes.com/spam

Markevych, M., & Dawson, M. (2023). A Review of Enhancing Intrusion Detection Systems for Cybersecurity Using Artificial Intelligence (AI). International Conference KNOWLEDGE-BASED ORGANIZATION, 29(3), 30–37. https://doi.org/10.2478/kbo-2023-0072

Matzelle, E. (2019). Types of cyber attacks. https://www.comptia.org/blog/types-of-cyber-attacks

Meddeb, R., Jemili, F., Triki, B., & Korbaa, O. (2023). A deep learning-based intrusion detection approach for mobile Ad-hoc network. Soft Computing, 27(14), 9425–9439. https://doi.org/10.1007/s00500-023-08324-4

Meira, J., Andrade, R., Praça, I., Carneiro, J., Bolón-Canedo, V., Alonso-Betanzos, A., & Marreiros, G. (2020). Performance evaluation of unsupervised techniques in cyber-attack anomaly detection. Journal of Ambient Intelligence and Humanized Computing, 11(11), 4477–4489. https://doi.org/10.1007/s12652-019-01417-9

Miao, Y. (2021). Machine Learning Based Cyber Attack Targeting on Controlled Information. Swinburne.

Michie, D., Spiegelhalter, D. J., Taylor, C. C., & Campbell, J. (1995). Machine learning, neural and statistical classification. Ellis Horwood.

Mohapatra, H. (2020). Handling of Man-In-The-Middle Attack in WSN Through Intrusion Detection System. International Journal of Emerging Trends in Engineering Research, 8(5), 1503–1510. https://doi.org/10.30534/ijeter/2020/05852020

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

Munir, M., Chattha, M. A., Dengel, A., & Ahmed, S. (2019). A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 561–566. https://doi.org/10.1109/ICMLA.2019.00105

Muntean, M., & Militaru, F.-D. (2023). Metrics for evaluating classification algorithms. Education, Research and Business Technologies: Proceedings of 21st International Conference on Informatics in Economy (IE 2022), 307–317.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics, 18(6), 275–285. https://doi.org/10.1002/cem.873

National Initiative for Cybersecurity Careers, & (NICCS), S. (2020). NICCS Glossary. https://niccs.cisa.gov/cybersecurity-career-resources/glossary#C

National Institute of Standards, & Technology. (2016). NIST cybersecurity framework: Introduction. https://doi.org/10.6028/NIST.SP.800-37r2

Nguyen, V.-A., Nguyen, D. Q., Nguyen, V., Le, T., Tran, Q. H., & Phung, D. (2022). ReGVD: Revisiting graph neural networks for vulnerability detection. Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, 178–182.

Note, J., & Ali, M. (2022). Comparative Analysis of Intrusion Detection System Using Machine Learning and Deep Learning Algorithms. Annals of Emerging Technologies in Computing, 6(3), 19–36. https://doi.org/10.33166/AETiC.2022.03.003

Olateju, O. O., Okon, S. U., Igwenagu, U. T. I., Salami, A. A., Oladoyinbo, T. O., & Olaniyi, O. O. (2024). Combating the Challenges of False Positives in AI-Driven Anomaly Detection Systems and Enhancing Data Security in the Cloud. Asian Journal of Research in Computer Science, 17(6), 264–292. https://doi.org/10.9734/ajrcos/2024/v17i6472

Palanivinayagam, A., & Damaševičius, R. (2023). Effective Handling of Missing Values in Datasets for Classification Using Machine Learning Methods. Information, 14(2), 92. https://doi.org/10.3390/info14020092

Pan, J.-S., Fan, F., Chu, S.-C., Zhao, H.-Q., & Liu, G.-Y. (2021). A Lightweight Intelligent Intrusion Detection Model for Wireless Sensor Networks. Security and Communication Networks, 2021, 1–15. https://doi.org/10.1155/2021/5540895

Patel, H. (2021). What is Feature Engineering? Importance, Tools, and Techniques for Machine Learning. https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10

Project, M. (2011). Malware Information Sharing Platform & Threat Sharing. https://www.misp-project.org/

Qaddoura, R., M. Al-Zoubi, A., Faris, H., & Almomani, I. (2021). A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. Sensors, 21(9), 2987. https://doi.org/10.3390/s21092987

Qazi, E. U. H., Faheem, M. H., & Zia, T. (2023). HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. Applied Sciences, 13(8), 4921. https://doi.org/10.3390/app13084921

R, S., Mary M, M. J., P, S. D., & S, S. M. (2023). IoT-Based Leaf Disease Detection and Alerting System Using K-means Algorithm. 2023 International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (ITech SECOM), 201–206. https://doi.org/10.1109/iTechSECOM59882.2023.10435090

R, V., & A, C. (2019). COMPREHENSIVE SURVEY OF WIRELESS COGNITIVE AND 5G NETWORKS. Journal of Ubiquitous Computing and Communication Technologies, 01(01), 23–32. https://doi.org/10.36548/jucct.2019.1.003

Radoglou-Grammatikis, P., Sarigiannidis, P., Efstathopoulos, G., Karypidis, P.-A., & Sarigiannidis, A. (2020). DIDEROT: an intrusion detection and prevention system for DNP3-based SCADA systems. Proceedings of the 15th International Conference on Availability, Reliability and Security, 1–8. https://doi.org/10.1145/3407023.3409314

Raina, V., & Krishnamurthy, S. (2022). Data Preparation in Building an Effective Data Science Practice. Apress Berkeley, CA. https://doi.org/10.1007/978-1-4842-7419-4

Revanesh, M., Gundal, S. S., Arunkumar, J. R., Josephson, P. J., Suhasini, S., & Devi, T. K. (2024). Artificial neural networks-based improved Levenberg–Marquardt neural network for energy efficiency and anomaly detection in WSN. Wireless Networks, 30(6), 5613–5628.

Rosenthal, S. (2017). Data Imputation. In The International Encyclopedia of Communication Research Methods (pp. 1–12). Wiley. https://doi.org/10.1002/9781118901731.iecrm0058

Salman, H. A., Alsajri, A., Kalakech, A., & Steiti, A. (2023). Difference Between 4G and 5G Networks. Babylonian Journal of Networking, 2023, 41–54. https://doi.org/10.58496/BJN/2023/006

SANS Internet Storm Center. (2023). SANS Internet Storm Center. https://www.dshield.org/

Sen, J., & Mehtab, S. (2020). Machine learning applications in misuse and anomaly detection. Security and Privacy from a Legal, Ethical, and Technical Perspective, 155.

Shah, B., & H Trivedi, B. (2012). Artificial Neural Network based Intrusion Detection System: A Survey. International Journal of Computer Applications, 39(6), 13–18. https://doi.org/10.5120/4823-7074

Sihag Vikas and Swami, A. and V. M. and S. P. (2020). Signature Based Malicious Behavior Detection in Android. In S. and A. K. Chaubey Nirbhay and Parikh (Ed.), Computing Science, Communication and Security (pp. 251–262). Springer Singapore.

Singh, G., & Khare, N. (2022). A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. International Journal of Computers and Applications, 44(7), 659–669. https://doi.org/10.1080/1206212X.2021.1885150

Sowmya, T., & Mary Anita, E. A. (2023). A comprehensive review of AI based intrusion detection system. Measurement: Sensors, 28, 100827. https://doi.org/10.1016/j.measen.2023.100827

Spadaccino, P., & Cuomo, F. (2020). Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Computing and Machine Learning. ArXiv Preprint ArXiv:2012.01174.

Study Group, I. (2008). ITU-T Rec. X.1205 (04/2008) Overview of cybersecurity.

Sucuri. (2022, May 10). IP reputation: What you need to know. Sucuri Blog. https://blog.sucuri.ne

Tableau. (2016). What is data cleaning? https://www.tableau.com/learn/articles/what-is-data-cleaning

Tekerek, A., & Bay, O. F. (2019). DESIGN AND IMPLEMENTATION OF AN ARTIFICIAL INTELLIGENCE-BASED WEB APPLICATION FIREWALL MODEL. Neural Network World, 29(4), 189–206. https://doi.org/10.14311/NNW.2019.29.013

Thapa, S., & Mailewa, A. (2020). The role of intrusion detection/prevention systems in modern computer networks: A review. Conference: Midwest Instruction and Computing Symposium (MICS), 53, 1–14.

Tsagourias, N. (2021). The legal status of cyberspace: sovereignty redux? In Research Handbook on International Law and Cyberspace (pp. 1–23). Edward Elgar Publishing. https://doi.org/10.4337/9781789904253.00010

Usman, N., Usman, S., Khan, F., Jan, M. A., Sajid, A., Alazab, M., & Watters, P. (2021). Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation

for Forensics Data Analytics. Future Generation Computer Systems, 118, 124–141. https://doi.org/10.1016/j.future.2021.01.004

Valkenborg, D., Rousseau, A.-J., Geubbelmans, M., & Burzykowski, T. (2023). Support vector machines. American Journal of Orthodontics and Dentofacial Orthopedics, 164(5), 754–757. https://doi.org/10.1016/j.ajodo.2023.08.003

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust Intelligent Malware Detection Using Deep Learning. IEEE Access, 7, 46717–46738. https://doi.org/10.1109/ACCESS.2019.2906934

Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A. (2016). MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, 49–56. https://www.misp-project.org/

Wei, R., Cai, L., Zhao, L., Yu, A., & Meng, D. (2021). Deephunter: A graph neural network based approach for robust cyber threat hunting. Security and Privacy in Communication Networks: 17th EAI International Conference, SecureComm 2021, Virtual Event, September 6–9, 2021, Proceedings, Part I 17, 3–24.

Wyre, M., Lacey, D., & Allan, K. (2020). The identity theft response system. Trends and *Issues in Crime and Criminal Justice*, *592*, 1–18.

Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. Journal of Big Data, 10(1), 15. https://doi.org/10.1186/s40537-023-00694-8

Yohanandhan, R. V., Elavarasan, R. M., Manoharan, P., & Mihet-Popa, L. (2020). Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis With Cyber Security Applications. IEEE Access, 8, 151019–151064. https://doi.org/10.1109/ACCESS.2020.3016826

Yokkampon, U., Chumkamon, S., Mowshowitz, A., Fujisawa, R., & Hayashi, E. (2021). Anomaly Detection Using Support Vector Machines for Time Series Data. Journal of Robotics Networking and Artificial Life.

Yu, Y., & Bian, N. (2020). An Intrusion Detection Method Using Few-Shot Learning. IEEE Access, 8, 49730–49740. https://doi.org/10.1109/ACCESS.2020.2980136

Zhang, L., Pan, Y., Liu, Y., Zheng, Q., & Pan, Z. (2022). Multiple domain cyberspace attack and defense game based on reward randomization reinforcement learning. ArXiv Preprint ArXiv:2205.10990.

Zhang, S., Yao, R., Du, C., Essah, E., & Li, B. (2023). Analysis of outlier detection rules based on the ASHRAE global thermal comfort database. Building and Environment, 234, 110155. https://doi.org/10.1016/j.buildenv.2023.110155

Zheng, C. (2021). Computer Network Security and Effective Measures for the Era of Big Data. In F. Jan Mian Ahmad and Khan (Ed.), Application of Big Data, Blockchain, and Internet of Things for Education Informatization (pp. 521–529). Springer International Publishing.

اكتشاف والسماح بحركة المرور المشروعة من عناوين الIP ذات السمعة الضارة.

بشار سعدي مصطفى جابر

اسماء لجنة الاشراف

الدكتور أسامة منصور

الدكتور مجدي عودة

الدكتورة كاترين قويدر

ملخص

تعد مراقبة الشبكة عنصرًا أساسيًا لضمان الأمن السيبراني. تقدم هذه الأطروحة نموذج NTD لاكتشاف حركة المرور العادية، حيث يميز بين العناوين الطبيعية وغير الطبيعية باستخدام مزيج من خوارزميات SVM، والشبكات العصبية الاصطناعية(ANN) ، وأشجار القرار.

تعتمد منهجية NTDعلى تحليل شامل لحركة المرور، حيث تبدأ العملية بمقارنة الأنماط السلوكية مع قاعدة بيانات الشذوذ .(ABD) يتم تصنيف أي تطابق مع ABDعلى الفور كحركة ضارة، مما يستدعي تحقيقًا إضافيًا. في المقابل، يتم تمرير البيانات غير المطابقة عبر عملية تصنيف متتابعة باستخدام SVMو ANNوأشجار القرار، حيث تسهم كل خوارزمية في تحليل البيانات من زوايا مختلفة للكشف عن الحالات الشاذة غير المعروفة. يتم إثراء ABDبشكل مستمر من خلال تسجيل التهديدات المكتشفة حديثًا، مما يعزز دقة النظام وقدرته على التكيف مع التهديدات السيبرانية المتطورة.ولإثبات فعالية NTD، تم إجراء اختبارات تجريبية باستخدام مجموعات بيانات حقيقية في مجال الأمن السيبراني وقد أظهرت النتائج تفوق NTD على الأساليب التقليدية أحادية الخوارزمية، حيث حقق معدلات أداء عالية وفقًا لمقاييس الدقة والاستدعاء وF1−Score. وتمتد تطبيقات NTD إلى مختلف القطاعات حيث يساعد في معالجة التصنيف الخاطئ للبيانات من خلال تقديم نهج تكاملي يعتمد على الذكاء الاصطناعي والتعلم الآلي .تؤكد هذه الأطروحة على دور NTD كحل مبتكر في اكتشاف الشذوذ في حركة المرور، مسلحًا بتقنيات تحليل متقدمة ونتائج تجريبية داعمة، مما يمهد الطريق لنموذج دفاعي أكثر تطورًا ومرونة.

الكلمات المفتاحية :اكتشاف حركة المرور(NTD) ، كشف الشذوذ، SVM، التعلم الآلي، الأمن السيبراني.