Arab American University

Faculty of Graduate Studies

Detection and Prediction of Ransomware Based on Network Behavior Using Machine Learning

By

Hamdi Yasser Ayoub Abu Helo

Supervisor

Dr. Mohammad M N Hamarsheh

This thesis was submitted in partial fulfillment of the requirements for the Master`s degree in Cyber Security

March /2024

© Arab American University -2024. All rights reserved.

Thesis Approval

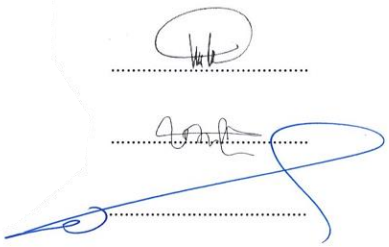
Detection and Prediction of Ransomware Based on Network Behavior Using Machine Learning

By

Hamdi Yasser Ayoub Abu Helo

This thesis was defended successfully on March 9, 2024 and approved by:

Committee members	Signature
1. Dr. Mohammad Hamarsheh: Supervisor	.....
2. Dr. Huthaifa Isam Ashqar: Internal Examiner	.....
3. Dr. Ahmad Awad: External Examiner	.....



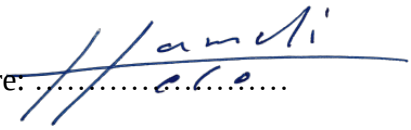
Declaration

I, the undersigned Hamdi Yasser Ayoub Abu Helo, hereby declare that, following my enrollment in the master's program in information security at Arab American University in Palestine, I am the sole author of the thesis titled "Detection and Prediction of Ransomware Based on Network Behavior Using Machine Learning." This thesis is being supervised by Dr. Mohammad Hamarsheh.

To the best of my knowledge, this thesis represents my original work, and I have appropriately credited any sources used. There is no content in this thesis that has been submitted for credit in fulfillment of the requirements of any other academic degree or non-degree program.

The Name of The Student: Hamdi Yasser Ayoub Abu Helo

ID: 202112487

Signature: 

Date: August 20, 2024

Dedication

This thesis is dedicated to the people who have supported me throughout my education.

Thanks for making me see this adventure through to the end.

Acknowledgments

“I am deeply grateful to my supervisor, Dr. Mohammed Hamarsheh, for his unwavering support and guidance throughout my master’s program. Their expertise and patience have been invaluable to me and have played a crucial role in the success of this thesis.

I am also grateful to Arab American University for providing me with the opportunity to conduct my research and for all the resources and support they provided. I want to extend a special thanks to my wife went above and beyond to help me with my work.

Yasser and Salma, my great children and my support in the first place, the safe embrace after all the tiredness, I am grateful for your presence, endurance, and support in this work.

I am deeply thankful to my friends and family for their love and support during this process. Without their encouragement and motivation, I would not have been able to complete this journey.

My father, mother, sisters, and brothers; your unwavering support has been the bedrock of my academic journey. This thesis is a reflection of the strength and inspiration I draw from each of you. Thank you for being the pillars of my success.”

Abstract

Computer malware has been growing at a rapid way in recent years, with ransomware emerging as a particularly powerful threat. Numerous victims, including companies, hospitals, and individuals, have suffered large financial losses as a result of the fast spread of ransomware. Due to the fact that they frequently rely on infection detection, traditional ransomware detection techniques have often proven ineffective. Using network behavior analysis to preventatively identify ransomware incidents provides a more effective solution to this problem. However, weaknesses identified in the literature review include limited sample sizes, reliance on few protocols, excessive feature usage leading to higher execution times, and experiments conducted in outdated environments. Addressing these gaps, this research offers a robust framework for ransomware detection and analysis, contributing significantly to the cybersecurity domain. This thesis uses a dataset of 145 ransomware samples from four different families to give an extensive analysis of network behavior with a particular focus on ransomware. A dedicated, isolated testbed was built to support this study. A traffic recording and information extraction from compromised host device from the testbed are part of the study. The investigation employs three machine learning algorithms - Random Forest (RF), k-Nearest Neighbors (KNN), and Gradient Boosting - to analyze the recorded data. This thesis presents a rigorous investigation into four ransomware families, achieving exceptional accuracy in binary and multiclass detection with notable execution time efficiency. The study also explores a diverse range of communication protocols used by ransomware such as HTTP, and TCP with focus on the use of the flags SYN, ACK, and RST.

Table of Contents

Thesis Approval i

Declaration ii

Dedication iii

Acknowledgments iv

Abstract v

Table of Contents vi

List of Tables ix

List of Figures x

List of Appendices xii

List of Abbreviations xiii

1 Chapter One Introduction 1

1.1 Introduction 2

1.2 Motivation 4

1.3 Problem Statement 6

1.4 Objective of the Study 8

1.5 Conclusion 10

2 Chapter Two Theoretical Background 11

2.1 Introduction 12

2.2 Cyber Security 13

2.3 Intrusion Detection System IDS 16

2.4 Ransomwares 18

2.4.1 Ransomware Attacks Classification 20

2.4.2 Ransomware Stages 22

2.4.3 Ransomware as a Service 24

2.4.4 Common Ransomware Attacks 25

2.5 Machine Learning 29

2.5.1 Machine Learning Algorithms 31

2.5.2 Classification 32

2.5.3 Dataset Preprocessing 34

3 Chapter Three Literature Review 36

3.1 Literature Review 37

4 Chapter Four Methodology 48

4.1 Introduction 49

4.2 Proposed Model 50

4.3 Testbed 52

4.4 Collecting Traffic 54

4.5 Traffic Analysis 56

4.5.1 HTTP Traffic 57

4.5.2 DNS Traffic 61

4.5.3 TCP flags 64

4.6 Dataset 68

4.7 Preprocessing 73

4.8 Performance Measurement 86

5 Chapter Five Results 89

5.1 Introduction 90

5.2 Overview 91

5.3 Labeling Results for Generated Dataset 92

5.4 Algorithmic Performance Analysis 93

5.5 Thesis Contributions: A Comparative Analysis with Related Approaches 103

5.6 Conclusion 105

5.7 Future Implications and Enhancements 107

References 109

Appendices 115

الملخص 127

List of Tables

Table 2.1 Used Ransomwares 29

Table 3.1 comprehensive analysis for ransomware detection 44

Table 4.1 PCAP files statistics 55

Table 4.2 Number of ransomware samples 56

Table 4.3 Protocols percentages 57

Table 4.4 Connected domains 62

Table 4.5 Dataset Features 68

Table 4.6 Experiment times 71

Table 4.7 Labeling Count 72

Table 5.1 Algorithms' results 96

List of Figures

Figure 2.1 Anomaly Based Intrusion Detection (Panagiotou et al., 2021) 17

Figure 2.2 Host Based Intrusion Detection (Panagiotou et al., 2021) 18

Figure 2.3 Cerber ransomware message after affect our system 20

Figure 2.4 Ransomware Stages 22

Figure 2.5 Decision surface (Why Is Logistic Regression a Classification Algorithm? | Built In, 2023) 33

Figure 4.1 Proposed model 50

Figure 4.2 Testbed 53

Figure 4.3 CryptoWall TCP connection 58

Figure 4.4 Virustotal result for bihartbass.org 59

Figure 4.5 Benign and ransomware HTTP traffic 60

Figure 4.6 DarkSide Connected domains 64

Figure 4.7 Benign and ransomware RST traffic 66

Figure 4.8 Benign and ransomware ACK traffic 67

Figure 4.9 TeslaCrypt traffic 68

Figure 4.10 Checking missing values 74

Figure 4.11 Use mean of Flow Byts column 75

Figure 4.12 Dropped features 76

Figure 4.13 replacing infinite values with mean value 77

Figure 4.14 Correlation Heatmap 78

Figure 4.15 highly correlated features 79

Figure 4.16 The score of each feature 85

Figure 5.1 Label distribution 92

Figure 5.2 Accuracy and Execution Time 96

Figure 5.3 Random Forest Confusion matrix 98

Figure 5.4 Gradient Boosting confusion matrix 100

Figure 5.5 KNN confusion matrix 102

Figure 6.1 HTTP request for Cerber Ransomware 115

Figure 6.2 DNS traffic for Cerber Ransomware 115

Figure 6.3 Checking one of the connected IP related to Cerber traffic 116

Figure 6.4 Checking one of the connected domains for CryptoWall 116

Figure 6.5 One of the connected Ip's used by CryptoWall 116

Figure 6.6 HTTP Post to malicious domain used by CryptoWall 117

Figure 6.7 TeslaCrypt DNS traffic 117

Figure 6.8 HTTP request for DarkSide ransomware 117

Figure 6.9 Malicious domain used by DarkSide ransomware 118

List of Appendices

Appendix (1) Dependencies for The Project 115

Appendix (2) Code 119

Appendix (3) Features Explanation 124

List of Abbreviations

ANN	Artificial Neural Network
C&C	Command And Control
CIA	Confidentiality, Integrity, and Availability
DGA	Domain Generation Algorithm
DES	Data Encryption Standard
DNS	Domain Name Service
HSR	High Survival Ransomware
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
MIC	Maximum Information Coefficient
PC	Personal Computer
RaaS	Ransomware As a Service
RSA	Rivest, Shamir, Adleman
SDN	Software Defined Network
TCP	Transmission Control Protocol

1 Chapter One
Introduction

1.1 Introduction

Nowadays, there is a tremendous increase in the use of the Internet, there were 5.3 billion internet users globally as of October 2023, almost 4.95 billion of them used social media of all kinds (Internet and social media users in the world 2023 | Statista, 2023). These numbers indicate the large number of devices and servers connected via the Internet, which in turn increases the probability of various attacks occurring due to weak security and protection on various devices despite using the default antivirus or even the Intrusion Detection Systems (IDS).

Malware attacks have become Increasingly widespread in targeting computer systems (Bae, Lee and Im, 2020). These malicious activities primarily occur through online channels, such as email phishing, rogue websites, and drive-by software downloads. Different types of malwares, including viruses, trojan horses, worms, and ransomware, pose significant threats to individuals and organizations (Singh and Singh, 2021). Among these, ransomware has gained special importance, exploiting a business model known as Ransomware as a Service (RaaS) that allows even non-technical individuals to carry out attacks and demand ransoms (Cabaj et al., 2015).

Ransomware is not a new concept, The 1989 AIDS Trojan, also known as PS Cyborg, was the first known case of ransomware (Kara and Aydos, 2022). 20,000 harmed floppy disks under the title "AIDS Information - Introductory Diskettes" were sent by scientist Joseph L. Popp to participants in the worldwide AIDS conference organized by the World Health Organization (A Brief History of Ransomware, 2023). On the client's PC, the Trojan

concealed folders and encrypted file names. The user would need to deposit $189 to PC Cyborg Corp in order to get access back.

Ransomware operates through various mechanisms, typically involving the locking and encryption of victims' devices, followed by ransom demands (Arabo et al., 2020). These attacks are facilitated by Command and Control (C&C) servers, which allow attackers to control the spread and impact of the malware. Annoyingly, statistics indicate that ransomware attacks affected over 51% of enterprises in 2019 (Berrueta et al., 2022b). Notably, even large-scale organizations have fallen victim to ransomware, such as the Colonial Pipeline Company, which paid approximately $4.4 million to recover encrypted data after a ransomware incident in May 2021 (Alqahtani and Sheldon, 2022). Perhaps the most famous ransomware attack that occurred locally recently was when the Palestinian Ministry of Health was subjected to an attack that led to the encryption of all data stored on the ministry's servers in August 2022 (*فك التشفير لخوادم وزارة الصحة بعد هجوم 'سيبراني' خارجي*, 2023). The report did not mention the type of family that attacked the servers. The report merely stated that the attack was external and it was sufficient that the encryption had been decrypted and the data returned until 8/25/2022. The interconnected nature of the Internet and cloud computing has amplified the propagation of ransomware through various communication protocols, enabling adversaries to employ diverse attack tactics.

Ransomware can be categorized into two main types based on its behavior. The first and most common type is crypto-ransomware, which encrypts victims' data and files, making them inaccessible without the decryption key (Richardson and North, 2017). The second type is locker ransomware, which locks victims out of their computers, preventing normal usage

(Richardson and North, 2017). Within the crypto-ransomware category, there are different varieties based on the cryptosystems employed (Kurniawan and Riadi, 2018). Symmetrical cryptosystem ransomware employs a single key for both encryption and decryption, using algorithms like Data Encryption Standard (DES). This presents victims with the possibility of recovering their files through techniques such as memory scanning or reverse engineering to uncover the secret key. on the other hand, asymmetric cryptosystem ransomware, encrypts files using a public key which only the attacker owns it, making it nearly impossible for victims to retrieve their files without paying the ransom. This type often uses more computational resources due to the encryption method, employing algorithms like RSA.

1.2 Motivation

The security of digital networks has become a critical problem in an era where digital networks are used more and more. New dangers and vulnerabilities surface as a result of the ongoing evolution of the digital environment. Since no network can ever be guaranteed to be completely secure, it is more important than ever to take preventative steps to defend against malicious access and data breaches. By building a special testbed, using open-source tools to capture the traffic and developing models by applying machine learning approaches to network flows for their ability to discern patterns and anomalies in complex data, enabling proactive detection and prediction of network security threats. This method harnesses the power of algorithms to analyze large-scale network traffic data and uncover subtle indicators of malicious activities, enhancing overall cybersecurity measures.

This thesis seeks to solve these urgent problems:

- No network is immune to security breaches due to the inherent instability of digital networks. Networks, which act as a medium for information exchange, communication, and information sharing, are the foundation of our everyday lives in today's linked world. But the nature of these networks - their interconnectivity - also makes them inherently weak. Threats to network security, including malware, phishing scams, and data breaches, are common and are always becoming more sophisticated.
- Malware known as “ransomware,” which encrypts data and demands payment to unlock it, has become a common and costly risk to individuals and businesses (Aldauiji, Batarfi and Bayousef, 2022). To confront this threat, new strategies are needed. Machine learning techniques are a viable method for detecting ransomware because they have proven efficient in processing large data sets. By using machine learning techniques on network traffic, this thesis seeks to advance the field of ransomware detection. In doing so, we want to create faster and more accurate detection systems that can protect networks from the harmful effects of ransomware attacks or minimize their spread on the network.

In short, the field of network security is characterized by constant transformation and dynamic threats. This thesis aims to support ongoing efforts to strengthen network defenses, while recognizing the intrinsic instability of digital networks. This research aims to strengthen the network security posture and enable individuals and organizations to navigate the digital world with greater flexibility and confidence by putting forward a data filtering mechanism based on open-source tools and investigating the creation of machine learning-

based ransomware models. It is impossible to overstate the importance of this work because it directly addresses the ongoing and urgent demand for creative and practical network security solutions.

The use of machine learning in network ransomware prediction and detection has a number of clear benefits. Machine learning algorithms are particularly good at detecting complex patterns and analyzing massive amounts of data, which makes them ideal for detecting small signs of ransomware activity in network traffic. Furthermore, machine learning models are dynamic defenses against changing ransomware strains because they may adjust and change in response to fresh data and new threats. By taking a proactive approach, ransomware attacks may be identified and mitigated early on, reducing the likelihood of harm and interruption to vital systems. Using machine learning, this study improves cybersecurity measures' efficacy and efficiency, strengthening their ability to withstand ransomware attacks in network environments.

1.3 Problem Statement

The constant growth of cyber threats, especially ransomware, poses a serious threat to network security in today's linked digital environment. Ransomware is a type of malicious software that encrypts data and demands a payment to unlock it. It has become a widespread and expensive threat that harms both individuals and companies significantly both financially and operationally. The topic addressed in this thesis arises in the context of the apparent need for reliable systems to identify and anticipate malicious activity related to ransomware.

The following primary issue is the focus of this study:

Problem: The inadequacy of traditional security solutions has been highlighted by the exponential rise in ransomware attacks and the constant creation of new ransomware variants. Existing security solutions often find it difficult to keep up with the ever-evolving strategies used by ransomware, leaving networks open to potentially catastrophic attacks.

Subproblems:

- Lack of Real-World Ransomware Traffic Data: Access to genuine ransomware traffic data is necessary for the development of a machine learning model for ransomware detection. Although there are a few existing datasets, they might not accurately capture the wide variety of ransomware variations and strategies that are encountered in everyday life. Moreover, the dynamic characteristics of ransomware necessitate the constant updating and enlargement of the dataset.
- Feature Extraction and Model Development: It is a difficult task to transform unprocessed network traffic data into useful features for machine learning models. One difficult part of this research is creating a feature extraction procedure that is reliable and efficiently captures the unique traits of ransomware traffic. Furthermore, sophisticated methods and techniques are needed to create machine learning models that can reliably identify and forecast malicious ransomware traffic based on these features.
- Dynamic and Changing Threat Landscape: Because ransomware attackers are always changing their tactics, it is difficult to stay ahead of them. One of the biggest challenges is keeping up with the ways ransomware publishers are changing their

strategies and improving machine learning models to keep high accuracy even when threats change.

Significance of the Problem:

The problem which this thesis attempts to solve has significant implications for network security. Organizations and people can become more resilient to one of the most common and dangerous cyber threats by being able to detect and predict ransomware traffic. Innovative approaches, such as a large and dynamic dataset, sophisticated feature extraction methods, and flexible machine-learning models, will be needed to accomplish this goal. By doing this, the research hopes to support ongoing initiatives to improve network security, fight ransomware, and protect infrastructure and important data from the damaging effects of ransomware attacks.

1.4 Objective of the Study

The main goal of this research is to comprehensively address the complex problem of ransomware in the context of network security. In order to achieve this general goal, the following specific goals were set:

- Detect Ransomware at the early stage

  The primary goal is to create and apply machine learning models that can analyze network traffic to detect ransomware in its early stages with the minimum time. Preventing or reducing the devastating effects of ransomware attacks requires early detection. Through early detection of ransomware, network administrators and

security experts can promptly quarantine impacted systems, preventing crucial data encryption and mitigating potential damage to the enterprise.

- Understanding Ransomware behavior

  This study's second goal is to learn more about the behavior of ransomware. This entails a thorough examination of ransomware's methods of propagation, target selection, and encryption techniques. The research aims to give a better understanding of the strategies used by malicious actors by clarifying the way ransomware operates. When creating the most effective incident response plans and security measures, these insights are invaluable.

- Developing Effective Countermeasures

  Developing and recommending improved network security techniques to combat the ransomware threat is another goal of this research. In order to reduce vulnerabilities and strengthen the network's defenses against ransomware attacks, these steps include proactive defenses, network configurations, and security policies. Organizations can greatly lessen their exposure to ransomware threats by matching security measures with known ransomware behaviors.

- Incident Response

  Giving organizations efficient incident response plans to utilize in the case of a ransomware attack is the study's ultimate goal. The goal of this research is to reduce the possible harm caused by an attack by developing well-defined and tested incident response plans.

1.5 Conclusion

This chapter go through the fundamental foundations of our study endeavor, revealing the purpose and urgency of our pursuit. We provide an informative overview at the outset and outline the context in which our study takes place, laying the groundwork for a thorough examination of machine learning and cybersecurity. The driving force behind this study is the growing threat that ransomware poses, a digital problem that necessitates creative approaches to detection and prediction. The problem statement articulated herein encapsulates the core challenge we seek to address the imperative to fortify digital infrastructures against ransomware assaults through machine learning prowess. Our objective is clarified as to develop robust models capable of not only detecting ransomware with precision but also predicting its insidious maneuvers, thus empowering organizations to proactively defend their digital assets.

2 Chapter Two
Theoretical Background

2.1 Introduction

Cybersecurity and machine learning intersect at a crucial point in modern digital defense strategies. Chapter 2 explores the detailed theoretical backgrounds that support effective ransomware detection and prediction. Cybersecurity has evolved from traditional prevention-focused methods to advanced detection and response frameworks, forming the foundation of modern defense. Intrusion Detection Systems (IDS) act as guards, analyzing the complex attack types, including the dangerous ransomware variants that threaten digital landscapes. The chapter examines ransomware attacks, from classification to operational stages and the growing trend of Ransomware as a Service (RaaS), providing insights into combating these threats. Additionally, the chapter covers machine learning, explaining the differences between supervised, unsupervised, and semi-supervised learning methods. Machine learning algorithms are key to predicting and classifying ransomware, improving defense strategies.

Chapter 2 provides a theoretical background for understanding and combating ransomware. It begins with a discussion on cyber security, detailing its evolution from basic prevention methods to advanced detection and response strategies. The chapter then explains Intrusion Detection Systems (IDS), highlighting their role in identifying various types of cyber attacks.

Next, the focus shifts to ransomware, covering its classification, the different stages of a ransomware attack, and the concept of Ransomware as a Service (RaaS). This section also includes examples of common ransomware attacks.

The chapter concludes with an exploration of machine learning, explaining various algorithms used in cyber security. It discusses the differences between supervised,

unsupervised, and semi-supervised learning, and emphasizes the importance of classification and dataset preprocessing in developing effective ransomware detection strategies.

2.2 Cyber Security

The wide-ranging area of cybersecurity, also known as information security, is devoted to preventing illegal access, attacks, theft, and damage to computer systems, networks, and data. With the increasing interconnectedness and reliance on digital technologies in our world, cybersecurity has become increasingly important. Computer security defined as the safeguard that an automated information system is given to achieve the relevant goals of maintaining the availability, integrity, and confidentiality of information system resources, which include firmware, hardware, software, and data as well as communications (Stallings, 2010). And the main goal of the cyber security is how to protect these three concepts Confidentiality, Integrity, and Availability (CIA) (Lapiedra, 2000).

Protecting digital systems and data from damage, unwanted access, and attacks is the dynamic process of cybersecurity (Stallings, 2010). It includes a thorough strategy that includes methods for prevention, detection, and response (Lapiedra, 2000). To act in advance to stop potential threats, prevention entails putting strong security measures in place including firewalls, encryption, and access controls. On the other hand, detection mechanisms employ ongoing network traffic monitoring and system behavior analysis to detect unusual activity or security breaches (Reddy et al., 2021). In the event of a security crisis, it becomes imperative to respond promptly and efficiently. Response techniques include taking immediate action to stop and mitigate the effects of a security breach as well as conducting an investigation to determine the nature of the attack and strengthening

defenses in the future (Lapiedra, 2000). Preventive, detection, and response are the three foundations that make up a robust cybersecurity architecture that guarantees a proactive defense against changing cyber threats.

Cybersecurity employs a number of techniques to defend the digital infrastructure of a company from attacks. Strong access control is a crucial component. This involves implementing security measures like two-factor authentication to guarantee that sensitive systems and data are only accessible by those who are authorized. By restricting user access to the bare minimum required for their tasks, the principle of least privilege is essential in reducing the possible impact of a hacked account (Lapiedra, 2000).

Encryption as another strategy to prevent unwanted access to confidential data (Ghouse and Nene, 2019). Data encryption ensures only the correct encryption key can decrypt it, protecting data during transmission and storage from interception and exploitation. Patch management and regular software updates are vital for preventing cyberattacks, as they fix known vulnerabilities. Firewalls serve as the first line of defense by monitoring and controlling network traffic based on security rules, blocking unauthorized access and harmful activities between trusted and untrusted networks.

Building a culture of cybersecurity awareness is crucial. Training keeps employees informed about cyber threats and safe online practices, reducing attack chances. A strong cybersecurity strategy includes firewalls, regular updates, encryption, access controls, and user education. Proactively addressing vulnerabilities helps companies stay secure.

Detection focuses on quickly identifying and resolving security issues. Though costly, it involves continuous monitoring of network activity and system behavior. Intrusion prevention (IPS) and detection systems (IDS) are essential for this. These systems examine network data in order to identify patterns that align with established attack signatures or deviate from typical behavior (Möller, 2023). Finding new or changing threats is further improved by anomaly detection techniques, which provide a baseline of typical system behavior and issue alarms when deviations happen.

Security information and event management (SIEM) a new approach offers a centralized platform for thorough analysis by aggregating and correlating data from many sources, including logs and alarms (Vielberth et al., 2020). Because of this, experts in security can identify patterns and trends that can point to a possible security problem.

Endpoint detection and response (EDR) systems concentrate on spotting malicious activity on specific devices and monitoring and reacting to threats at the endpoint level (Hassan, Bates and Marino, 2020). EDR act like an agent application installed on each endpoint and communicate with a centralized system such as Firewalls. This is especially crucial because endpoints are frequently the target of malware and other online threats.

Security teams monitor user behavior, network traffic, and system logs to detect illegal access, data theft, and suspicious activity. Threat intelligence helps identify risks early. Smart tools track and pinpoint security events instantly, reducing the impact of cyber-attacks and boosting resilience.

A quick response is crucial when a security issue arises. Incident containment involves isolating affected systems to stop the attack from spreading (Computer network isolation, no date). A thorough investigation follows, analyzing the tactics, scope, and root cause. Clear communication with stakeholders is vital for managing the crisis and maintaining trust. Remediation measures restore compromised systems and improve security procedures based on lessons learned.

Post-event analysis helps to enhance cybersecurity by pointing out strengths and weaknesses. In order to strengthen future defenses, the reaction phase entails coordinated efforts to contain, investigate, notify, fix, and learn from incidents.

Prevention, detection, and reaction are necessary for ransomware protection. Strong access restrictions, frequent updates, and user education are examples of preventive measures. Intrusion systems and ongoing monitoring are used in detection in order to quickly isolate threats. A well-thought-out reaction guarantees prompt containment, in-depth analysis, and effective communication, safeguarding information and facilitating recovery.

2.3 Intrusion Detection System IDS

Modern cybersecurity methods must include intrusion detection systems (IDS), which provide a proactive protection against a variety of cyberthreats. An attack signature database is used by signature-based intrusion detection systems to compare system activity and network traffic. In essence, these signatures are patterns or traits linked to recognized risks, which help the system identify matches and raise an alarm when they are discovered. This approach works effectively against recognized and established dangers.

Anomaly-based intrusion detection systems, on the other hand, use a dynamic strategy by creating a baseline of typical network activity. It keeps an eye out for changes from this baseline and sends out alarms when actions deviate from what is expected. Because of this, anomaly-based intrusion detection systems (IDS) are especially good at identifying new or developing threats that signature-based systems can miss. Figure 2.1 shows the architecture of anomaly-based intrusion detection and shows the position of the IDS on the network. It is usually positioned inside the network and close to the router. Where firewall is located outside the local network at the exit port of the router, the print and mail server are located in a different LAN which is connected to other firewall port. Figure 2.2 shows another way of using the host based intrusion detection system where each host device have an IDS agent installed in it. The other servers are outside the network and connected to another firewall port.

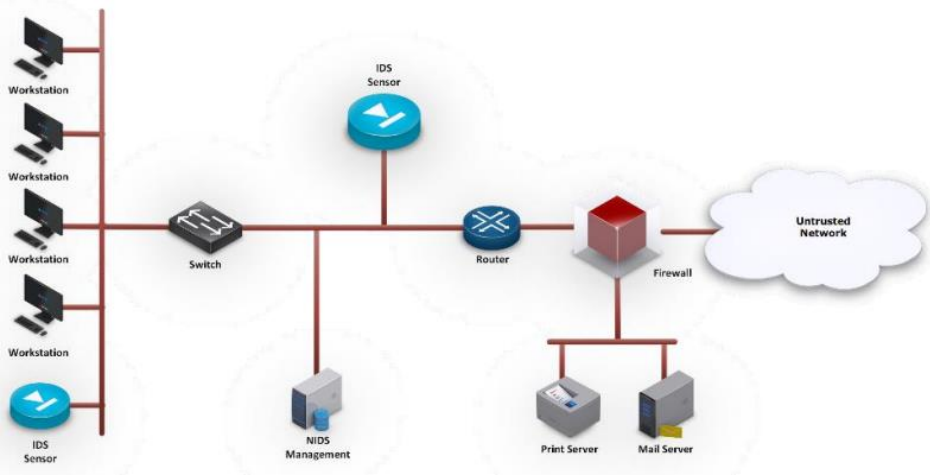


Figure 2.1 Anomaly Based Intrusion Detection (Panagiotou et al., 2021)

Networks and individual devices are two frequent locations for the deployment of intrusion detection systems (host-based IDS). Their combined use with firewalls and other security measures offers a thorough defense-in-depth approach.

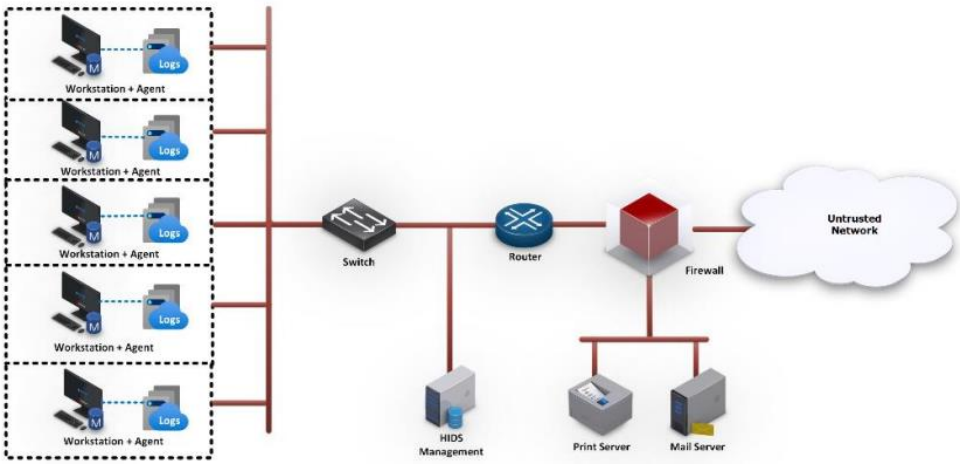


Figure 0.2 Host Based Intrusion Detection (Panagiotou et al., 2021)

Early detection of harmful activity, illegal access, and possible security breaches are among the main objectives of IDS. IDS enables quick identification and notification of suspicious activity to security professionals, facilitating mitigation and quick responses. IDS can also be set up to react to certain threats automatically, which cuts down on the amount of time needed to handle security issues.

2.4 Ransomwares

The integrity and operation of digital systems are at risk by a variety of attack types in the constantly changing field of cybersecurity. A few common attacks are Malware-based attacks, Phishing attacks, Denial of Service (DOS) and Distributed Denial of Service (DDOS) attacks, Man In The Middle (MITM) attacks, and Ransomware attacks.

Malware attacks: such as those caused by viruses, worms, and Trojan horses, are one common type of attack that compromises computers maliciously.

- Phishing attacks: Phishing attacks use false emails or messages to take advantage of people's weaknesses. They deceive people into disclosing personal information and are frequently used as a means of spreading ransomware
- Denial of Service and Distributed Denial of Service: overwhelm systems or networks with traffic, making them inaccessible
- Man in the middle attack: interception and manipulation of communication between parties which might result in data interception or unauthorized access.
- Ransomware attack: A type of virus encrypts or blocks access to data and demands a ransom to be paid by the victim in order to unlock it. These sneaky attacks frequently use encryption-based techniques to obstruct access to important files until a ransom is paid.

Understanding a variety of attack types is essential for establishing strong cybersecurity plans that include predictive defense, prompt detection, and efficient reaction systems to reduce the constant risk of cyberattacks, especially the dynamic and devastating domain of ransomware.

Ransomware is a malicious software that encrypts files or prevents users from accessing their computer systems with the main objective of demanding money from the victim(Aldauiji, Batarfi and Bayousef, 2022). The phrase "ransomware" refers to the act of attackers demanding cryptocurrency, such as Bitcoin or Ethereum, as ransom for system access or a decryption key. The ransomware of today does more than encrypt your data and demand several hundred dollars as ransom. It has the power to destroy whole companies, steal private

information, and even bring down vital infrastructure. The use of strong encryption algorithms and sophisticated distribution techniques to target a wide range of people, businesses, and organizations has significantly increased the sophistication of ransomware attacks. Ransomware works by breaking into a victim's system using a variety of techniques, such as phishing emails, malicious attachments, or taking advantage of software flaws. The ransomware encrypts data after it has entered the system, leaving the user unable to access them. After that, the victim receives a ransom letter with instructions on how to pay, usually within a set amount of time.

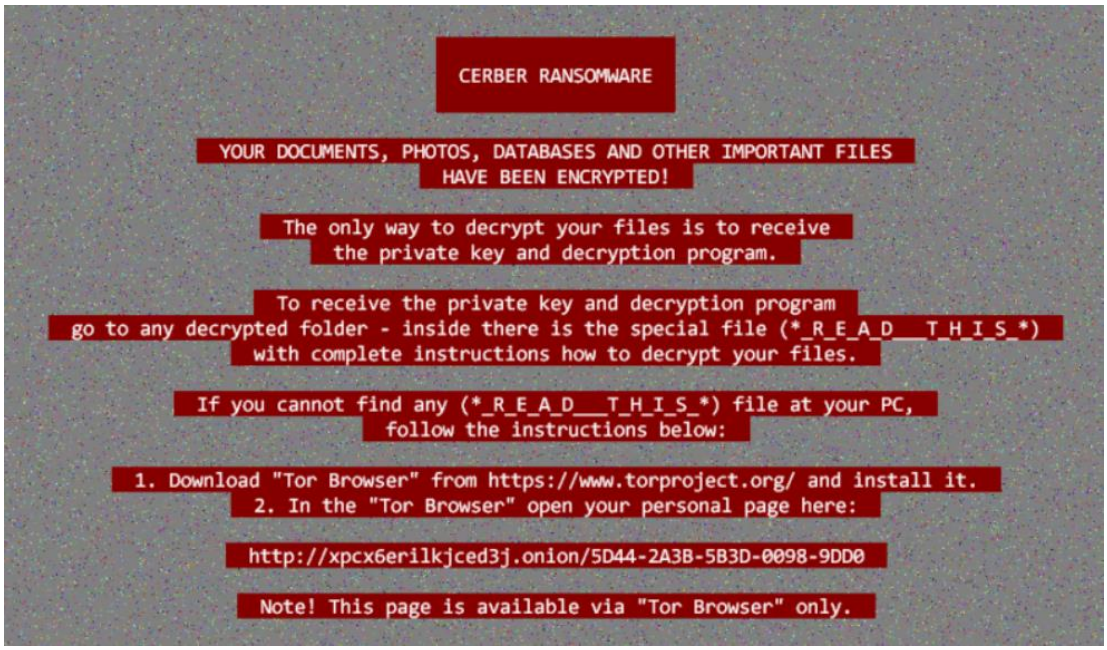


Figure 2.3 Cerber ransomware message after affect our system

2.4.1 Ransomware Attacks Classification

We can classify ransomware attacks into several categories based on their characteristics, methods of distribution, and impact on targeted systems.

1. Simple Ransomware:

Simple ransomware attacks frequently have simple encryption algorithms and don't use any sophisticated tactics. They often employ phishing emails with harmful files or links to target specific victims. The ransomware encrypts data on the victim's computer as soon as it is active, then demands a fee to unlock the encrypted contents.

2. Advanced Persistent Threat (APT) Ransomware:

APT ransomware is a very sophisticated type of malware in which attackers, often well-funded or state-sponsored, use modern techniques. These attacks can be done using Zero-day vulnerabilities, which focused efforts to compromise certain high-level targets - such as government organizations, large corporations, or critical infrastructure –

3. Ransomware-as-a-Service (RaaS):

RaaS has made ransomware more accessible by enabling hackers with less technical expertise to launch assaults. According to this approach, ransomware is created and maintained by developers, who then sell it as a service to other criminals so they may utilize it for their own operations. This strategy makes ransomware attacks more common against a wider variety of targets.

4. Targeted Attacks:

Attacks using targeted ransomware focus on specific companies or people, often customizing the ransomware to take advantage of known flaws in the target's system in order to identify vulnerable targets.

5. Crypto-Ransomware vs. Locker Ransomware:

Files on the victim's computer are encrypted with crypto ransomware, making them unreadable until the ransom is paid. Locker ransomware prevents users from accessing essential components such as the operating system or desktop environment by locking out the entire system or just some features, instead of encrypting the data.

6. Mobile Ransomware:

As mobile devices become more prevalent, ransomware has evolved to target tablets and smartphones. Mobile ransomware typically encrypts data or attempts to erase it after infecting devices through malicious apps or URLs. These attacks highlight the importance of securing mobile platforms as well as traditional computer systems.

2.4.2 Ransomware Stages

Ransomware, a massive cyber threat, unfolds through a series of strategic stages starting by infection process, encryption, Deletion, and ransom request after exchange the keys(Benzekri et al., 2020). The infection process, often initiated by spam emails or self-publishing, represents the initial breach. Therefore, strong encryption algorithms make certain file types inaccessible. In a hard move, attackers delete the original files and may target Microsoft Shadow Volumes, limiting recovery options. The final stage includes a ransom demand, with details of payment and key exchange instructions, forcing victims into a complex decision-making process amid the organized chaos of a ransomware attack.

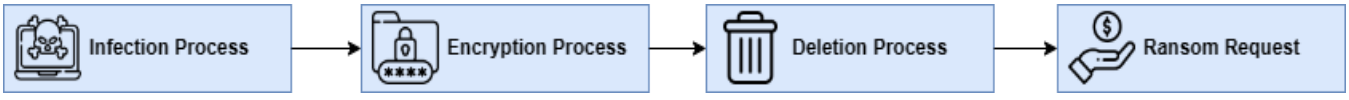


Figure 2.4 Ransomware Stages

An initial infection caused by spam emails with malicious attachments or links as the first step of a ransomware attack. By tricking users into opening attachments or clicking on links, these emails aim to trigger the ransomware payload. Certain advanced varieties possess the ability to replicate themselves by taking advantage of weaknesses in networks to infect further linked devices, and this is the most danger act if it is happened. The ransomware enters the targeted machine during the crucial infection period, which sets off other harmful actions.

Once within the system, ransomware encrypts specific kinds of files using encryption methods like Rivest-Shamir-Adleman (RSA) or Advanced Encryption Standard (AES). Without the associated decryption key - which the attackers have - this encryption makes the data inaccessible and unavailable. Documents (.doc,.xls), photos (.jpg), and several other data formats are often targeted file types, guaranteeing that the attack will have a significant impact. These make the encryption step is an important step in which the attackers gain control over the victim's data and prepare the basis for their ransom demands in the encryption procedure.

Ransomware frequently destroys the original files after successfully encrypting them. This accomplishes two goals at once: it makes the victim realize how urgent the situation is and increases the attack's effect by removing any chance of recovery. Furthermore, Microsoft Shadow Volumes, which are backup copies of data made by the Windows operating system, may be the target of some highly advanced ransomware outbreaks. Attackers also reduce the victim's chances of recovering their files without having to pay the ransom by deleting these

shadow folders. Our machine learning model was specifically applied in a Windows system to address these challenges effectively.

After the attackers have erased the original copies and encrypted the victim's data, they proceed to the next step and demand a ransom. Usually, the victim receives a ransom letter with contact information for the attacker as well as instructions on how to pay, frequently in bitcoin. The attack ends with the exchange of keys, which entails paying the ransom in return for the decryption key. The victim may be able to access their data again if the decryption key is released following a successful payment. There is no assurance that the attackers would keep their end of the agreement, and paying the ransom is not the best option.

2.4.3 Ransomware as a Service

Ransomware attacks have become more dangerous, especially with the spread of ransomware as a service in mid-2010, as it became possible for any non-technical person to use this attack several targets(DarkSide Ransomware as a Service (RaaS) - United States Department of State, 2023). This implied that anyone with a basic understanding of technology and internet access could possibly become a distributor of ransomware, which explains the widespread of these attacks on the Internet.

Ransomware as a Service (RaaS) has surfaced in the area of cybercrime, turning ransomware attacks into easily accessible and commercial criminal companies. Given this serious threat, cybercriminals -often highly skilled at coding- create and develop sophisticated strains of ransomware and then make them accessible to less technical people, known as affiliates or clients.

The affiliates may start their own ransomware attacks by using the RaaS platform. RaaS suppliers usually keep a percentage of the ransom payments as a fee, creating a dark economy that profits from ransomware attacks. This strategy has made it possible for a wider variety of threat actors to participate in cyber extortion, which has led to the widespread spreading of ransomware. RaaS platforms have made ransomware more accessible, which has increased the threat environment overall and highlighted the necessity of strong cybersecurity measures to counteract this profitable and decentralized criminal activity.

2.4.4 Common Ransomware Attacks

When the **DarkSide** Ransomware appeared in August 2020, it was quickly used to initiate a global campaign of ransomware attacks in more than 15 countries, focusing on a variety of business sectors, including technology, financial services, manufacturing, legal services, and others(DarkSide Ransomware as a Service (RaaS) - United States Department of State, 2023). The 5,500-mile pipeline that transports 45% of the oil consumed on the US East Coast was temporarily shut down by Colonial Pipeline Company in May 2021 as a result of a ransomware incident caused by the DarkSide gang(DarkSide Ransomware as a Service (RaaS) - United States Department of State, no date). By offering this motivation, the United States is demonstrating that it is committed to protecting ransomware victims anywhere from exploitation by cybercriminals. A reward of up to $10,000,000 is being offered by the U.S. Department of State for information that leads to the identity or location of any person or people who are related to the organized crime gang that uses the DarkSide ransomware(DarkSide Ransomware as a Service (RaaS) - United States Department of State,

no date). Hospitals, schools, universities, and non-profit organizations are not targets of this type of ransomware, with the possibility of targeting systems running Windows and Linux.

The danger of this ransomware is that it is considered as a type of ransomware as a service (RaaS) which is greatly increases the speed of its spread. The DarkSide ransomware family encrypts files using RSA-1024 encryption and a mix of asymmetric and symmetric cryptography using a key which is randomly generated.

The infamous ransomware version **CryptoWall** first appeared in 2015, it was responsible for 406,887 attempted infections and $325 million in damages(Digital Scholar et al., 2017). With the use of sophisticated encryption methods, most notably the AES (Advanced Encryption Standard) algorithm, CryptoWall quickly established itself as a powerful adversary, encrypting files with a degree of complexity that presented significant obstacles for victims trying to retrieve their data(KARA, AYDOS and BOZKIR, 2020). In a well-known example, a healthcare company in USA was the target of a catastrophic CryptoWall attack that encrypted critical medical data and patient information, healthcare is a target for ransomware, given that the possibility of payment is very large due to the sensitivity of the lost data. A significant Bitcoin ransom was requested by the attackers in exchange for the decryption key, demonstrating the seriousness and focused nature of CryptoWall operations. The primary target of this ransomware was Microsoft Windows operating systems. It gained access to computers by taking advantage of vulnerabilities in the operating system and used social engineering techniques to expose confidential data.

Early in 2016, the ransomware **Cerber** made an appearance on the cyber threat scene and soon proved to be a powerful danger(Kurniawan and Riadi, 2018b). Cyberattacks increased

dramatically within its first year of existence, alarming both people and corporations. To safely lock victims' data, Cerber uses a highly advanced encryption method that frequently combines RSA or AES with RC4 cryptography(Kurniawan and Riadi, 2018). Interestingly, Cerber functions as a ransomware-as-a-service (RaaS), giving hackers access to its nefarious powers via a paid subscription approach. This strategy had a major role in the broad adoption of it by hackers with differing degrees of technical proficiency.

Cerber targets a wide range of operating systems, including Windows and Linux, and it is platform independent. It is ability to operate across platforms expands its influence and reach, making it a serious threat in a variety of computer contexts. One prominent case of Cerber ransomware attack occurred in 2017, when the Hollywood Presbyterian Medical Center in Los Angeles fell victim to a devastating cyber-attack. The ransomware infiltrated the hospital's network, encrypting critical patient records, medical documents, and other essential files, crippling daily operations. The attackers demanded a ransom of 40 bitcoin (approximately $17,000 at the time) in exchange for the decryption key, leaving the medical center with a difficult decision to make.

TeslaCrypt ransomware first appeared on the cybersecurity scene in 2015, introducing a new wave of threats that attracted attention due to its targeted approach. In its initial iterations, TeslaCrypt encrypted files by looking for 185 file extensions associated with 40 distinct games, such as World of Warcraft, Call of Duty, World of Tanks, and Minecraft(17 Ransomware Examples & How They Occurred | UpGuard). These files included player profiles, custom maps, game modifications, and save data that were kept on the victim's hard

drive. Word, PDF, JPG, and other file extensions were also encrypted by more recent TeslaCrypt versions, requiring the user to pay a $500 Bitcoin ransom to unlock the files.

TeslaCrypt demonstrated encryption using a combination of RSA and AES encryption algorithms, The "ecc" extension on encrypted files is renamed, indicating the usage of an Elliptic Curve Cryptographic (ECC) technique. The ECC algorithm is not used by the ransomware to encrypt data; rather, it is used to generate Bitcoin addresses(TeslaCrypt Ransomware Threat Analysis | Secureworks). This method holds victims' files hostage, and demands a ransom payment in exchange for revealing their data. Unlike some other ransomware variants, TeslaCrypt focused mostly on Windows platforms, with a particular focus on exploiting vulnerabilities within the Windows operating system.

Table 2.1 lists used types of ransomware along with their characteristics. DarkSide, which emerged in 2020, uses RSA-1024 encryption, which involves both symmetric and asymmetric cryptography. It operates as a Ransomware-as-a-Service (RaaS) and targets both Windows and Linux platforms. CryptoWall, from 2015, uses AES encryption, focusing on symmetric cryptography and targeting Windows systems. Cerber, starting in 2016, employs AES and RSA/RC4 encryption, combining symmetric and asymmetric cryptography. Like DarkSide, it's RaaS-enabled and targets Windows and Linux. Finally, TelsaCrypt, appearing in 2015, uses RSA/AES/ECC encryption, again using both symmetric and asymmetric cryptography, but unlike DarkSide and Cerber, it doesn't operate as RaaS and only targets Windows platforms.

Table 2.1 Used Ransomwares

Ransomware	First Appear	Encryption	Cryptography type	RaaS	Platform target
DarkSide	2020	RSA-1024	Symmetric and asymmetric	✓	Windows and Linux
CryptoWall	2015	AES	Symmetric	×	Windows
Cerber	2016	AES / RSA & RC4	Symmetric and asymmetric	✓	Windows and Linux
TelsaCrypt	2015	RSA / AES / ECC	Symmetric and asymmetric	×	Windows

2.5 Machine Learning

Machine learning is a field of artificial intelligence focused on the development of algorithms and models that allow computers to learn from data and improve their performance over time. It includes a range of methods and strategies intended to provide machines the ability to predict events, identify trends, and come to judgments without explicit programming. As a result, machine learning systems are capable of using an almost infinite variety of models and algorithms to learn from experience and adapt and improve their output. Machine learning has the ability to improve a wide range of different areas, including healthcare, education, government, financial services, and transportation.

A key concept in machine learning is supervised learning, which is the process of training algorithms on labeled datasets so they can understand the complex connections between input attributes and matching output labels. During this procedure, a training dataset containing input data matched with the appropriate output is given to the algorithm. The algorithm optimizes its internal settings to maximize its prediction accuracy on new, unseen data through repeated learning. Algorithms such as neural networks, decision trees, support vector machines, and linear regression are all included in supervised learning. Supervised learning is rich because it may be used to a wide range of fields, including natural language

processing, picture and audio recognition, and predictive analytics. The quality and representativeness of labeled training data is critical to the success of supervised learning, highlighting the importance of careful curation of the dataset to ensure the algorithm performs effectively in a variety of real-world conditions. Supervised learning is useful, but it also has disadvantages. In particular, it relies on labeled data, which can be expensive and time-consuming to collect in some situations. However, the ability of supervised learning to make accurate predictions and automate difficult decision-making processes highlights how important it is in developing AI applications in a variety of sectors.

An important aspect of machine learning is unsupervised learning, where algorithms search through unlabeled samples for different patterns and structures. In contrast to supervised learning, the algorithm searches for associations or clusters within the data rather than pre-defining output labels. In unsupervised learning, clustering is an iterative function that groups similar data points together, while dimensionality reduction methods extract important features. Unsupervised learning works well when there is a lack of labeled data and is frequently used in customer segmentation, anomaly detection, and exploratory data analysis. Even if there are difficulties in interpreting the results without clear guidance, it is still a crucial technique for uncovering hidden patterns in a variety of data sets.

Both supervised and unsupervised learning components are included in semi-supervised learning. Using this method, a dataset with both labeled and unlabeled data is used to train the algorithm. The model uses the labeled data to identify trends and then calculates this understanding to predict the unlabeled data. This is particularly helpful when gathering tagged data requires a lot of resources.

2.5.1 Machine Learning Algorithms

In this study, state-of-the-art machine learning algorithms were employed to analyze and predict ransomware behavior. The algorithms utilized include Random Forest, known for its robustness and ability to handle large datasets while maintaining high accuracy; K-Nearest Neighbors (KNN), a versatile algorithm for pattern recognition and classification tasks; and Gradient Boosting, which combines the strengths of multiple weak learners to achieve strong predictive performance. These algorithms were selected based on their proven effectiveness in cybersecurity applications and their capacity to handle the complex and dynamic nature of ransomware threats.

Random Forest is a powerful ensemble learning method that combines multiple decision trees to make predictions. Its strengths lie in its ability to handle large datasets efficiently, making it robust against overfitting. Random Forest is particularly effective in scenarios with high-dimensional data and complex relationships between features. Additionally, it provides built-in feature importance scores, aiding in feature selection and interpretation. However, one of its weaknesses is its computational cost, especially when dealing with a large number of trees. Tuning hyperparameters is crucial to prevent overfitting and optimize performance. The execution time of Random Forest varies depending on the number of trees and the complexity of the dataset, often ranging from moderate to high due to its ensemble learning nature.

K-Nearest Neighbors (KNN) is a straightforward and intuitive algorithm used for classification and regression tasks. Its strengths include simplicity of implementation and the ability to handle non-linear decision boundaries. KNN is non-parametric, making it suitable for data with unknown or complex distributions. However, its weaknesses include high

memory consumption as it requires storing all training data for prediction. Additionally, KNN can be computationally expensive during prediction, especially with large datasets or high-dimensional feature spaces. It may also suffer from the curse of dimensionality and perform poorly with noisy or irrelevant features. The execution time of KNN depends on the size of the training set and the number of neighbors considered during prediction, often resulting in high execution times for large datasets.

Gradient Boosting is an ensemble learning technique that builds strong predictive models by sequentially combining weak learners, typically decision trees. Its strengths lie in its ability to produce highly accurate predictions and handle complex relationships and non-linearities in the data. Gradient Boosting can capture feature interactions and importance, making it suitable for a wide range of tasks, including classification and regression. However, one of its weaknesses is its susceptibility to overfitting, especially with deep trees and insufficient regularization. Training time can also be longer compared to simpler algorithms due to the sequential nature of boosting. The execution time of Gradient Boosting depends on factors such as the number of boosting iterations, tree depth, and dataset complexity, often resulting in higher execution times compared to simpler algorithms.

2.5.2 Classification

One of the main supervised learning functions in machine learning is classification, where the computer learns to assign incoming data points to specific categories or main categories. Using a labeled dataset, where each sample is associated with a specific class or category, the model is trained. Decision regions, often referred to as decision borders and decision

surfaces, are created by partitioning the input space. Using this labeled data, the trained model generalizes to predict new, unseen events.

A demonstration of the decision surface is shown in Figure 2.5. The class labels in this example are shown by the two distinct colors, blue and orange. Two lines create a border

for the presentation of new data between the points, as seen in the image below. If it is not shown in an extremely heated environment, it will be clear which class it belongs to.

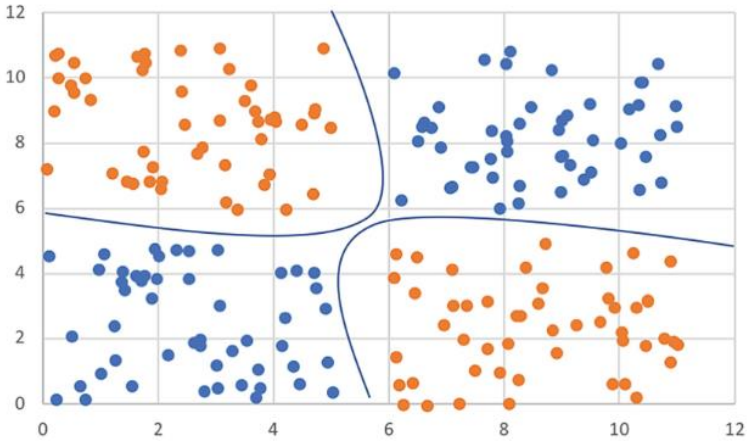


Figure 2.5 Decision surface (*Why Is Logistic Regression a Classification Algorithm? | Built In*, 2023)

Different types of data and problem areas are represented by different classification methods. Popular classifiers for image identification, sentiment analysis, spam detection, and support vector machines include neural networks, decision trees, logistic regression, and support vector machines.

Depending on the nature of the classification challenge, metrics such as accuracy, precision, recall, and F1 score are frequently used to evaluate the effectiveness of classification models. The ability of classification algorithms to provide accurate predictions finds widespread use in many sectors, providing useful tools for automating the decision-making process.

2.5.3 Dataset Preprocessing

Data preprocessing, which involves organizing, transforming, and cleaning raw data to improve model performance and accuracy, is an essential step in the machine learning process. The following preprocessing actions help get data ready for efficient machine learning:

- **Handling Missing Data:** Techniques such as imputation and missing value elimination make that the dataset is full and ready for analysis.

- **Scaling Features:** By ensuring that all variables contribute equally to training the model, standardizing or normalizing features helps avoid a single feature being dominated by its size.

- **Encoding Categorical Variables:** Converting categorical variables into numerical format allows algorithms to interpret and learn from these variables effectively.

- **Removing Outliers:** Preserving the integrity of the dataset by detecting and removing outliers helps avoid biased model training.

- **Feature Engineering:** A model's ability to identify relevant patterns in data can be improved by adding new properties or changing existing properties.

Achieving accurate and dependable machine learning models requires proper data preparation. It guarantees that the input data is clean, well-structured, and suitable for training models that perform well when applied to new, untested data.

Conclusion of Chapter 2, unveils the symbiotic relationship between cybersecurity and machine learning, forging a formidable arsenal against ransomware incursions. The cyber defenses, from prevention strategies to real-time detection and swift response protocols, underscores the dynamic nature of modern cybersecurity ecosystems. IDS is essential for preventing new cyberattacks, and examining several ransomware types reveals the

complexity of these attacks. The major transformation that is RaaS requires smart defenses. Ransomware attacks may be predicted and prevented using machine learning, especially when data is handled carefully. Cyber safety advances with the explanation of ransomware detection and prediction ideas in Chapter 2.

3 Chapter Three
Literature Review

3.1 Literature Review

Over the past ten years, the field of cybersecurity has undergone significant change, mostly due to the advent of ransomware as a serious threat that calls for the development of novel detection and analysis techniques. In this chapter, we conduct a chronological literature review covering the period from 2015 to 2022 and explore important research projects that try to clarify the complexities of ransomware analysis and detection. As cyber attackers continuously enhance their tactics, the need for adaptive and robust protection measures has become increasingly critical. This review delves into a strategic timeline, offering insights into the evolution of ransomware attacks and the corresponding advancements in detection systems. It encompasses pivotal research that has not only deepened our understanding of ransomware behavior but also led to the development of effective detection models. By tracing the trajectory of research efforts, this chapter sheds light on the dynamic landscape of ransomware detection, setting the stage for the exploration of novel techniques and methodologies in subsequent sections.

The publications were chosen examine several methods for ransomware detection, including feature extraction methods and supervised learning algorithms. The articles also emphasize the significance of employing various datasets for machine learning model training and testing, as well as the requirement for ongoing upgrades to stay up with increasing ransomware threats. Overall, these articles give useful recommendations for further research in this field and significant insights into the possibilities of machine learning for ransomware detection.

In 2015 the author in (Cabaj *et al.,* 2015) used dynamically analyzes CryptoWall ransomware network behavior using a honeypot and an automated technique for analyzing malware in real-time. for this purpose, an environment called Maltester which contains four virtual machines is used. The author was able to extract some features that belong to CryptoWall ransomware such as some proxy servers, the protocols used, and the hardcoded addresses of some servers. The output was good, but it is not behavioral even for this type of ransomware, as most addresses and proxy servers may change easily. Another research (International ISC Conference on Information Security and Cryptology 12. 2015 Rescht et al.) Proposing a unique way to identify ransomware that is incredibly resilient. Since hybrid ransomware poses the biggest threat, they focus on it. High survivability ransomware is defined as malware that infects users and only allows the encryption process to be reversed in order to recover data. Finally, releasing one victim does not automatically release the remaining devices that were the target. The public key obtained from Command and Control (C&C) to encrypt the compromised system is the main target of the authors' detection method. In order to communicate with command-and-control candidates to retrieve the key, it targets the range generation algorithm. They propose a method that can identify DNS queries sent by domain generation algorithms (DGA). The results were good, but there is a large group of ransomware families that do not rely on DNS, which in turn results in them not being detected by their system.

In 2016 the research in (Sgandurra *et al.,* 2016) proposed a method using machine learning for dynamic analysis and classification of ransomware called EldeRan. The author uses Regularized Logistic regression algorithm with Maximum Information Coefficient (MIC)

feature selection on many different ransomware families such as Critoni, Crypto locker, CryptoWall, Kovter, MARSNU, and Tesla Crypt. EldeRan keeps track of a number of operations carried out by programs during the initial stages of installation. The results were promising as the model was able to detect ransomware under the receiver operating characteristic (ROC) curve of 0.995, but the problem remains with some types of ransomwares that remain dormant for an indefinite time until they start encrypting the victim's device. The author in (Tseng et al.) are looking into several ransomware families' such as Locky, Cerber, and CryptoWall with various signatures, the author suggested a deep learning-based detection method to distinguish between ransomware traffic and benign traffic by investigating the HTTP requests and TCP protocols. Their accuracy rate was 93.92%, although it was heavily reliant on the HTTP protocol.

In 2017, the WannaCry family is the main focus of the research in (Institute of Electrical and Electronics Engineers. Bangladesh Section and IEEE Communications Society, 2017), the authors introduced "RansHunt," a hybrid analyzer that used SVM as a classifier to analyze samples of both benign and ransomware software. With 1,283 samples and 21 ransomware families, the study compared the accuracy of the detection with tools such as Virus Total, obtaining an accuracy of 97.10%. A number of features were taken from VirusShare as the IDAPro was used for static analysis, and the Cuckoo sandbox was used for dynamic analysis. A dataset consisting of 360 samples of ransomware, 532 samples of malware, and 460 samples of benign software was used for testing on Windows XP and Windows 7 systems. The results are promising, but it has been observed that there is a greater reliance on static features, which many types of ransomwares can avoid by remaining dormant for a certain

period and then activating themselves. In addition to adopting the top 100 features, which will require more time.

In 2018 the author in (Kurniawan and Riadi, 2018b) analyzed Cerber ransomware using a dataset installed, by reconstructing the events and checking packets, the author determined when the first time the hostname and IP address is infected, He was also able to trace how the Cerber ransomware got to the victim's device. No behavioral activity was extracted through the analysis, only the IP addresses and domain names that were used by the ransomware were detected. The other research in (Daku, Zavarsky and Malik, 2018) uses of behavioral reports from 150 ransomware samples across 10 families. It evaluates three machine learning classification algorithms (KNN, Naïve Bayes, and J48 Decision Tree), with J48 utilizing nine features to achieve 78% accuracy. Data was gathered from VirusTotal, and the two methods of obtaining malware samples - from classified samples and from the wild - were discussed. WEKA was utilized to process the dataset in order to classify ransomware, highlighting the potential of machine learning - specifically J48 - to improve ransomware detection and classification.

In 2019 the author in (Lee, Lee and Yim, 2019) proposed a technique to find ransomware-infected files in backup systems, and measure the property of the encrypted file using the entropy method. Machine learning is used to categorize infected files. using system files, document files, image files, source code files, executable files, compressed files, and other items that are essential to the system and users to compare the entropy of the original files with the encrypted files. By identifying ransomware-infected files that have been synchronized to the backup system, the suggested solution can recover the original file from

the backup system. To keep track of the potential consequences different varieties of ransomware can have on files, they need to make sure they test and train this system on a very wide range of malware.

In 2020 the author in (Hwang *et al.,* 2020) categorize system activity as either ransomware or benign, utilize a Random Forest (RF) classifier and an Artificial Neural Network (ANN) classifier. A set of features that were gathered from a dynamic investigation of system activity are used to train the classifiers. The findings demonstrate that in terms of accuracy, precision, and recall, the RF classifier performs better than the ANN classifier. The proposed approach's usage of machine learning techniques shows how useful these methods are for creating precise and effective ransomware detection systems. The study focuses on a two-stage strategy that necessitates machine learning and dynamic analysis techniques, which may be costly in terms of time and processing. This makes the method less scalable, especially for complex systems. In order to improve detection capabilities on Windows systems, the study in (Ahmed, Koçer and Al-Rimy, 2020) presents a methodology for dynamically studying the behavior of High Survivable Ransomware (HSR). The suggested approach uses Term Frequency-Inverse Document Frequency (TF-IDF) in a sandbox setting to extract relevant information and shows better performance in detecting new malware. Detection models are created and combined with notable features by using supervised machine learning methods, particularly Support Vector Machine (SVM) and Artificial Neural Network (ANN). High accuracy and a low false positive rate are achieved by the method, especially in the early phases of HSR attacks. Cuckoo Sandbox, a well-known open-source program for automated malware research, is used in the study in a controlled virtualized environment with a guest

PC running Windows XP SP3. It takes noteworthy efforts to successfully run ransomware and monitor its actions, such as turning off security protections. The authors investigate fourteen distinct families of ransomware: WannaCry, Revento, Dirty Decrypt, Cerber, Kollah, Peyta, CryptoWall, and TeslaCrypt. SVM achieves 97.6% accuracy using the top 40 features, whereas ANN exceeds with 98.79% accuracy using the top 30 features, according to the experimental data, which show outstanding accuracy. The methodology presents a strong approach to machine learning-driven detection of HSR and behavioral-based analysis, offering insightful information on how to improve cybersecurity defenses against ransomware attacks that are always developing. But the problem lies in using an old operating system to activate the ransomware on it and relying on a virtual environment for the experiment, as many types of ransomwares can now identify this environment and not activate themselves on it.

In 2021 The authors in (Umar, Riadi and Kusuma, 2021) provide a live forensic examination in a simulated environment of the Conti ransomware attack on a computer network. The study makes use of a specially created virtual environment to simulate the attack and assess how well live forensic procedures for spotting and addressing the ransomware attack work. The study also highlights a number of crucial characteristics that the Conti malware utilized to avoid detection and carry out the attack. These features involve manipulating Windows Management Instrumentation (WMI) and using Process Injection to run the ransomware payload. The investigation is restricted to a single ransomware version and a particular network setup, which might not be indicative of other attack situations. The paper (Alotaibi and Vassilakis, 2021) explores the use of Software-Defined Networking (SDN) as a detection

and mitigation tool for self-propagating ransomware. The proof of concept used in this study is the BadRabbit ransomware. Deep packet inspection, ARP scanning detection, packet header inspection, honeypot, and SMB checker are the five modules that make up the suggested system. Notably, novel SDN-based techniques for ransomware detection are used by SMB checker, honeypot, and packet header inspection. The efficiency and performance measures of the system, including TCP, ping latency, CPU utilization, and detection time, are evaluated in the study. Some other ransomware families are used, such as NotPetya and WannaCry, the authors conducted experiments using a testbed featuring three PCs and one gateway, running Windows 10 and 7 operating systems. The interconnected network aims to test the propagation patterns of BadRabbit ransomware. In contrast to traditional Intrusion Detection and Prevention Systems (IDPS), the study highlights the limitations of IDPS in terms of performance and efficiency, emphasizing the potential advantages of SDN in addressing the challenges posed by self-propagating ransomware in the networks. The results are good and promising, but the investigation relied on detecting the spread of the ransomware on the network and it is not suitable for use in the live network due to complexity in integration, potential interoperability issues with legacy systems, and increased security risks, requiring careful planning for scalability, reliability, and maintaining network operations.

In 2022 The authors in (Berrueta *et al.,* 2022a) suggests a machine learning-based method for spotting crypto-ransomware intrusions in scenarios including file-sharing networks and encrypted traffic. To analyze network data and spot potential ransomware attacks, the method combines deep learning models and feature extraction methods. The findings demonstrate

that the suggested method successfully detects ransomware attacks with a rate of 86% in 22.3 seconds. The proposed strategy is only evaluated in the study using a small number of ransomware variants and network conditions. For further research to assess the approach's efficacy and generalizability, a wider variety of ransomware attacks and network setups must be used, and 30.2 second may be too long for early detection. The research in (Arfeen *et al.,* 2022) suggests a framework that allows for regular memory dumps for thorough and precise examination. The framework was developed using WannaCry, HiddenTear, Cerber, TeslaCrypt, and Vipasana ransomware. Using the XGBoost method, binary classifications were performed on the dataset with the suggested framework. Consequently, 88% accuracy in ransomware was achieved using 5 ransomware samples, the results are good, but the number of samples tested is relatively small.

Table 3.1 comprehensive analysis for ransomware detection

Research	**Source of Dataset**	**Algorithm**	**Number of samples**	**Source of samples**	**Results**
(Cabaj *et al.*, 2015)	×	×	6	malware-traffic	Analysis for CryptWall
(International ISC Conference on Information Security and Cryptology 12. 2015 Rescht *et al.*, 2015)	×	×	20	Malwaretips.com BleepingComputer.com	100%
(Sgandurra *et al.*, 2016)	Generated	Logistic Regression	582	VirusShare.com	93.3%
(Tseng *et al.*, 2016)	Generated	Neural Networks	×	×	93.9%

(Institute of Electrical and Electronics Engineers. Bangladesh Section and IEEE Communications Society, 2017)	Generated	SVM	360	VirusShare.com	97.1%
(Kurniawan and Riadi, 2018b)	Malware-traffic	×	×	×	Analysis for Cerber
(Daku, Zavarsky and Malik, 2018)	VirusTotal	J48 Decision Tree	150	×	78%
(Lee, Lee and Yim, 2019)	Generated	KNN	×	×	100%
(Hwang *et al.*, 2020)	Generated	RF & ANN	×	VirusShare.com	97.3%
(Ahmed, Koçer and Al-Rimy, 2020)	Generated	SVM & ANN	1254	VirusShare.com VirusTotal.com	98.7%
(Umar, Riadi and Kusuma, 2021)	Generated	×	1	×	Analysis for Conti
(Alotaibi and Vassilakis, 2021)	Generated	×	×	×	SDN-Based IDS
(Berrueta *et al.*, 2022a)	Generated	Neural Networks	70	×	100%
(Arfeen *et al.*, 2022)	Generated	XGBoost	5	VirusTotal.com	88%

In Table 3.1, each number represents a different way researchers studied ransomware detection. We can divided the papers into two groups according to how each approaches analysis and detection. Studies that use analytic techniques primarily to comprehend ransomware behaviors or network traffic patterns fall under the first group. For example, study (Cabaj *et al.*, 2015) analyzes network-traffic data to analyze CryptWall activity, while research (Kurniawan and Riadi, 2018b) employs similar analytic techniques to look at Cerber's characteristics. Furthermore, studies (Daku, Zavarsky and Malik, 2018) and (Umar,

Riadi and Kusuma, 2021) examine the use of an SDN-Based IDS and malware behavior analysis for Conti, respectively.

In contrast, the second category comprises studies that leverage machine learning algorithms for malware detection and classification tasks. Notable works in this category include research (International ISC Conference on Information Security and Cryptology 12. 2015 Rescht et al., 2015), which utilizes manual analysis from sources like Malwaretips.com and BleepingComputer.com, achieving a 100% detection rate. Research (Sgandurra et al., 2016) employs Logistic Regression on a dataset from VirusShare.com, achieving a 93.3% accuracy rate. Research (Tseng et al., 2016) explores the use of Neural Networks for malware detection, while research (Institute of Electrical and Electronics Engineers. Bangladesh Section and IEEE Communications Society, 2017) achieves a 97.1% detection rate using SVM on VirusShare.com data. Similarly, research (Lee, Lee and Yim, 2019) applies KNN for malware detection, achieving a 100% detection rate, and research (Hwang et al., 2020) combines Random Forest (RF) and Artificial Neural Networks (ANN) to achieve a 97.3% detection rate. Moreover, research (Ahmed, Koçer and Al-Rimy, 2020) combines SVM and ANN for classification tasks, achieving a high accuracy rate of 98.7%. Finally, research (Berrueta et al., 2022a) utilizes Neural Networks for detection, achieving a 100% detection rate, and research (Arfeen et al., 2022) implements XGBoost on VirusTotal.com data, achieving an 88% accuracy rate.

As we conclude our investigation of ransomware detection techniques, we've observed a variety of approaches to the issue. In contrast to others that concentrate on a single feature, such as DNS or HTTP, we examined many factors in parallel to strengthen our

system's defenses against various attack tactics. Additionally, we examined other ransomware types with a large number of samples, which increased the reliability and practical applicability of our findings. With this wide-ranging approach, our research is well-positioned to address the dynamic issues of cybersecurity.

4 Chapter Four
Methodology

4.1 Introduction

Chapter 4 delves into the heart of our research endeavor, presenting a comprehensive exploration of the suggested model designed to combat ransomware threats through advanced machine learning techniques. The proposed model represents a culmination of strategic planning, technical ingenuity, and a deep understanding of cybersecurity dynamics. Within this chapter, we unveil the architecture, functionalities, and operational framework of our proposed model, elucidating its potential to revolutionize ransomware detection and prediction.

The testbed serves as a crucible where our model undergoes rigorous evaluation, stress-testing its capabilities under simulated real-world conditions. By meticulously collecting and analyzing network traffic data, including HTTP, DNS, and TCP protocols with TCP SYN, RST, and ACK flags set. We gain invaluable insights into the behavioral patterns and signatures of ransomware activity. This empirical foundation paves the way for informed decision-making and algorithmic refinement within our model.

A pivotal aspect of our methodology is the construction and utilization of a robust dataset, meticulously curated to represent diverse ransomware scenarios and network contexts. Preprocessing techniques are employed to cleanse, normalize, and enhance the dataset, ensuring optimal performance and accuracy during model training and evaluation. Performance measurement metrics serve as litmus tests, quantifying the efficacy and efficiency of our model in real-world scenarios.

4.2 Proposed Model

The proposed ransomware detection model is a comprehensive framework created to efficiently identify and classify network data, and differentiate between benign and ransomware activities. Figure 4.1 shows the model and highlights the important stages from data collection to final classification. Improving model accuracy and reliability requires careful consideration of each stage.

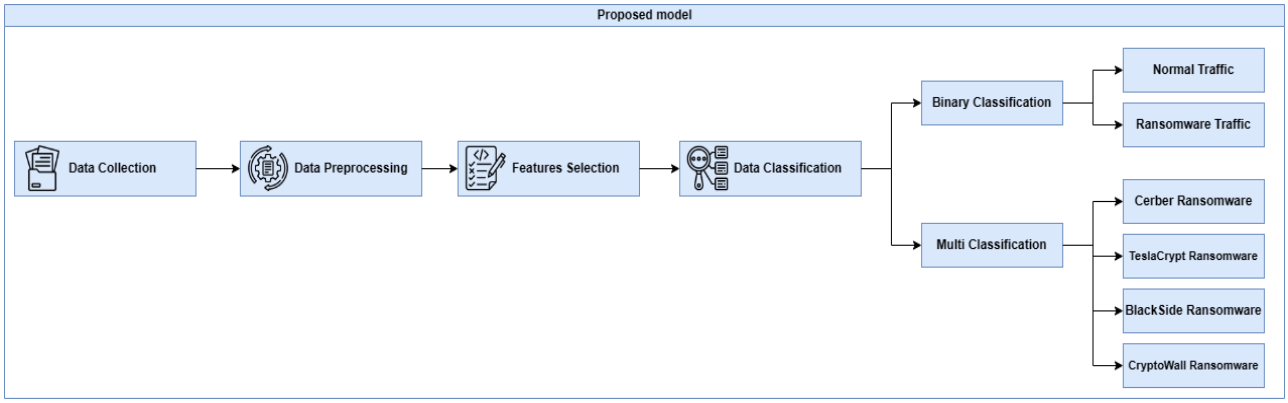


Figure 4.1 Proposed model

The model begins by collecting data about network traffic, which captures a wide range of actions that occur in the network environment in the isolated testbed. The basis for further investigation and model training is these data. Preprocessing is done on raw data to make sure it is suitable for machine learning and of high quality. During this stage, the dataset is optimized for model training by managing missing values, scaling features, encoding categorical variables, and eliminating outliers. One of the most important steps in locating and keeping the most important attributes from the dataset is feature selection. Through this method, the model's efficiency is increased by concentrating on important features that make significant improvements to the classification job. Classification step is where the suggested

model resides at its core. In order to do this, a machine learning model must be trained to separate network traffic into different classifications. Accurate data classification by the model is essential for efficient ransomware detection.

The model splits into two directions during the data classification stage. First, there is Binary Classification, which separates ransomware activity from regular network data. This binary differentiation is fundamental to detecting any dangers in the network. The traffic is categorized as Ransomware traffic and Benign Traffic. the other classification is the multi-classification. The ability of the model to distinguish between certain ransomware attacks is expanded by the second step in the data categorization process. This covers the ransomware families Cerber, TeslaCrypt, DarkSide, and CryptoWall. Gaining a more detailed picture of the kind of ransomware in the network is made possible by multi-class categorization. The ransomware traffic is classified into Cerber, TeslaCrypt, DarkSide, or CryptoWall.

Using specific ransomware families like Cerber, which has been analyzed in some research but often without the application of machine learning techniques, provides an opportunity to bridge the gap between traditional analysis methods and advanced machine learning approaches. Additionally, including newer families like DarkSide allows for a more comprehensive understanding of emerging threats and the evolution of ransomware tactics. By focusing on these prominent families such as TeslaCrypt and CryptoWall alongside Cerber and DarkSide, the analysis can capture a wide spectrum of ransomware behaviors, from established patterns to evolving strategies, leading to a more nuanced and effective approach in cybersecurity research and defense.

The proposed model provides a comprehensive and adaptable method for detecting ransomware, offering cutting-edge visibility into network activity and facilitating preventive measures against ever-changing cybersecurity risks. The model's adaptability is improved through binary and multi-class classification components, which enable it to handle a variety of ransomware situations in network traffic.

4.3 Testbed

The testbed, consisting of three computers, switch and a router, was created with care to capture network activity associated with several ransomware families. With one computer serving as a potential victim system, another monitoring traffic, and a third adding further complexity to the network, this isolated setup simulates a real-world network configuration. The switch ensures isolation of the controlled environment by acting as the main hub of communications. By facilitating the monitoring and analysis of network activities associated with different strains of ransomware, this strategic preparation provides important information that helps improve and validate the proposed ransomware detection model. The testbed is intentionally isolated to reduce potential risks and provide a secure environment for researching and modeling ransomware activity.

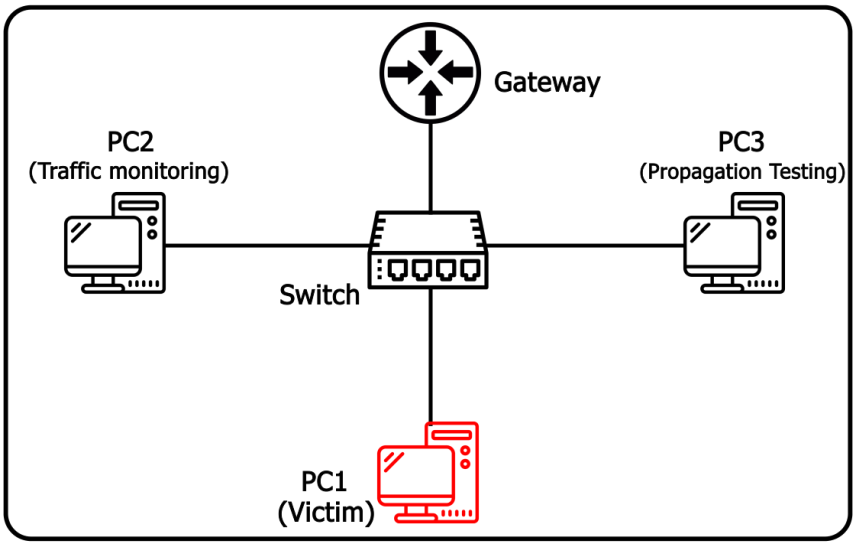


Figure 4.2 Testbed

As shown in Figure 4.2, the testbed consists of three personal computers (PCs) that are connected to the internet. This testbed's main goal is to gather ransomware traffic and keep track of network activities while assuring isolation from the main network by connecting this environment to a virtual local area network (VLAN). Each PC in this environment has a certain function.

Firstly, there is the Victim PC1, it was set up to perform the role of a ransomware victim. The ransomware program that has been purposefully placed on this computer has encrypted the files on it. This configuration enables a realistic evaluation of the possible impact and outcomes of such an assault by simulating a targeted user in a ransomware attack.

Secondly, Traffic Monitoring PC2 is equipped with Wireshark, a network protocol analyzer. It is connected to a mirror port on the switch, which mirrors all incoming and outgoing traffic. With this setup, PC2 can record and look at real-time network traffic, including any ransomware-related actions like file encryption or communication with command-and-

control servers. PC2's job is to keep track of and log the network activity related to the ransomware assault.

Lastly, Propagation Testing PC3 serves as a platform for studying the ransomware's propagation behavior within the network. This computer replicates an infected machine that actively tries to infect other machines with ransomware. We can learn a lot about ransomware's mechanics and possible effects on the network as a whole by studying how it behaves and how it spreads.

A managed switch that connects these PCs makes it easier for the three systems to communicate and offers some degree of network traffic control. To link the testbed to the outside world and provide access to resources like web servers, email servers, and other network services, a router device is also present. The internet is a potential source of ransomware attacks and other network dangers within this testbed; thus, it must be watched over and investigated.

4.4 Collecting Traffic

In an effort to thoroughly investigate the complexities of ransomware attacks and their network behaviors, we went on a multilayered strategy that made use of cutting-edge technologies and approaches. we methodically collected the network data generated by four unique ransomware families over 79 hours using Wireshark. This time frame allowed for a thorough examination of the ransomware's lifecycle, from infection through data exfiltration or encryption efforts.

Collecting network traffic data for ransomware faces a distinct set of difficulties, especially when searching for datasets covering a long period of time. A notable challenge is the lack

of large-scale datasets documenting ransomware activity over extended time frames like 79 hours in this case. Because ransomware attacks are often dynamic and irregular, it is difficult to identify datasets that capture the many strategies, methods, and practices used by bad actors over long periods of time. Furthermore, the dynamic nature of ransomware variations introduces an additional degree of complexity, as datasets must include a range of samples to ensure model flexibility. Due to their limited availability, these datasets require a difficult curation procedure that entails combining samples from many sources and carefully assessing the representativeness of each. Despite these difficulties, generating a broad and diverse set of ransomware traffic is essential for reliable training and validation of ransomware detection models, which in turn improves the effectiveness of cybersecurity measures by providing a more realistic representation of actual threats. Table 4.1 provides some statistics for each PCAP file that generated from the experiments

Table 4.1 PCAP files statistics

Name	Cerber	TeslaCrypt	CryptoWall	DarkSide
Download Date	11/8/2023	27/11/2023	1/12/2023	3/12/2023
Execution Date	21/8/2023	29/11/2023	2/12/2023	5/12/2023
PCAP Size	1.4 GB	1.3 GB	116 MB	200 MB
Client IP	192.168.50.2	192.168.1.222	192.168.1.222	192.168.1.222
Experiment Duration	24 hours	23 hours	12 hours	20 hours
Bytes	77237974	1338024179	115106367	18929957
Average Bytes/S	896	16	2880	261
Connected IP's	732	267	149	178

The process of gathering network traffic for many ransomware samples that are downloaded from three sources which are virusshare.com, bazaar.abuse.ch, and vx-underground.org requires a careful examination of the ever-changing field of dangerous software. During our project, 259 ransomware samples were examined, table 4.2 shows the number of total and running samples for each family. The variety of ransomware variations in this collection showed the dynamic nature of cybersecurity threats. However, since these attempts are unpredictable, 145 of the ransomware samples tested showed signs of operational success. The diversity highlights the difficulty of detecting ransomware as criminals continue to improve and change their tools. The study is made more difficult by inconsistency in the behavior of the sample, with some appearing inactive while others behave as expected. The nuances included in these findings provide important insights into the diversity in ransomware characteristics, which impacts the flexibility needed for any successful detection program.

Table 4.2 Number of ransomware samples

Ransomware	Number of Samples	Number of Running Samples
Cerber	77	41
TeslaCrypt	51	35
DarkSide	63	32
CryptoWall	53	37
Razy	27	0
Lockbit	51	0

4.5 Traffic Analysis

Before starting the process of making a data labelling, it is necessary to work on analyzing the traffic for each ransomware that was collected, and here lies the most challenging task,

as we need to do an analysis and examination of each PCAP file and using some protocols such as DNS, HTTP, TCP in order to know the behavioral pattern based on them, and then extract the connected ips and scan them through open-source tools and sites such as VirusTotal, abuseipdb, otx.alienvault and others to be sure that are related to malicious activates, and then we will be able to label the final dataset, we used Python with Scapy to extract some statistics for some protocols such as HTTP and TCP with focusing on RST and ACK flags.

Table 4.3 shows the percentage of protocols packets for each ransomware family, these protocols were examined in order to observe the behavior of each ransomware, and thus later know the ip addresses that were contacted during the ransomware's attempt to contact C&C server.

Table 4.3 Protocols percentages

	Cerber	TeslaCrypt	CryptoWall	DarkSide
HTTP	25.58%	45.3%	21.11%	14.81%
DNS	1.23%	0.15%	0.80%	5.99%
TCP-RST	0.27%	1.13%	1.55%	6.46%
TCP-ACK	18.67%	3.68%	73.01%	28.31%

4.5.1 HTTP Traffic

When targeting victims, viruses and ransomware often use HTTP POST as a key element of their attack tactics. The HTTP POST method is used in the context of cyber risks to transfer data from a user's browser to a web server, usually for form submission or sensitive data transfer such as windows credentials, and personal data. Using this communication channel,

malicious actors can carry out attacks by hiding their payloads inside HTTP POST requests that appear to be normal traffic. When it comes to viruses, this can lead to the spread of malicious code or directives aimed at compromising the system. On the contrary, ransomware may connect to command-and-control servers using the HTTP POST protocol, facilitating the exchange of encryption keys or ransom payment information. Attackers may disguise their operations by taking advantage of this basic feature of online communication, making it difficult for security measures to properly detect and neutralize risks. Understanding and monitoring how HTTP POST is used in network traffic is essential to protecting against these complex and ever-evolving cyber threats as cybersecurity continues to advance.

Attackers continually design covert communications links and robust command and control infrastructures to enable their malware to evade intrusion detection systems. Therefore, the attackers try to connect using common ports such as HTTP and HTTPS to their servers. According to the captured PCAP analysis as shown in Figure 4.3, a TCP connection is established with the files created by CryptoWall with a remote server, where the data is sent through HTTP protocol.

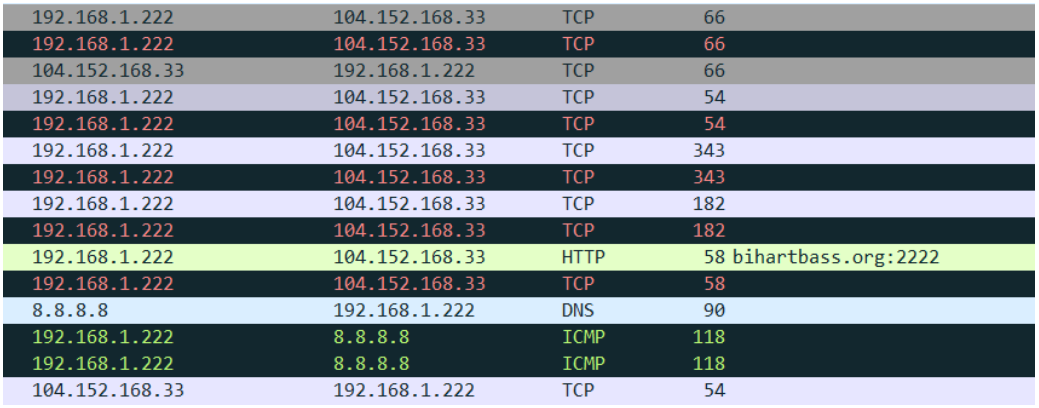


Figure 4.3 CryptoWall TCP connection

The next step after suspecting that the traffic that was monitored is suspicious is to work on examining this IP address or domain through the sites available online. As Figure 4.4 shows, the domain contained in the previous http post (bihartbass.org) was examined and found to be suspicious according to virustotal.

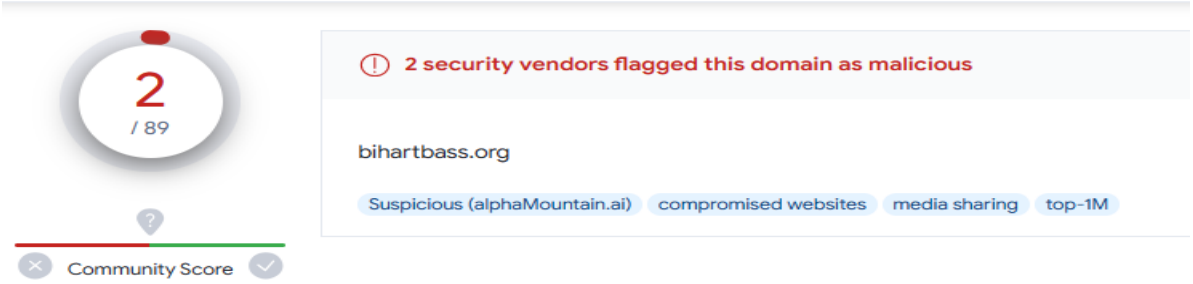


Figure 4.4 Virustotal result for bihartbass.org

The http is one of the indicators that can be verified in order to observe the behavior on the network after the ransomware is injected. However, despite its importance, it cannot be relied solely on, as not all ransomware families generate a large amount of http traffic, as shown in Figure 4.5, as very few suspicious http traffics in the case of both Cerber and CryptoWall samples.

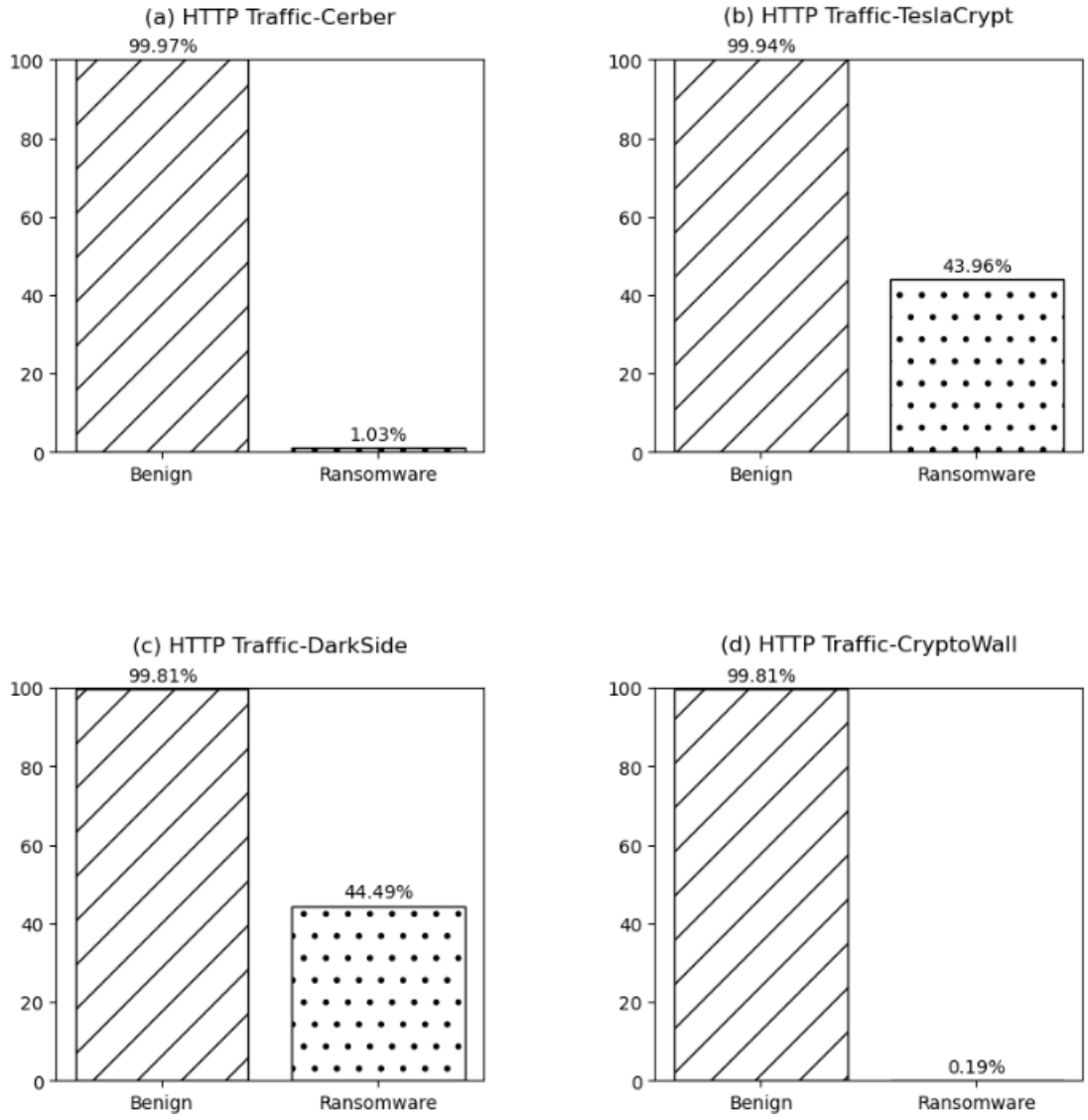


Figure 4.5 Benign and ransomware HTTP traffic

To conduct a more comprehensive investigation on each PCAP file and retrieve the largest possible number of addresses that were used by each ransomware sample, it is necessary to search through other protocols that may be used during the encryption process. Thus, the largest number of suspicious addresses will have been retrieved.

4.5.2 DNS Traffic

Cybercriminals frequently utilize DNS for a variety of objectives, including preventing the need to inject a list of servers IP addresses that are hard-coded since they can be quickly identified and blocked. Instead, DNS queries are continually made for pseudo-random domain names using DGA until the location of the actual server (Sgandurra et al., 2016). DNS can also be used to send data over secure network when no outward TCP access is permitted. Additionally, UDP is only able to support DNS queries, to send many DNS queries to a fictitious DNS server located at the attacker's end, the virus might split the file contents into numerous requests. According to other ransomware analyses, there are a lot of results that can be considered while analyzing the DNS traffic, such as DNS name error, meaningless domain name, DNS-time, DNS-qry-typ, and others.

Most of these things were determined to be normal in our investigation, particularly when we discovered that some of the domains to which these connections are made are not random. This presents a challenge for certain studies that use the Shannon Entropy metric to determine how random the domain name distribution is. Shannon entropy is a concept from information theory that measures the uncertainty or average amount of information produced by a random variable (May et al., 1994). It's often used in machine learning and data analysis to quantify the unpredictability or randomness of data. The equation for Shannon entropy (H(X)) of a discrete random variable (X) with probability mass function (P(X)) is:

$$H(X) = -\sum_(x \in X)\, P(x)\, log_2(P(x)) \quad \text{..... (May et al., 1994)}$$

Here's a breakdown of the terms in the equation:

- X: The set of possible outcomes of the random variable.

- P(x): The probability of outcome (x).

To calculate Shannon entropy, you iterate over all possible outcomes (x) in (X), multiply the probability of (x) occurring by its logarithm base 2, and sum up these values with a negative sign. In applications like data analysis, higher entropy values indicate greater uncertainty or randomness in the data, while lower entropy values suggest more predictable or structured data.

Only the first label of each domain name was used for the entropy calculation, as the second label - which may include .org, .edu, .com, .net, and so on -typically indicates the root DNS servers. For example, the domain "dnfyupcnwghbqao.be" has an entropy of 3.7736, a maximum of 8 sequential consonants, and 2 sequential vowels. Similarly, "qkbwinwgdvbufu.de" has an entropy of 3.3788, with 6 sequential consonants and 1 sequential vowel(Almashhadani et al., 2019). These high entropy values suggest that these domain names are complex and likely malicious. Some of domain names for each ransomware are displayed in Table 4.4, and as we noted in the table, there are some domains that are not complex or random, and therefore they can be considered normal according to Shannon entropy, even though they are due to the activity of ransomware.

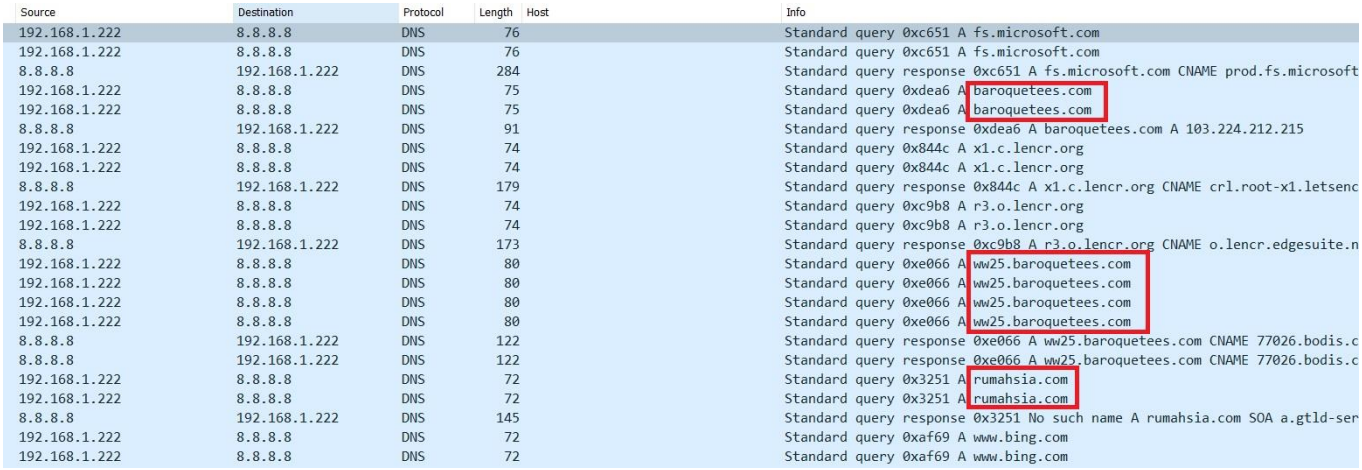
Table 4.4 Connected domains

Ransomware	**Domains**
Cerber	- bqyjebfh25oellur.onion.to - ip-api.com - dns.msftncsi.com - api.blockcypher.com - chain.co
TeslaCrypt	- myexternalip.com - homeopathischdierenarts.nl

	- crown.essaudio.pl - graysonacademy.com - e-slubneobraczki.pl - gjesdalbrass.no - todayinbermuda.co - mosaudit.com - polyhedrusgroup.com - survey-smiles.com - lutheranph.com - hongsi.com - dustinhansenbook.com - onegiantstore.com - www.informaticauno.net - clientconfig.passport.net
CryptoWall	- ip-addr.es - 173.230.130.167:8080 - bihartbass.org:2222 - clientconfig.passport.net - gdspirit.com - imanaging.info - cranburylibraryfoundation.org - mawum.com - japaneselink.net - kmreich.com - www.kmr-group.de - status.geotrust.com - everestmarketinggroup.com - sissolarventures.com - easbrain.com - karateserbia.org - canbroc-bg.com - bettercatch.com - skprints.com - jump-and-reach.com - superhamali.com
DarkSide	- ww25.baroquetees.com - bihartbass.org:2222 - rumahsia.com

Although we do not attempt to detect ransomware traffic by relying on DNS in this research, it has become clear that it is not possible to rely on it individually, whether by identifying suspicious domains or even by using Shannon Entropy.

As in the case of HTTP, verifying the domains and knowing the normal traffic from the abnormal is essential for the labeling stage. Figure 4.6 show some connected domains in DarkSide ransomware traffic, even if this feature is not exploited during the labeling process, it can be utilized during the process of tracking traffic and following up on the ip addresses that follow the process of connecting to the domain.



Source	Destination	Protocol	Length	Host	Info
192.168.1.222	8.8.8.8	DNS	76		Standard query 0xc651 A fs.microsoft.com
192.168.1.222	8.8.8.8	DNS	76		Standard query 0xc651 A fs.microsoft.com
8.8.8.8	192.168.1.222	DNS	284		Standard query response 0xc651 A fs.microsoft.com CNAME prod.fs.microsoft
192.168.1.222	8.8.8.8	DNS	75		Standard query 0xdea6 A baroquetees.com
192.168.1.222	8.8.8.8	DNS	75		Standard query 0xdea6 A baroquetees.com
8.8.8.8	192.168.1.222	DNS	91		Standard query response 0xdea6 A baroquetees.com A 103.224.212.215
192.168.1.222	8.8.8.8	DNS	74		Standard query 0x844c A x1.c.lencr.org
192.168.1.222	8.8.8.8	DNS	74		Standard query 0x844c A x1.c.lencr.org
8.8.8.8	192.168.1.222	DNS	179		Standard query response 0x844c A x1.c.lencr.org CNAME crl.root-x1.letsenc
192.168.1.222	8.8.8.8	DNS	74		Standard query 0xc9b8 A r3.o.lencr.org
192.168.1.222	8.8.8.8	DNS	74		Standard query 0xc9b8 A r3.o.lencr.org
8.8.8.8	192.168.1.222	DNS	173		Standard query response 0xc9b8 A r3.o.lencr.org CNAME o.lencr.edgesuite.n
192.168.1.222	8.8.8.8	DNS	80		Standard query 0xe066 A ww25.baroquetees.com
192.168.1.222	8.8.8.8	DNS	80		Standard query 0xe066 A ww25.baroquetees.com
192.168.1.222	8.8.8.8	DNS	80		Standard query 0xe066 A ww25.baroquetees.com
192.168.1.222	8.8.8.8	DNS	80		Standard query 0xe066 A ww25.baroquetees.com
8.8.8.8	192.168.1.222	DNS	122		Standard query response 0xe066 A ww25.baroquetees.com CNAME 77026.bodis.c
8.8.8.8	192.168.1.222	DNS	122		Standard query response 0xe066 A ww25.baroquetees.com CNAME 77026.bodis.c
192.168.1.222	8.8.8.8	DNS	72		Standard query 0x3251 A rumahsia.com
192.168.1.222	8.8.8.8	DNS	72		Standard query 0x3251 A rumahsia.com
8.8.8.8	192.168.1.222	DNS	145		Standard query response 0x3251 No such name A rumahsia.com SOA a.gtld-ser
192.168.1.222	8.8.8.8	DNS	72		Standard query 0xaf69 A www.bing.com
192.168.1.222	8.8.8.8	DNS	72		Standard query 0xaf69 A www.bing.com

Figure 4.6 DarkSide Connected domains

DNS gives attackers a way to operate covertly. Ransomware may blend in with regular network traffic by using these widely used protocols, which makes it more difficult for security solutions to recognize and stop criminal activity. An extra degree of evasion is added by the dynamic nature of DNS, particularly when paired with methods like DGAs.

4.5.3 TCP flags

In the context of cyberattacks, ransomware and viruses often exploit the RST (reset) flag of the TCP protocol as a tactical element in their attack plans. A control bit in the TCP header

called the RST flag indicates immediate termination of the current connection. Malicious creators may use the RST tag to manipulate or alter network connections when distributing viruses or ransomware. The malware has the ability to generate RST packets during the attack in order to terminate communications that made between the victim's computer and servers. This strategy aims to attack services, disrupt network connections, and cause chaos in the targeted area. Furthermore, malicious traffic may be generated by attackers who use RST signals to hide their activity, making it difficult for security systems to distinguish between malicious and normal network activity. The integrity and availability of networked systems depends on defenders identifying and minimizing the impacts of RST-based attacks as their cybersecurity defenses advance. Figure 4.7 shows a lot of malicious RST connections which makes us explore many of the ip addresses that the ransomware uses to access its servers, and thus gave us the ability to identify and label them to training the model on them.

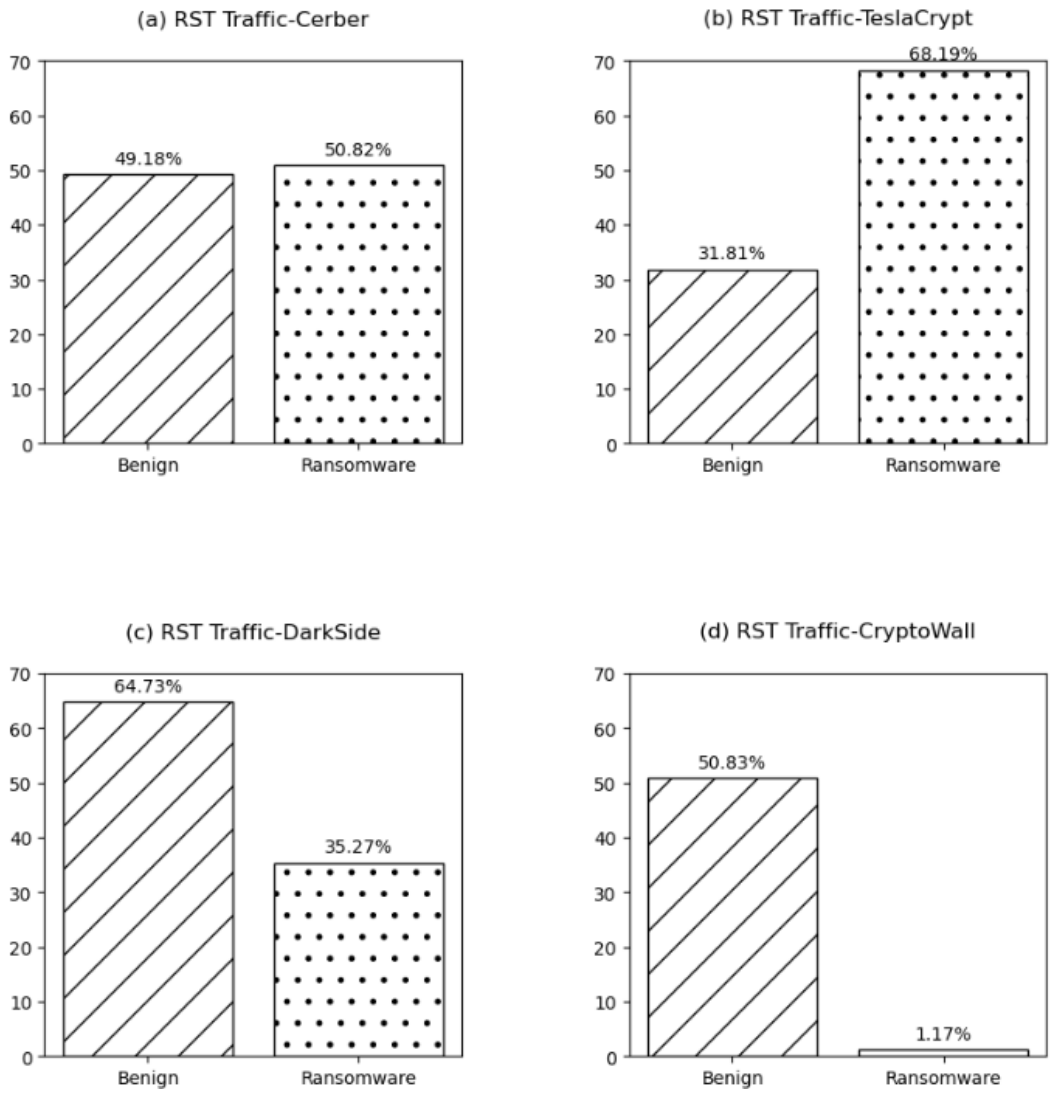


Figure 4.7 Benign and ransomware RST traffic

AS the previous Flags, ransomware employ the TCP ACK (Acknowledgment) flag in their assault techniques. In the Transmission Control Protocol, the ACK flag is used to indicate that data has been received (TCP). This flag is used by malicious actors to control network traffic and create secret communication routes. ACK packets with certain values can be used by viruses and ransomware in some assaults to conceal their activity within what appear to be genuine transactions. Attackers can insert malicious payloads into the communication

stream while making regular acknowledgments appear to be occurring by carefully manipulating the acknowledgment sequence numbers. By using this tactic, the virus can avoid detection by security systems and covertly communicate commands or updates from command-and-control servers.

As the other investigated flags, As shown in Figure 4.8, the Acknowledgment Flag appears significantly in the ransomware's traffic as it also appears greatly in benign traffic, but this is also an indication of the possibility of extracting addresses, which is the goal of this process.

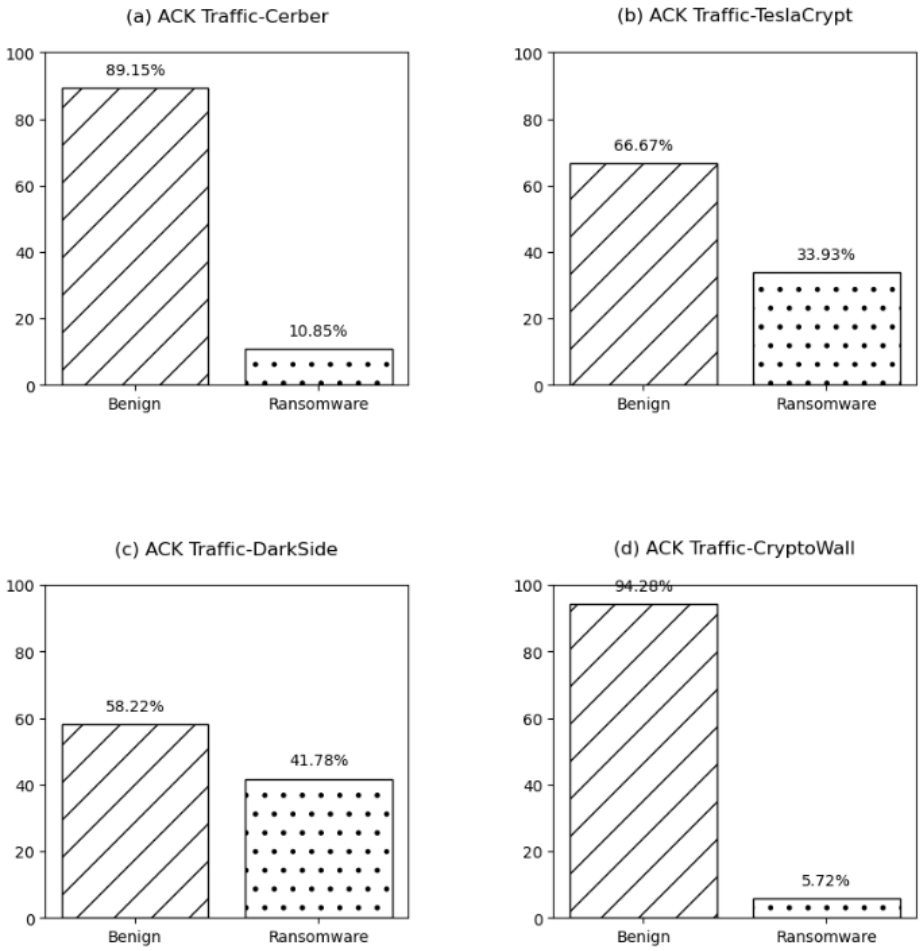


Figure 4.8 Benign and ransomware ACK traffic

Also shown in Figure 4.9 are several flag packets that appeared in the TeslaCrypt ransomware traffic which belong to a suspicious ip address.

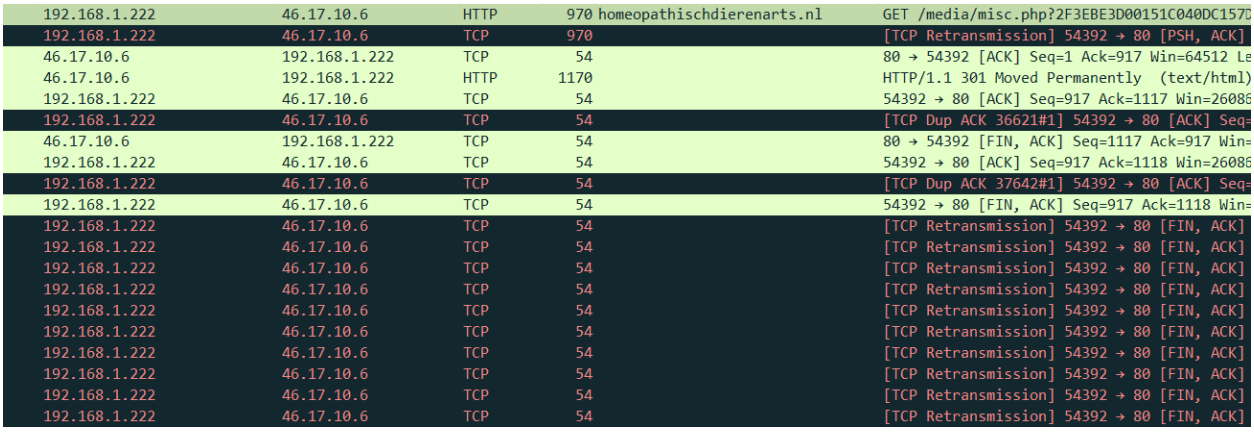


Figure 4.9 TeslaCrypt traffic

4.6 Dataset

Once we collected the initial network traffic data, we moved on to the next critical phase: refining and deriving meaningful insights. CICFlowmeter, famous for its prowess in converting raw packet data into actionable flow statistics with 84 features (cicflowmeter · PyPI, 2023). By processing traffic captured by CICFlowmeter, we gained an enhanced understanding of the traffic patterns, data volumes, and communication protocols used by ransomware traffic. Table 4.5 shows all features that extracted from the PCAP files using CICFlowmeter.

Table 4.5 Dataset Features

Attribute No.	Feature	Attribute No.	Feature	Attribute No.	Feature
1	Flow ID	29	Fwd IAT Std	58	ECE Flag Cnt
2	Src IP	30	Fwd IAT Max	59	Down/Up Ratio
3	Src Port	31	Fwd IAT Min	60	Pkt Size Avg

4	Dst IP	32	Bwd IAT Tot	61	Fwd Seg Size Avg
5	Dst Port	33	Bwd IAT Mean	62	Bwd Seg Size Avg
6	Protocol	35	Bwd IAT Std	63	Fwd Byts/b Avg
7	Timestamp	36	Bwd IAT Max	64	Fwd Pkts/b Avg
8	Flow Duration	37	Bwd IAT Min	65	Fwd Blk Rate Avg
9	Tot Fwd Pkts	38	Fwd PSH Flags	66	Bwd Byts/b Avg
10	Tot Bwd Pkts	39	Bwd PSH Flags	67	Bwd Pkts/b Avg
11	TotLen Fwd Pkts	40	Fwd URG Flags	68	Bwd Blk Rate Avg
12	TotLen Bwd Pkts	41	Bwd URG Flags	69	Subflow Fwd Pkts
13	Fwd Pkt Len Max	42	Fwd Header Len	70	Subflow Fwd Byts
14	Fwd Pkt Len Min	43	Bwd Header Len	71	Subflow Bwd Pkts
15	Fwd Pkt Len Mean	44	Fwd Pkts/s	72	Subflow Bwd Byts
16	Fwd Pkt Len Std	45	Bwd Pkts/s	73	Init Fwd Win Byts
17	Bwd Pkt Len Max	46	Pkt Len Min	74	Init Bwd Win Byts
18	Bwd Pkt Len Min	47	Pkt Len Max	75	Fwd Act Data Pkts
19	Bwd Pkt Len Mean	48	Pkt Len Mean	76	Fwd Seg Size Min
20	Bwd Pkt Len Std	49	Pkt Len Std	77	Active Mean

21	Flow Byts/s	50	Pkt Len Var	78	Active Std
22	Flow Pkts/s	51	FIN Flag Cnt	79	Active Max
23	Flow IAT Mean	52	SYN Flag Cnt	80	Active Min
24	Flow IAT Std	53	RST Flag Cnt	81	Idle Mean
25	Flow IAT Max	54	PSH Flag Cnt	82	Idle Std
26	Flow IAT Min	55	ACK Flag Cnt	83	Idle Max
27	Fwd IAT Tot	56	URG Flag Cnt	84	Idle Min
28	Fwd IAT Mean	57	CWE Flag Count		

However, the real challenge arose during the data classification process, which is a pivotal step in attributing malicious intent to specific network activities. While traditional classification often relies on the stability of IP addresses, ransomware often uses disposable, transient infrastructure, making it difficult to maintain a reliable list of malicious IP addresses. However, we used a combination of techniques, including cross-referencing threat intelligence feeds and analyzing patterns of suspicious behavior, to identify and document malicious IP addresses that demonstrated connections.

By leveraging the advantage of precise knowledge of injection time into victim device, a strategic approach is adopted to investigate malicious traffic with high accuracy. By aligning the captured network data with the precise moments of ransomware injection, focused analysis was performed. This enabled the isolation of anomalous patterns and encrypted communications, highlighting the complex mechanisms used by ransomware strains. This precise timing alignment greatly enhanced the depth of our analysis, contributing to a more precise understanding of ransomware spread and data leakage tactics.

Table 4.6 provides essential information about the four ransomware families, including the ransomware type, the time at which the captured network traffic started, and the moment at which the ransomware was injected into the targeted system. This concise arrangement of data offers valuable insights into the timing and characteristics of these cyber threats, facilitating a better understanding of their modus operandi and aiding in the development of effective defense strategies.

Table 4.6 Experiment times

Ransomware	**Start Traffic**	**Inject Ransomware**
Cerber	21-8-2023 at 14:07	21-8-2023 at 14:21
TeslaCrypt	29-11-2023 at 18:44	29-11-2023 at 18:52
CryptoWall	2-12-2023 at 1:11	2-12-2023 at 1:20
DarkSide	5-12-2023 at 15:51	5-12-2023 at 15:59

It is important to highlight that labeling is difficult due to the dynamic nature of ransomware attacks. We have ensured that the tagging procedures remain robust and flexible when IP addresses are replaced due to their temporary nature. Although time-consuming, this iterative technique ultimately produced a dataset that was highly annotated and provided in-depth insights into the behavior and strategies of the four different ransomware families.

In essence, this project involved the careful orchestration of data collection, amplification, and labeling, which called for a combination of technological aptitude and methodological flexibility. The collected information serves as a basis for improving cybersecurity tactics and bolstering our online defenses against new threats in addition to offering an understanding of ransomware activity.

To be used in the multi-classification analysis of ransomware types, each instance of the obtained traffic received a strict classification throughout the classification step. The results reveal the distribution of cases for each ransomware variant and are presented in the table 4.7. The numbers indicate the number of cases classified with “1” to indicate the presence of ransomware, or “0” to indicate its normal. Cerber, TeslaCrypt, CryptoWall, and DarkSide are the individual ransomware variants specifically analyzed. Each has a different number of events that are classified as ransomware ('1') or not ('0'). The 37,751 labeled dataset provides a comprehensive basis for training and evaluating a multi-classification model to recognize and classify different types of ransomware attacks.

Table 4.7 Labeling Count

Ransomware	**1: Ransome**	**0: Normal**
Cerber	665	4045
TeslaCrypt	489	3129
CryptoWall	8173	4195
DarkSide	9733	7322
	19,060	18,691

After the binary classification stage, in which cases were classified as '1' or '0', the multi-classification phase started for cases with the label '1'. In order to better distinguish and categorize the particular version, a unique class label was given to each form of ransomware in the next step. The labels assigned to the Cerber instances were '1', TeslaCrypt as '2', CryptoWall as '3', and DarkSide as '4'. This methodical labeling guarantees that the multi-classification model is trained to detect and classify the particular type of ransomware that is present in the network data, in addition to differentiating between ransomware and non-

ransomware cases. The use of these improved labels improves the model's capability to correctly categorize and react to various ransomware attacks, providing a more understanding of the nature of potential attacks.

4.7 Preprocessing

Before diving into the analysis of the ransomware traffic dataset, it is essential to perform comprehensive preprocessing activities in order to build a strong model. This dataset, which includes dimensions of 37751 and 84, requires careful attention to make sure it is prepared for efficient machine learning exploration. Strategic feature curation, handling zero and missing values, managing infinite values, standardizing numerical ranges, and encoding categorical variables are important preprocessing tasks. In addition, the dataset's vulnerability to outliers demands careful thought, which leads to the implementation of techniques designed to handle extreme numbers. The main objective of this preprocessing method is to improve the dataset's applicability for machine learning applications and reduce any potential biases. These steps enable the next stages of analysis to gather significant insights from the complicated environment of ransomware traffic, in addition to providing a basis for precise model training. Here we discuss some of the tasks through which the dataset was processed:

- Checking missing values

Missing values in a dataset of ransomware network traffic can arise from various sources. These include data capture issues during collection, such as network interruptions or incomplete recordings, as well as intentional anonymization processes for privacy. Filtering or sampling methods may also exclude certain observations, leading to missing values. Additionally, errors in data processing or instrumentation issues can contribute to data gaps.

Natural variability in network traffic patterns may further contribute to missing values. Understanding these sources is crucial for implementing effective preprocessing techniques, such as data imputation or exclusion strategies, to ensure the integrity and reliability of subsequent analyses.

Examining the dataset for missing values is an important part of the preprocessing process that makes sure the dataset is complete and undamaged for the analyses that follow. Upon closer inspection, we discovered as shown in figure 4.10 that there was a total of 183 missing values in the 'Flow Byts/s' column.

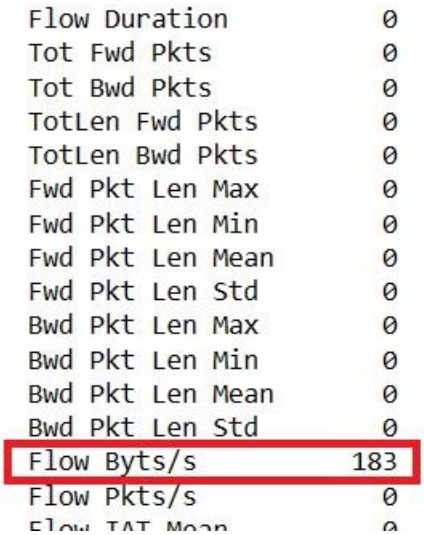


Figure 4.10 Checking missing values

As a result, a fix was required to ensure the dataset stayed undamaged. Figure 4.11 shows how the mean of the 'Flow Byts/s' column was used to impute the missing data in order to close the gap. This was a wise option.

```
mean_value = data['Flow Byts/s'].mean()
data['Flow Byts/s'].fillna(mean_value, inplace=True)
```

Figure 4.11 Use mean of Flow Byts column

Figure 4.11 show the filling of missing values on column Flow Byts/s with the mean of column, this method not only fixes the missing data but also adds a central tendency measure that fits the feature's overall distribution. We enhance the dataset for machine learning model training by methodically resolving missing information, creating an environment that supports precise and trustworthy insights regarding ransomware traffic habits.

- Drop some feature

During the feature selection process, many features were intentionally excluded. These attributes are often relevant to network analysis, but not relevant to the complex task of ransomware detection, Figure 4.12 shows the features that dropped from the dataset in the first stage. Although the Flow ID is intrinsically distinct for each network flow, it is usually more easily identifiable than educational, and provides no benefit in distinguishing between ransomware actions. Ransomware identification is also complicated by the 'Source port' and 'Destination port,' in addition to the 'Source IP' and 'Destination IP' addresses. The flexible nature of ransomware operations is well-known, since they often switch up their Command and Control (C2C) servers. since a result, IP addresses are not always a trustworthy indicator of malicious activity. Moreover, source and destination ports in ransomware communications are dynamic, which makes them less indication of malicious intent. Furthermore, although the 'Protocol' function is fundamental to network research in general, it has little discriminating ability when it comes to ransomware detection. Because ransomware

variations frequently conceal their communication patterns inside widely used protocols, this characteristic is less useful for separating malicious traffic. Eliminating these characteristics causes the emphasis to change to more behavioral and intrinsic signs, providing a stronger basis for the precise detection of ransomware activity in the always changing cybersecurity environment.

```
FinalDataset.drop(['Timestamp','Flow ID','Src IP', 'Src Port', 'Dst IP', 'Dst Port', 'Protocol'
          ], axis=1, inplace=True)
```

Figure 4.12 Dropped features

- Infinite values

Infinite values appearing in a dataset of ransomware network traffic can stem from several underlying factors. These factors include anomalies or outliers in the data that result in extreme values, such as exceptionally large packet sizes or durations that exceed typical ranges. These anomalies may arise from network disruptions, misconfigurations, or unusual network activity that leads to data points with values beyond the normal scale. Additionally, computational issues during data processing or analysis, such as division by zero or mathematical operations involving infinite or undefined values, can also generate infinite values in the dataset. Understanding these potential sources of infinite values is essential for implementing robust data validation and cleaning procedures to handle such anomalies effectively and ensure the accuracy and reliability of subsequent analyses.

In order to guarantee a consistent and meaningful representation of the data, it is important to address infinite values inside the data set. To address this, a methodical approach was put into place. First, every occurrence of an infinite value in every column was found and replaced with NaN (Not a Number) in a uniform manner. The purpose of this first step is to standardize the representation of infinite values so that the dataset can be ready for further processing. After that, each column's mean was calculated to provide a representative central tendency measure. Because mean can fill in missing or infinite values while preserving the statistical properties of the data, it is the measure of choice for imputation. The computed mean unique to each column was then used to fill in any NaN values that may have existed. This comprehensive method guarantees the reduction of infinite values, enhancing the dataset's stability and preparing it for the complicated requirements of machine learning model training and analysis.

```
import pandas as pd
import numpy as np

data.replace([np.inf, -np.inf], np.nan, inplace=True)
mean_values = data.mean()
data.fillna(mean_values, inplace=True)
```

Figure 4.13 replacing infinite values with mean value

- Correlated features

One of the most important steps in tuning the dataset for the best machine learning model performance is identifying and controlling correlated features. In our investigation, a careful review of feature correlations revealed substantial correlations between particular features. In order to prevent overlap and optimize the dataset, features with a high degree of correlation

were carefully eliminated. This procedure improves the overall quality of the dataset by reducing redundancy and helping to remove noise. The selection criteria for features were based on maintaining the depth of data while removing needless repeats. The initial feature count was successfully down to a more manageable set of 32 features. In addition to modifying the dataset for effective model training and ensuing analysis centered on ransomware traffic detection, this careful curation guarantees that the dataset maintains its inherent patterns and critical information.

Figure 4.15 shows the correlation heatmap of the features before removing the high correlated features from the dataset.

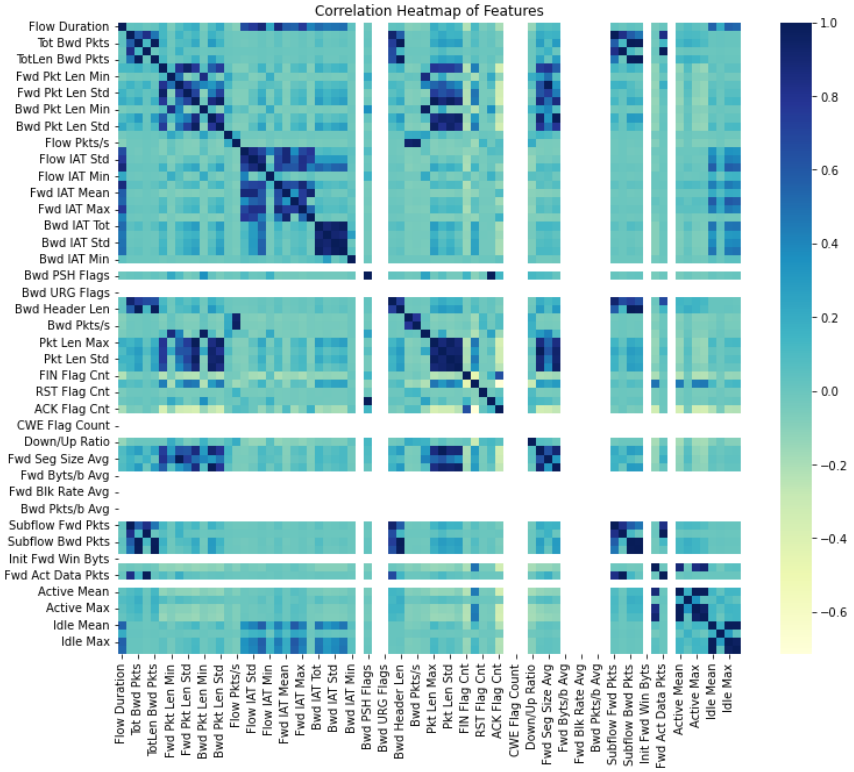


Figure 4.14 Correlation Heatmap

In machine learning and data analysis, understanding the correlations between features is crucial for interpreting the relationships within a dataset. Correlated feature pairs highlight variables that change together, offering insights into how certain aspects of the data are interconnected. Exploring these correlations helps in identifying redundant or highly influential features, which is fundamental for building accurate predictive models and gaining deeper insights into the underlying data pattern. Figure 4.15 shows the correlated features with a 0.9 threshold.

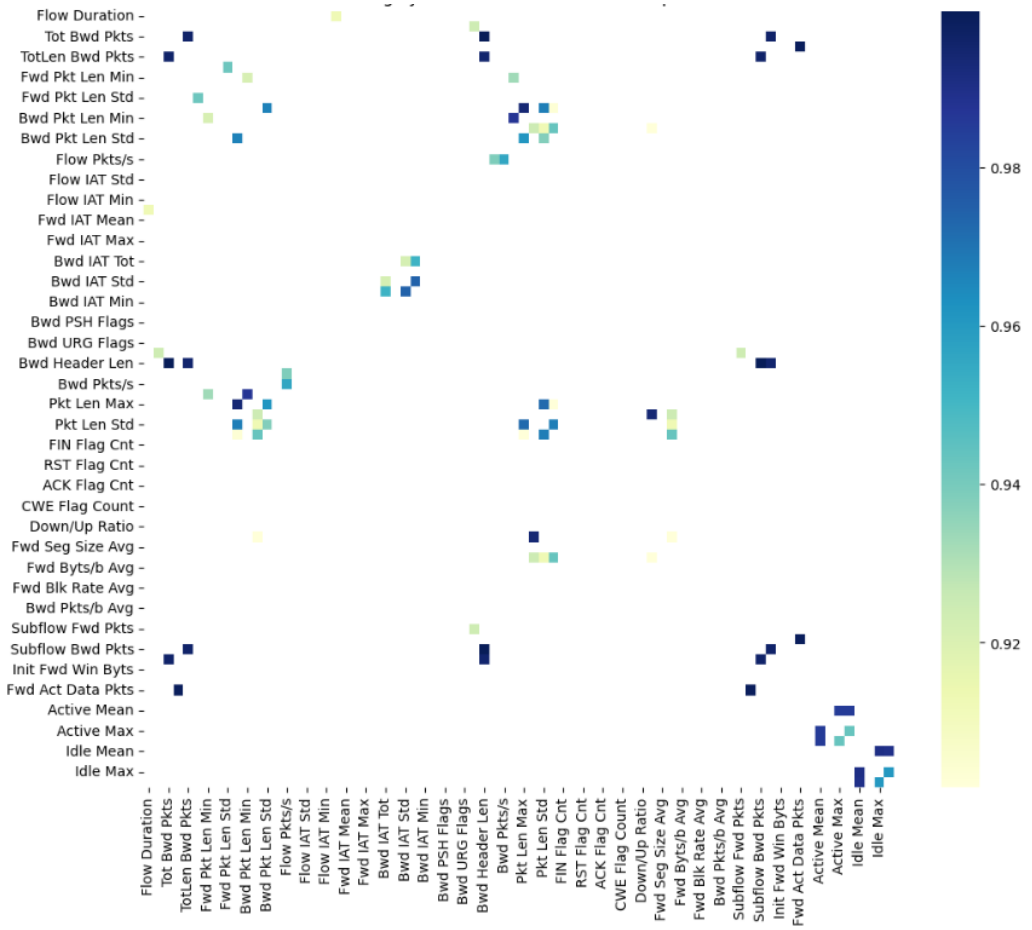


Figure 4.15 highly correlated features

Based on the above heatmap, we can delve into why these features exhibit correlation. Here are the explanations for each correlated feature pair:

(Fwd Pkt Len Min, Pkt Len Min): These features are highly correlated because the minimum forward packet length directly influences the overall minimum packet length within the flow. When the smallest forward packet increases or decreases, it directly affects the smallest packet length observed in the entire flow, resulting in a strong positive correlation.

(Pkt Len Max, Pkt Len Std): The maximum packet length and the standard deviation of packet lengths are highly correlated because larger packet sizes tend to contribute more significantly to the overall variability in packet sizes. Therefore, as the maximum packet length increases or decreases, it leads to a corresponding change in the variability (standard deviation) of packet lengths.

(Flow Pkts/s, Bwd Pkts/s): These features exhibit high correlation because the flow packets per second and backward packets per second are interconnected in a flow. Variations in the rate of packets moving in one direction (forward or backward) often coincide with similar variations in the opposite direction, resulting in a strong positive correlation between these rates.

(Fwd IAT Tot, Flow Duration): The total forward inter-arrival time and flow duration are highly correlated because the cumulative time between consecutive forward packets (inter-arrival time) directly contributes to the overall duration of the flow. As the total time between

forward packets increases or decreases, it directly impacts the total duration of the flow, leading to a strong positive correlation.

(Bwd IAT Std, Bwd IAT Tot): The standard deviation of backward inter-arrival times and the total backward inter-arrival time exhibit high correlation because the variability (standard deviation) in the time intervals between consecutive backward packets reflects the overall duration of backward packet arrivals. Changes in the variability of backward inter-arrival times often correspond to changes in the total time taken for backward packets to arrive, resulting in a strong positive correlation.

(Bwd IAT Max, Bwd IAT Std): The maximum backward inter-arrival time and the standard deviation of backward inter-arrival times are highly correlated because longer intervals between consecutive backward packets (maximum inter-arrival time) contribute significantly to the variability (standard deviation) in backward inter-arrival times. Variations in the longest intervals often coincide with variations in the overall variability of backward packet arrivals, leading to a strong positive correlation.

(Subflow Fwd Pkts, Fwd Header Len): The number of subflow forward packets and the forward header length exhibit high correlation because the structure and content of forward packets within subflows directly influence the size of their headers. As the number of forward packets in subflows increases or decreases, it directly impacts the cumulative size of their headers, resulting in a strong positive correlation.

(Bwd Header Len, Subflow Bwd Byts): The backward header length and the number of subflow backward bytes are highly correlated because the size of backward packet headers

directly affects the total length of data carried by subflow backward packets. Changes in the size of backward headers often coincide with changes in the total data volume of subflow backward packets, leading to a strong positive correlation.

(Bwd Pkt Len Std, Pkt Len Max): The standard deviation of backward packet lengths and the maximum packet length are highly correlated because the variability (standard deviation) in backward packet sizes tends to increase with larger packet sizes. Therefore, as the maximum packet length increases or decreases, it leads to a corresponding change in the variability of backward packet sizes, resulting in a strong positive correlation.

(Pkt Len Var, Bwd Pkt Len Std): The variance of packet lengths and the standard deviation of backward packet lengths exhibit high correlation because the dispersion (variance) in packet sizes reflects the overall variability (standard deviation) of backward packet sizes. Changes in the dispersion of packet lengths often coincide with changes in the variability of backward packet sizes, leading to a strong positive correlation.

(Bwd Pkt Len Mean, Pkt Len Var): The mean of backward packet lengths and the variance of packet lengths are highly correlated because the average size of backward packets influences the dispersion (variance) in packet sizes. Variations in the average backward packet size often correspond to variations in the dispersion of packet lengths, resulting in a strong positive correlation.

(Pkt Size Avg, Pkt Len Mean): The average packet size and the mean packet length exhibit high correlation because the overall average size of packets directly influences the mean

length of packets. Changes in the average packet size often coincide with changes in the mean packet length, leading to a strong positive correlation.

(Bwd Seg Size Avg, Pkt Len Std): The average segment size and the standard deviation of packet lengths are highly correlated because the average size of packet segments reflects the overall variability (standard deviation) in packet sizes. Variations in the average segment size often correspond to variations in the variability of packet sizes, resulting in a strong positive correlation.

(Subflow Fwd Byts, Fwd Act Data Pkts): The number of forward bytes in subflows and the number of forward packets containing actual data exhibit high correlation because the volume of data within subflow forward packets directly relates to the number of packets carrying actual data. Changes in the volume of data in subflows often coincide with changes in the number of packets containing actual data, leading to a strong positive correlation.

(Subflow Bwd Pkts, TotLen Bwd Pkts): The number of subflow backward packets and the total length of backward packets exhibit high correlation because the quantity of backward packets in subflows directly influences the total length of data in backward packets. Changes in the number of subflow backward packets often correspond to changes in the total length of backward packets, leading to a strong positive correlation.

(Active Max, Active Min): The maximum active time and the minimum active time are highly correlated because the longest and shortest durations of active states directly influence each other. As the maximum active time increases or decreases, it leads to a corresponding change in the minimum active time, resulting in a strong positive correlation.

(Idle Max, Idle Min): The maximum idle time and the minimum idle time exhibit high correlation because the longest and shortest durations of idle states directly influence each other. Changes in the longest idle time often coincide with changes in the shortest idle time, leading to a strong positive correlation.

- **Features importance**

Feature importance serves as a pivotal metric in characterizing the contribution of each feature to the predictive power of the machine learning model. By leveraging the Random Forest Classifier (rfc.feature_importances_), we extracted importance scores for each feature in our ransomware traffic dataset. This technique evaluates the importance of each feature by measuring its impact on the model's accuracy. It is worth noting that reducing the number of features is a strategic step towards model efficiency and interpretability. By selecting the top 5 features based on their importance scores, we distill the essence of the dataset, focusing on the most influential features that drive accurate ransomware traffic detection. A visual representation of the top 12 feature scores, shown in figure 4.16, provides a brief overview of the relative importance of each feature. This simplified approach not only enhances the computational efficiency of the model, but also facilitates a clearer understanding of the underlying patterns, simplifying the path to effective ransomware detection.

Init Bwd Win Byts	0.103523
Bwd Header Len	0.062688
Flow IAT Mean	0.045456
Flow Pkts/s	0.044874
Bwd Pkts/s	0.042218
Flow IAT Max	0.041591
Flow Duration	0.040857
Flow IAT Std	0.030138
Fwd Header Len	0.027717
Fwd Pkts/s	0.026119
Flow IAT Min	0.025929
Fwd IAT Mean	0.025436

Figure 4.16 The score of each feature

Here are clear definitions for the top features:

Init BWD Win Byts: The initial backward window size in bytes, representing the size of the TCP window in the backward direction at the start of the flow.

Bwd Header Len: The total length of headers in the backward direction, including TCP, IP, and other protocol headers.

Flow IAT Mean: The mean inter-arrival time of flow packets, calculated as the average time between two consecutive packets belonging to the same flow.

Flow Pkts/S: The flow packet rate, indicating the number of packets per second in the flow.

Bwd Pkts/S: The backward packet rate, representing the number of packets per second in the backward direction of the flow.

Flow IAT Max: The maximum inter-arrival time of flow packets, indicating the longest time interval between two consecutive packets in the flow.

Flow Duration: The duration of the flow from start to end, measured in seconds.

Flow IAT Std: The standard deviation of inter-arrival times of flow packets, providing a measure of the variation in packet arrival times within the flow.

Fwd Header Len: The total length of headers in the forward direction, including TCP, IP, and other protocol headers.

Fwd Pkts/S: The forward packet rate, indicating the number of packets per second in the forward direction of the flow.

Flow IAT Min: The minimum inter-arrival time of flow packets, indicating the shortest time interval between two consecutive packets in the flow.

Fwd IAT Mean: The mean inter-arrival time of forward packets, calculated as the average time between two consecutive packets in the forward direction of the flow.

4.8 Performance Measurement

At the final end of the thesis methodology, evaluation of machine learning techniques becomes crucial, and several measures are operationalized to evaluate performance. Accuracy, precision, recall, F1 score, and execution time are among the chosen criteria. These metrics were used in the dataset generated in the testing phase, focusing on three expected outcomes.

The performance evaluation metrics are contingent upon the following expected results:

True Positive (TP): Accurate detection of ransomware attacks(Abdulrazaq and Al-Morjan, 2010).

True Negative (TN): Correct recognition of normal records(Abdulrazaq and Al-Morjan, 2010).

False Positives (FP): Incorrect detection of ransomware attack(Abdulrazaq and Al-Morjan, 2010).

False Negatives (FN): Incorrect recognition of normal traffic(Abdulrazaq and Al-Morjan, 2010).

Execution Time: The duration taken for the machine learning methods to process and deliver results.

Using an in-depth analysis of these results, the effectiveness of machine learning techniques can be evaluated, providing insight into their recall, precision, accuracy, F1 score and efficiency when it comes to detecting ransomware and identifying regular traffic and the corresponding execution time.

Accuracy: A metric measuring how well predictions align with actual outcomes(Abdulrazaq and Al-Morjan, 2010).

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

Precision: measures the accuracy of identifying ransomware attacks among all real detections, providing insight into the system's ability to differentiate one type of ransomware attack from other ransomware attacks (Abdulrazaq and Al-Morjan, 2010).

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

Recall: Recall is calculated by dividing the number of predicted attacks by the actual attacks, indicating the system's effectiveness in identifying true ransomware attacks. Also known as the "true positive rate"(Abdulrazaq and Al-Morjan, 2010) .

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

F1 Score: A balanced metric between precision and recall, ranging from 0 to 1.

$$F1\ score = \frac{2TP}{2TP+FP+FN}$$

In conclusion, Chapter 4 represents the peak of our research, revealing an innovative model ready to change how we detect and predict ransomware. This model, carefully designed and thoroughly tested in a controlled environment, shows our dedication to improving cybersecurity resilience.

Through the lens of traffic analysis, dataset construction, preprocessing methodologies, and performance measurement metrics, we have meticulously engineered a solution that transcends traditional approaches. Our model's ability to discern subtle nuances in HTTP, DNS, and TCP traffic, coupled with its adaptive learning capabilities, heralds a new era in proactive cybersecurity defenses.

5 Chapter Five
Results

5.1 Introduction

Chapter 5 explores the outcomes and insights gained from our extensive experiments and methods described in the previous chapters. This chapter gets to the heart of our research and provides a detailed look at the results we obtained, from classifying our dataset to analyzing how well our algorithms performed.

We start by explaining how we labeled our created dataset, which is important for training and testing our machine learning models. This part gives clear information about how we labeled the data, making sure our experimental results are trustworthy and dependable.

Subsequently, our focus shifts to a comprehensive analysis of algorithmic performance, where we delve into the intricacies of model evaluation metrics, execution times, and predictive accuracies. By benchmarking our proposed approach against established algorithms and related approaches, we gain valuable insights into the efficacy, efficiency, and innovation encapsulated within our methodology.

Central to this chapter is a comparative analysis that highlights the unique contributions of our thesis in the context of existing research and methodologies. This comparative lens not only validates the novelty and relevance of our approach but also illuminates areas of improvement and future research directions.

As we navigate through Chapter 5, we show the empirical evidence and analytical rigor that underpin our research outcomes. The insights garnered here not only validate the hypotheses posited earlier but also lay the groundwork for informed conclusions and actionable recommendations in the realm of ransomware detection and prediction.

5.2 Overview

At the end of this thesis, we present the results of our efforts to create a reliable ransomware detection and prediction model using three different algorithms: Gradient Boosting, Random Forest (RF), and K-Nearest Neighbors (KNN). The main goal of this chapter is to provide a comprehensive analysis of the effectiveness and performance of the models created. We examine the output of each algorithm through a systematic evaluation, revealing their unique advantages and disadvantages. This chapter also provides insights into the labeling results obtained for this carefully constructed dataset, improving our understanding of the models capabilities to differentiate and predict accidents.

The choice of Random Forest, Gradient Boosting, and K-Nearest Neighbors (KNN) algorithms in this study was strategic and based on several factors. Random Forest and Gradient Boosting were selected due to their ensemble learning nature, which combines multiple decision trees to improve predictive accuracy and handle complex datasets effectively. These algorithms are known for their robustness against overfitting and ability to capture nonlinear relationships within the data, making them suitable for our classification task of identifying ransomware families and benign traffic.

Additionally, K-Nearest Neighbors (KNN) was included for its simplicity and effectiveness in handling data without assuming any underlying distribution. Its instance-based learning approach makes it suitable for smaller datasets and provides a straightforward method for classification based on similarities between data points.

Considering the criticality of execution time in our research, these algorithms were chosen for their computational efficiency while maintaining competitive accuracy. Furthermore, the

dataset's size, with a vast number of observations, influenced our decision to prioritize algorithms that could handle such scale effectively. While neural networks offer powerful capabilities, their resource-intensive training and potential complexities were not aligned with our emphasis on computational efficiency and scalability for our research objectives of detecting and predicting ransomware based on network behavior. Thus, the chosen ensemble and instance-based algorithms provided a pragmatic and effective approach to address our research goals within the constraints of execution time and dataset size.

5.3 Labeling Results for Generated Dataset

Our study is based on a carefully constructed dataset that was generated from a testbed designed to replicate different cyber threat scenarios. The generated dataset is an essential part of the analysis process for our machine learning algorithms, which include Gradient Boosting, Random Forest (RF), and K-Nearest Neighbors (KNN).

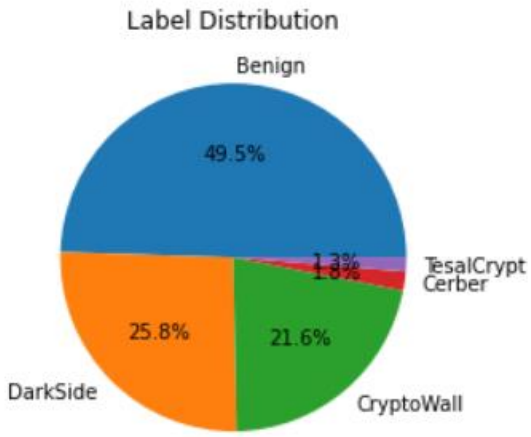


Figure 5.1 Label distribution

Figure 5.1 provides a snapshot of the distribution of labels within the generated dataset. The figure indicates that 49.5% of instances were classified as “benign,” representing typical, harmless network traffic. In contrast, the dataset includes malicious activities, with

“DarkSide” accounting for 25.8%, “CryptoWall” 21.6%, “Cerber” 1.8%, and “TeslaCrypt” 1.3%. This diverse classification distribution is pivotal for algorithms to learn and generalize patterns associated with normal and malicious network behaviors. All of these malicious activities came from the injection of each ransomware for each experiment.

A significant portion of the 'Benign' label in our dataset stems from routine network activities originating from victim devices. These include benign connections related to everyday operations, such as Windows updates, legitimate software installations, and other routine communication protocols. The injection of these benign instances ensures a realistic representation of normal network traffic, enabling our algorithms to discern between genuine and potentially harmful activities.

As we explore the next parts, the results of our algorithms within the framework of this labeled dataset provide insight into how well they work to differentiate between benign and harmful behaviors, which helps improve cybersecurity threat detection and mitigation tactics.

5.4 Algorithmic Performance Analysis

In this part, the three algorithms that are being used -K-Nearest Neighbors (KNN), Random Forest (RF), and Gradient Boosting- are carefully examined in order to show their unique advantages and relative performance measures. The evaluation includes important measures including recall, accuracy, precision, and F1 score to give a useful comparison of each one's effectiveness.

- Random Forest (RF)

In our investigation, Random Forest came out on top with an impressive 88% accuracy. This model shows unparalleled recall, F1 score, and accuracy, highlighting its strong ability to reliably identify incidents. 6.3 seconds is a very short execution time, making it an excellent choice for real-time applications.

- Gradient Boosting

Gradient boosting comes in a close second, with an excellent accuracy rate of 87%. This approach has a longer execution time of 22 seconds, but achieves competitive precision, recall, and F1 score. The trade-off between accuracy and execution time requires careful consideration for application in scenarios where computational speed is less important.

The Gradient Boosting algorithm showcasing its effectiveness in classifying ransomware and benign traffic based on network behavior. However, this came at the cost of an execution time of 22 seconds, significantly higher compared to the Random Forest and K-Nearest Neighbors (KNN) algorithms. This trade-off between accuracy and execution time reflects the algorithm's complexity and computational requirements. Gradient Boosting, while offering high accuracy by iteratively improving weak learners, tends to require more computational resources and time for training and inference. In contrast, Random Forest and KNN, known for their computational efficiency and faster execution times, may sacrifice a slight degree of accuracy compared to Gradient Boosting but provide a more practical solution, especially in scenarios where real-time or near-real-time predictions are crucial. Therefore, the decision between these algorithms should consider the balance between

accuracy and computational efficiency based on specific application requirements and constraints.

- K-Nearest Neighbors (KNN)

Despite achieving a slightly lower accuracy of 79%, KNN offers an interesting performance profile with an exceptionally low execution time of 0.4 seconds. Efficiency in implementation can be attributed to the lazy learning approach, which postpones computation until classification is required. Although KNN may lag in accuracy compared to RF and Gradient Boosting, its fast implementation makes it a compelling choice for scenarios that require real-time responsiveness.

A detailed summary of the performance metrics -Accuracy, Precision, F1-Score, Recall, and Execution Time- for the three algorithms -Gradient Boosting, Random Forest (RF), and K-Nearest Neighbors (KNN)- is shown in table 5.1. The graph illustrates the subtle advantages of each approach, with RF topping the group with an astounding accuracy of 88%. With an accuracy of 87%, Gradient Boosting trails closely, while KNN shows a little lower accuracy of 79%. The computing efficiency of the methods is demonstrated by the execution time, while the precision, F1-Score, and recall numbers offer insights into their classification skills. With the help of this comprehensive representation, one may make well-informed decisions based on the unique performance needs of various real-world applications.

Table 5.1 Algorithms' results

Metric	Random Forest	Gradient Boosting	KNN
Accuracy	0.88081	0.873394	0.796716
Precision	0.790142	0.875059	0.79222
Recall	0.716227	0.873394	0.796716
F1-score	0.74043	0.870478	0.792863
Execution Time (s)	6.33815	22.6853	0.426105

The performance dynamics are further illustrated by the graphical representations accompanying in figure 5.2. The accuracy graph -the blue line- clearly shows how the three algorithms perform in comparison, with RF advancing, gradient boosting, and KNN. The execution time graph which is the red one, contrasts sharply with the longer execution times of RF and gradient boosting, clearly highlights the efficiency of KNN.

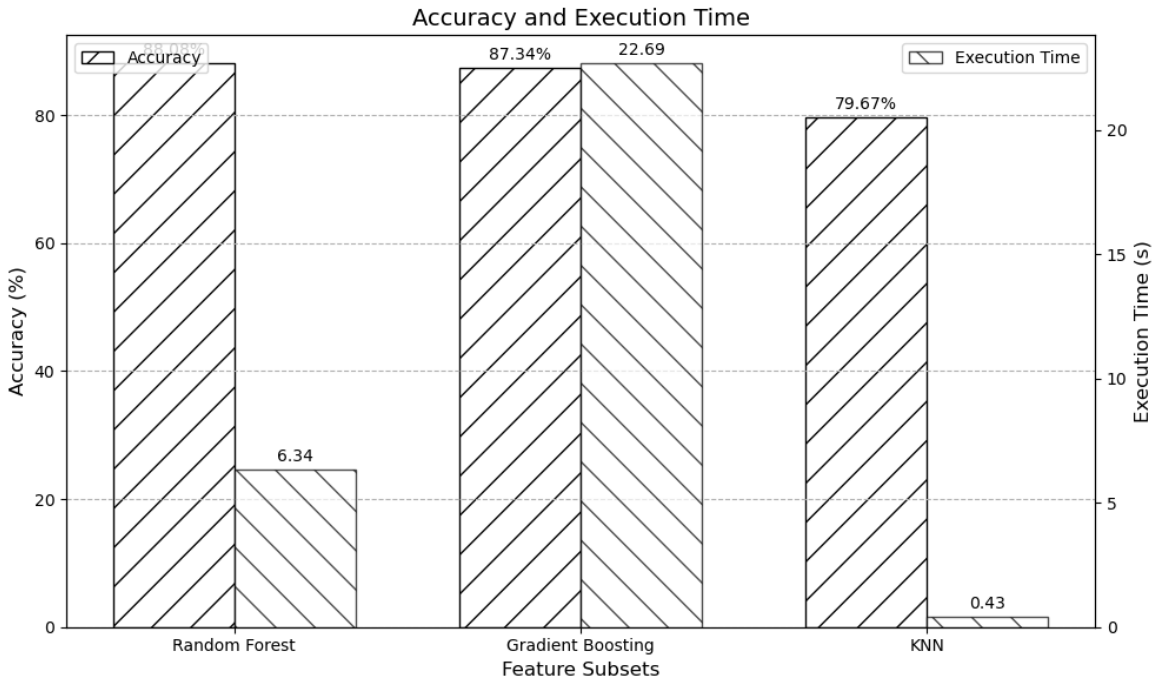


Figure 5.2 Accuracy and Execution Time

The Random Forest algorithm achieved an accuracy of 88% with an F1 score of 74% and an execution time of 6 seconds. This algorithm's strength lies in its ensemble learning approach, where multiple decision trees are combined to improve accuracy and handle complex data effectively. The relatively high accuracy indicates that Random Forest is successful in correctly classifying ransomware and benign traffic instances, albeit with some misclassifications that impact its F1 score.

On the other hand, the Gradient Boosting algorithm achieved an accuracy of 87% with an F1 score of 87% but had a longer execution time of 22 seconds. Gradient Boosting excels in capturing complex relationships within the data and improving predictive accuracy through iterative training of weak learners. The higher F1 score suggests better precision and recall balance, leading to fewer misclassifications compared to Random Forest. However, the trade-off is the increased computational resources and time required for training and inference.

Lastly, the K-Nearest Neighbors (KNN) algorithm achieved an accuracy of 79% with an F1 score of 79% and the shortest execution time of 0.4 seconds. KNN's simplicity and instance-based learning approach make it efficient for smaller datasets and quick predictions based on similarity to neighboring data points. While it may not match the accuracy of Random Forest or the precision-recall balance of Gradient Boosting, its fast execution time makes it suitable for real-time or near-real-time applications where speed is critical.

In summary, the choice of algorithm depends on the specific requirements of the application. Random Forest offers a good balance between accuracy and execution time, Gradient Boosting prioritizes precision-recall balance at the cost of longer execution times, and KNN provides fast predictions but with slightly lower accuracy compared to ensemble methods.

Confusion Matrices:

Confidence matrices for each model are provided in order to provide further insight into algorithmic performance. These matrices highlight the distinct advantages and disadvantages of classification, offering insightful information on how each technique performs with various classes.

The integration of numerical outcomes, graphical representations, and confusion matrices allows for a comprehensive comprehension of the intricate operations of KNN, RF, and Gradient Boosting. This, in turn, aids in making well-informed decisions on their use in diverse real-world contexts.

Confusion Matrix Analysis for Random Forest:

The Random Forest algorithm's confusion matrix offers a thorough analysis of how well it performs in categorization for various classes. Every column shows the predicted class, while every row shows the actual class. Labels are assigned to the classes as the following: Benign traffic (0), Cerber (1), TeslaCrypt (2), CryptoWall (3), and DarkSide (4).

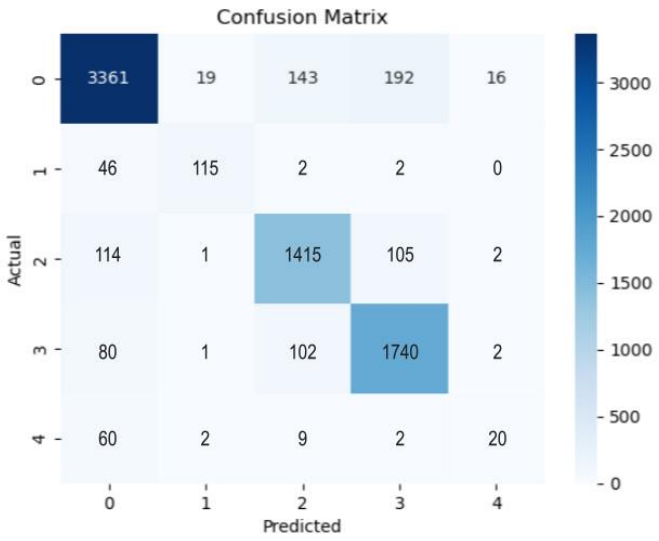


Figure 5.3 Random Forest Confusion matrix

Benign Traffic (0): The majority of benign traffic instances (3361) are correctly classified, with some misclassifications into Cerber (46), TeslaCrypt (114), CryptoWall (80), and DarkSide (60).

Cerber (1): Cerber instances are predominantly classified correctly (115), but there are instances misclassified into Benign traffic (19), TeslaCrypt (1), CryptoWall (1), and DarkSide (2).

TeslaCrypt (2): Most TeslaCrypt instances are accurately identified (1415), with misclassifications into Benign traffic (143), Cerber (2), CryptoWall (102), and DarkSide (9).

CryptoWall (3): The Random Forest algorithm effectively recognizes CryptoWall instances (1740) with misclassifications into Benign traffic (192), Cerber (192), TeslaCrypt (105), and DarkSide (2).

DarkSide (4): The algorithm performs well in identifying DarkSide instances (20), with misclassifications into Benign traffic (16), TeslaCrypt (2), and CryptoWall (2).

The confusion matrix analysis for the Random Forest (RF) algorithm results provides valuable insights into its performance in classifying ransomware and benign traffic. The model exhibits robust predictive capabilities for benign traffic ('Benign'), as evidenced by 3361 true positives and high precision and recall values. However, challenges arise in accurately identifying certain ransomware families. For instance, the 'Cerber' class shows lower precision, recall, and F1 score compared to other classes, indicating potential misclassifications and the need for improved model accuracy. Similarly, the 'DarkSide' class also demonstrates lower F1 scores, highlighting difficulties in distinguishing this particular

type of ransomware. Conversely, the model performs reasonably well for 'TesalCrypt' and 'CryptoWall' classes, albeit with some misclassifications. These findings suggest that while the RF algorithm effectively detects benign traffic, further optimization or additional data may be necessary to enhance its accuracy in identifying specific ransomware families, particularly 'Cerber' and 'DarkSide'.

Confusion matrix for gradient boosting:

Figure 5.4 illustrates the confusion matrix for the Gradient Boosting algorithm, providing an insightful perspective on its classification performance. The matrix reveals how well Gradient Boosting distinguishes between different classes, including Benign traffic, Cerber, TeslaCrypt, CryptoWall, and DarkSide. As the random forest matrix each row represents the actual class, and each column corresponds to the predicted class. The interpretation of the confusion matrix allows for a comprehensive evaluation of Gradient Boosting's strengths and areas that may require attention in handling specific classes.

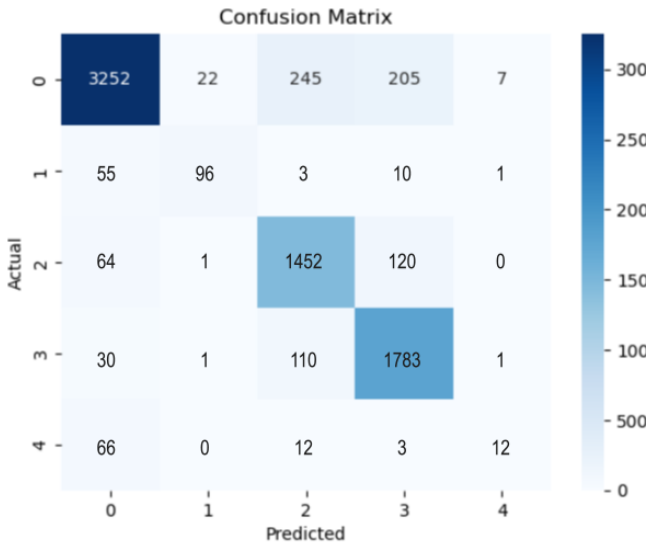


Figure 5.4 Gradient Boosting confusion matrix

The confusion matrix analysis for the Gradient Boosting classifier results reveals notable patterns in its classification performance for ransomware and benign traffic. The model demonstrates a relatively high number of true positives for benign traffic ('Benign'), indicating good accuracy in identifying this class. However, it faces challenges in correctly classifying certain ransomware families. For example, the 'Cerber' class shows moderate precision and recall values, with an F1 score reflecting a balance between precision and recall. This suggests a reasonable but not optimal performance in distinguishing 'Cerber' ransomware. The 'DarkSide' class also exhibits similar performance characteristics, with a balanced F1 score indicating room for improvement in precision and recall. On the other hand, the model performs well for 'TesalCrypt' and 'CryptoWall' classes, showing high precision, recall, and F1 scores. These results imply that the Gradient Boosting classifier excels in identifying 'TesalCrypt' and 'CryptoWall' ransomware variants but may require fine-tuning or additional data to improve its accuracy in classifying 'Cerber' and 'DarkSide' ransomware types.

Confusion matrix for KNN:

Figure 5.5 shows the confusion matrix for the K-Nearest Neighbors (KNN) algorithm, shedding light on its classification performance across the same classes. The matrix delineates the correspondence between actual and predicted labels, offering valuable insights into the algorithm's efficacy in differentiating between diverse traffic patterns. As with Gradient Boosting, this analysis aids in understanding the nuances of KNN's performance, providing a foundation for optimization and refinement in real-world applications.

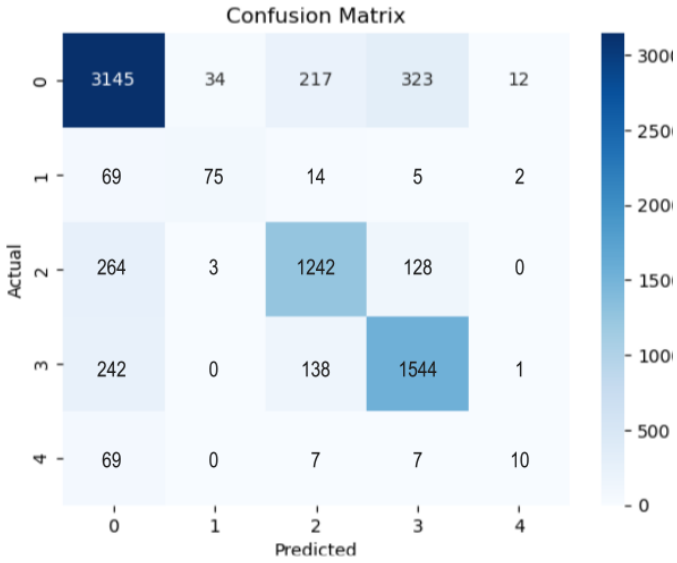


Figure 5.5 KNN confusion matrix

The confusion matrix analysis for the K-Nearest Neighbors (KNN) algorithm showcases its performance in classifying ransomware and benign traffic. The model exhibits a notable number of true positives for benign traffic ('Benign'), indicating a strong ability to identify this class accurately. However, challenges are evident in correctly classifying certain ransomware families. For instance, the 'Cerber' class shows lower precision and recall values, leading to a lower F1 score, which suggests potential misclassifications and the need for improved model accuracy. Similarly, the 'DarkSide' class also displays lower precision, recall, and F1 score, indicating difficulties in distinguishing this type of ransomware. On the other hand, the model performs well for 'TesalCrypt' and 'CryptoWall' classes, with high precision, recall, and F1 scores, indicating its effectiveness in identifying these ransomware variants. These findings suggest that while the KNN algorithm is proficient in detecting benign traffic and certain ransomware families, further optimization or additional data may be beneficial for enhancing its accuracy in classifying 'Cerber' and 'DarkSide' ransomware types.

5.5 Thesis Contributions: A Comparative Analysis with Related Approaches

This thesis is at the forefront of ransomware detection and prediction, making it unique through comprehensive studies, creative approaches, and noteworthy achievements. A comprehensive comparison with similar approaches demonstrates the distinct contributions and progress made in the search for effective cybersecurity solutions.

- **Investigation of Four Ransomware Families**

Unlike previous research, this study looks at the unique features of four known ransomware families, using 145 samples for a comprehensive examination. This comprehensive investigation provides a more complete and representative dataset for model training and evaluation, enabling precise knowledge of various ransomware behaviors.

- **Outstanding Accuracy in Binary and Multiclass Detection**

The proposed model performs extremely well, detecting binary malware with an accuracy rate of 86%. Furthermore, it achieves a remarkable 88% accuracy rate in multi-category detection, successfully distinguishing between secure traffic and complex patterns of four different ransomware families. This performance highlights the flexibility of the model and demonstrates its effectiveness in practical situations.

- **Leverage network flow data to enhance efficiency**

The decision to use network traffic data rather than static behavioral indications such as host behavior or registry files is an important addition. Our model's ability to record real-time patterns from dynamic network traffic allows it to quickly adjust to changing ransomware

strategies. Compared to static behavioral assessments, this method helps achieve a detection system that is both more responsive and accurate.

- **Exceptional Execution Time Using Random Forest**

The Random Forest model has an impressive speed of 6.3 seconds, which the thesis describes as an establishing achievement in terms of execution time. This performance is particularly noteworthy when compared to other research projects that frequently suffer from longer computing periods. The suggested model's practical applicability in time-sensitive cybersecurity scenarios is enhanced by its rapid execution.

- **Protocol Diversity**

Our thesis takes a protocol-agnostic approach, in contrast to some previous research that mainly focuses on one or two specific protocols, such HTTP or DNS. The protocols of each ransomware family are highlighted in detail, and the model is trained and tested on a wide range of protocols. This methodology improves the model's flexibility in response to the many strategies used by ransomware, guaranteeing a more thorough and adaptable detection capacity.

In summary, the contributions of this thesis set it apart by not only achieving outstanding detection accuracies but also by introducing innovative methodologies and considerations. The emphasis on diverse ransomware families, efficient execution times, and the protocol-agnostic approach collectively positions this research as a pioneering effort in advancing the state of ransomware detection and prediction.

5.6 Conclusion

This study highlights the importance of using modern technology to improve ransomware detection capabilities. To ensure detailed analysis of ransomware behaviors is performed in a secure and authentic environment, we created a custom-designed testbed as part of our research for a reliable solution. In contrast to traditional methods, we intentionally avoided virtual environment to avoid any ransomware detection tools, which strengthens the validity and reliability of our results.

One of the most important parts of our contribution is the challenging task of classifying and analyzing each ransomware traffic pattern in our dataset. This comprehensive examination explores the unique characteristics of four known ransomware families, including 145 samples. By examining in-depth the distinct characteristics and strategies used by each form of ransomware, we enhanced our model's ability to identify small patterns, resulting in a more accurate and flexible detection process.

Furthermore, our approach extends beyond binary detection, achieving an exceptional 88% accuracy in multiclass detection, effectively differentiating between benign traffic and the intricate patterns of four distinct ransomware families. This accomplishment underscores the versatility and real-world applicability of our model in addressing the evolving landscape of cyber threats.

We have creatively used network flow statistics to our advantage, using a protocol-agnostic, dynamic approach. Instead of relying on static behavioral indicators, our methodology leverages the dynamic nature of network traffic to quickly adapt to new ransomware strategies. This difference from traditional approaches highlights the potential impact of our

model in a variety of dynamic and diverse network contexts and shows how effective it is in identifying threats in real-time.

Moreover, the execution time of our Random Forest model, a 6.3 seconds execution time, marks a significant advancement in computational efficiency compared to other research endeavors. This achievement not only contributes to the practical viability of our proposed solution but also positions it as a promising candidate for time-sensitive cybersecurity applications.

It is clear that our contributions go beyond the immediate context of ransomware detection when we consider the findings and techniques presented in this study. We propose detecting cyber threats by adopting the latest technologies, customizing our testing, classifying and examining ransomware traffic with great care. The results, confirmed through extensive testing, support the accuracy with which our approach can identify ransomware using digital forensics. This, coupled with the use of application analysis tools and artificial intelligence algorithms, highlights the strength and effectiveness of our strategy in the ongoing battle against changing cyber threats.

Our study illustrates the risks inherent in connecting servers directly to the Internet in a larger context. Examining the connections of each ransomware variant shows the large number of connections made to the malicious actors' existing servers. This underscores the urgency of stronger security measures to mitigate the threats these communications present. As we move forward, our study improves our knowledge of the complex interactions between cyber threats and the vulnerable architecture of interconnected systems, as well as improving detection technology.

5.7 Future Implications and Enhancements

Our successful efforts in detecting ransomware open the door for next improvements that increase the reach and effectiveness of cybersecurity defenses:

- Inclusion of More Ransomware Families

  Increasing the size of our dataset to include a wider variety of ransomware families improves the stability of our detection model. More families should be included in future studies to guarantee a thorough comprehension of various ransomware behaviors and to strengthen the model against new threats.

- Exploration of Different Data Sources

  Even though we are now concentrating on network traffic data, we can improve our detection skills even more by investigating other data sources. Future research might look into how other data types, including metadata or endpoint behavior, can be integrated while yet maintaining or perhaps increasing execution time efficiency.

- Real-Time Decision Support

  Incorporating real-time decision support features into our model can empower security professionals with actionable insights. Enhancements that provide instant guidance on response strategies, leveraging the analyzed network behavior, will be instrumental in fortifying organizations against ransomware attacks.

- Extending to Cloud Environments

  Applying our concept to cloud environments is essential given the growing use of cloud computing. Following investigations may examine modifications and

enhancements for cloud-based systems, defending enterprises against ransomware risks in various implementation situations.

Our future directions align with the overarching goal of fortifying organizations against ransomware threats. By expanding the scope of ransomware families, exploring alternative data sources, and focusing on rapid response strategies, we aim to provide a more comprehensive and adaptive defense mechanism. These improvements, coupled with the ability to leverage network behavior to mitigate damage, place our approach at the forefront of proactive cybersecurity measures.

References

17 Ransomware Examples & How They Occurred | UpGuard (2023). Available at: https://www.upguard.com/blog/ransomware-examples (Accessed: 14 December 2023).

A Brief History of Ransomware (2023). Available at: https://www.varonis.com/blog/a-brief-history-of-ransomware (Accessed: 2 December 2023).

Abdulrazaq, T. and Al-Morjan, A. (2010) An Investigation into a Digital Forensic Model to Distinguish between 'Insider' and 'Outsider'.

Ahmed, Y.A., Koçer, B. and Al-Rimy, B.A.S. (2020) 'Automated Analysis Approach for the Detection of High Survivable Ransomware', KSII Transactions on Internet and Information Systems, 14(5), pp. 2236–2257. Available at: https://doi.org/10.3837/tiis.2020.05.021.

Aldauiji, F., Batarfi, O. and Bayousef, M. (2022) 'Utilizing Cyber Threat Hunting Techniques to Find Ransomware Attacks: A Survey of the State of the Art', IEEE Access, 10, pp. 61695–61706. Available at: https://doi.org/10.1109/ACCESS.2022.3181278.

Almashhadani, A.O. et al. (2019) 'A Multi-Classifier Network-Based Crypto Ransomware Detection System: A Case Study of Locky Ransomware', IEEE Access, 7, pp. 47053–47067. Available at: https://doi.org/10.1109/ACCESS.2019.2907485.

Alotaibi, F.M. and Vassilakis, V.G. (2021) 'SDN-Based Detection of Self-Propagating Ransomware: The Case of BadRabbit', IEEE Access, 9, pp. 28039–28058. Available at: https://doi.org/10.1109/ACCESS.2021.3058897.

Alqahtani, A. and Sheldon, F.T. (2022) ‘A Survey of Crypto Ransomware Attack Detection Methodologies: An Evolving Outlook’, Sensors. MDPI. Available at: https://doi.org/10.3390/s22051837.

Arabo, A. et al. (2020) ‘Detecting ransomware using process behavior analysis’, in Procedia Computer Science. Elsevier B.V., pp. 289–296. Available at: https://doi.org/10.1016/j.procs.2020.02.249.

Arfeen, A. et al. (2022) ‘Process based volatile memory forensics for ransomware detection’, Concurrency and Computation: Practice and Experience, 34(4). Available at: https://doi.org/10.1002/cpe.6672.

Bae, S. Il, Lee, G. Bin and Im, E.G. (2020) ‘Ransomware detection using machine learning algorithms’, in Concurrency and Computation: Practice and Experience. John Wiley and Sons Ltd. Available at: https://doi.org/10.1002/cpe.5422.

Benzekri, A. et al. (eds) (2020) Foundations and Practice of Security. Cham: Springer International Publishing (Lecture Notes in Computer Science). Available at: https://doi.org/10.1007/978-3-030-45371-8.

Berrueta, E. et al. (2022a) Crypto-ransomware detection using machine learning models in file-sharing network scenario with encrypted traffic.

Berrueta, E. et al. (2022b) ‘Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic’, Expert Systems with Applications, 209. Available at: https://doi.org/10.1016/j.eswa.2022.118299.

Cabaj, K. et al. (2015) ‘Network activity analysis of CryptoWall ransomware’, Przeglad Elektrotechniczny, 91(11), pp. 201–204. Available at: https://doi.org/10.15199/48.2015.11.48.

cicflowmeter · PyPI (2023). Available at: https://pypi.org/project/cicflowmeter/ (Accessed: 15 January 2024).

Computer network isolation (no date). Available at: https://support.kaspersky.com/KESWin/12.0/en-US/214777.htm (Accessed: 3 December 2023).

Daku, H., Zavarsky, P. and Malik, Y. (2018) 'Behavioral-Based Classification and Identification of Ransomware Variants Using Machine Learning', in Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018. Institute of Electrical and Electronics Engineers Inc., pp. 1560–1564. Available at: https://doi.org/10.1109/TrustCom/BigDataSE.2018.00224.

DarkSide Ransomware as a Service (RaaS) - United States Department of State (2023). Available at: https://www.state.gov/darkside-ransomware-as-a-service-raas/ (Accessed: 11 December 2023).

Digital Scholar, M. et al. (2017) Ransomware in Healthcare Facilities: The Future is Now. Available at: http://mds.marshall.edu/mgmt_faculty.

Ghouse, M. and Nene, M.J. (2019) Data Leakage Prevention for Data in Transit using Artificial Intelligence and Encryption Techniques.

Hassan, W.U., Bates, A. and Marino, D. (2020) 'Tactical provenance analysis for endpoint detection and response systems', in Proceedings - IEEE Symposium on Security and Privacy. Institute of Electrical and Electronics Engineers Inc., pp. 1172–1189. Available at: https://doi.org/10.1109/SP40000.2020.00096.

Hwang, J. et al. (2020) 'Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques', Wireless Personal Communications, 112(4), pp. 2597–2609. Available at: https://doi.org/10.1007/s11277-020-07166-9.

Institute of Electrical and Electronics Engineers. Bangladesh Section and IEEE Communications Society (2017) ICCIT : 2017 20th International Conference of Computer and Information Technology : 22-24 December 2017.

International ISC Conference on Information Security and Cryptology 12. 2015 Rescht et al. (2015) 12th International ISC Conference on Information Security and Cryptology University of Guilan, Rasht, Iran, 8-10 September 2015.

Internet and social media users in the world 2023 | Statista (2023). Available at: https://www.statista.com/statistics/617136/digital-population-worldwide/ (Accessed: 2 December 2023).

Kara, I. and Aydos, M. (2022) 'The rise of ransomware: Forensic analysis for windows based ransomware attacks', Expert Systems with Applications, 190. Available at: https://doi.org/10.1016/j.eswa.2021.116198.

KARA, İ., AYDOS, M. and BOZKIR, A.S. (2020) 'Zararlı Yazılımların Karekterislik Analizi: Cryptowall Fidye Yazılım Analizi', European Journal of Science and Technology, pp. 486–493. Available at: https://doi.org/10.31590/ejosat.araconf63.

Kurniawan, A. and Riadi, I. (2018a) 'Detection and Analysis Cerber Ransomware Based on Network Forensics Behavior', Article in International Journal of Network Security, 20(5), pp. 836–843. Available at: https://doi.org/10.6633/IJNS.201809.

Kurniawan, A. and Riadi, I. (2018b) 'Detection and Analysis Cerber Ransomware Based on Network Forensics Behavior Detection and Analysis Cerber Ransomware Using Network Forensics Investigative Methodology View project Detection and Analysis Cerber Ransomware Based on Network Forensics Behavior', Article in International Journal of Network Security, 20(5), pp. 836–843. Available at: https://doi.org/10.6633/IJNS.201809.

Lapiedra, J. (2000) The Information Security Process Preve ntion, Detection and Response.

Lee, K., Lee, S.-Y. and Yim, K. (2019) Machine Learning based File Entropy Analysis for Ransomware Detection in Backup Systems.

May, E.C. et al. (1994) SHANNON ENTROPY: A POSSIBLE INTRINSIC TARGET PROPERTY, The Journal of Parapsychology.

Möller, D.P.F. (2023) Advances in Information Security 103 Guide to Cybersecurity in Digital Transformation Trends, Methods, Technologies, Applications and Best Practices.

Panagiotou, P. et al. (2021) 'Host-based Intrusion Detection Using Signature-based and AI-driven Anomaly Detection Methods', Information & Security: An International Journal, 50, pp. 37–48. Available at: https://doi.org/10.11610/isij.5016.

Reddy, D.K.K. et al. (2021) 'Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities', Transactions on Emerging Telecommunications Technologies, 32(7). Available at: https://doi.org/10.1002/ett.4121.

Richardson, R. and North, M.M. (2017) Ransomware: Evolution, Mitigation and Prevention. Available at: https://digitalcommons.kennesaw.edu/facpubs/4276.

Sgandurra, D. et al. (2016) 'Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection'. Available at: http://arxiv.org/abs/1609.03020.

Singh, Jagsir and Singh, Jaswinder (2021) 'A survey on machine learning-based malware detection in executable files', Journal of Systems Architecture. Elsevier B.V. Available at: https://doi.org/10.1016/j.sysarc.2020.101861.

Stallings, W. (2010) Cryptography and network security : principles and practice.

TeslaCrypt Ransomware Threat Analysis | Secureworks (2023). Available at: https://www.secureworks.com/research/teslacrypt-ransomware-threat-analysis (Accessed: 14 December 2023).

Tseng, A. et al. (2016) 'Deep learning for ransomware detection', IEICE Technical Report; IEICE Tech. Rep., 116(282), pp. 87–92.

Umar, R., Riadi, I. and Kusuma, R.S. (2021) 'Analysis of Conti Ransomware Attack on Computer Network with Live Forensic Method', IJID (International Journal on Informatics for Development), 10(1), pp. 53–61. Available at: https://doi.org/10.14421/ijid.2021.2423.

Vielberth, M. et al. (2020) 'Security Operations Center: A Systematic Study and Open Challenges', IEEE Access [Preprint]. Available at: https://doi.org/10.1109/ACCESS.2020.3045514.

Why Is Logistic Regression a Classification Algorithm? | Built In (2023). Available at: https://builtin.com/machine-learning/logistic-regression-classification-algorithm (Accessed: 9 November 2023).

فك التشفير لخوادم وزارة الصحة بعد هجوم 'سيبراني' خارجي (2023). Available at: https://www.bnews.ps/index.php/ar/node/19799 (Accessed: 5 November 2023).

Appendices

Appendix (1) Dependencies for The Project

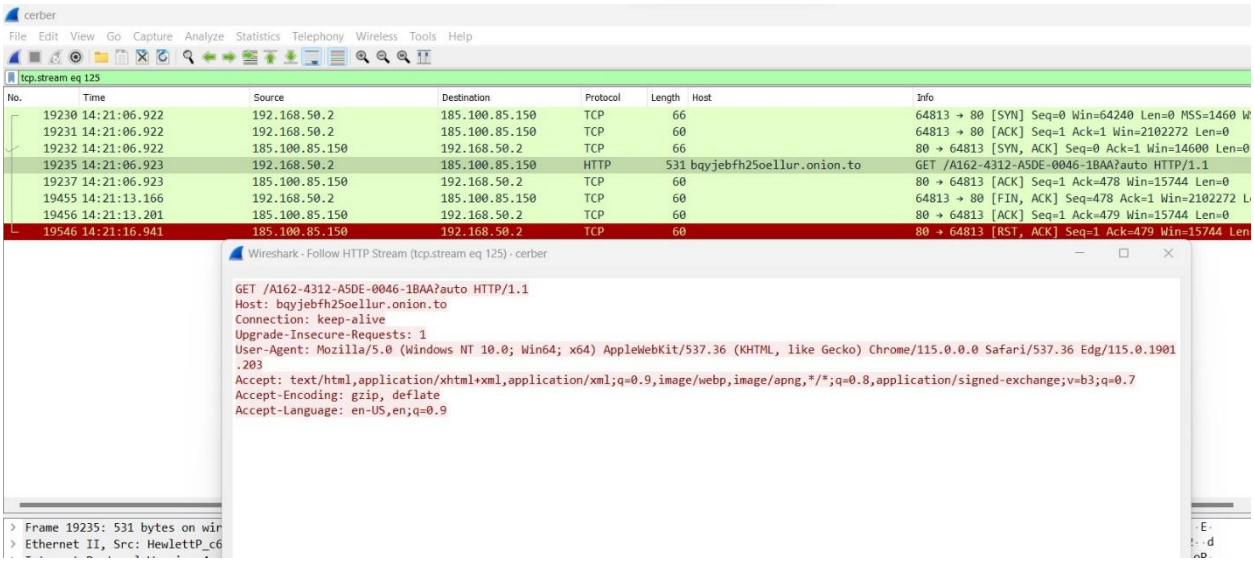


Figure 6.1 HTTP request for Cerber Ransomware

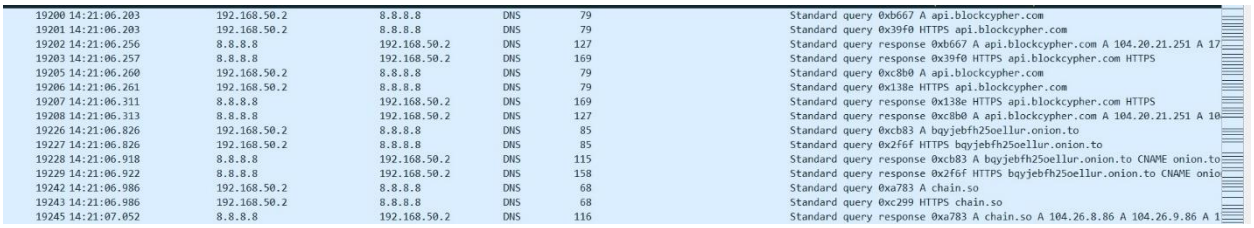


Figure 6.2 DNS traffic for Cerber Ransomware

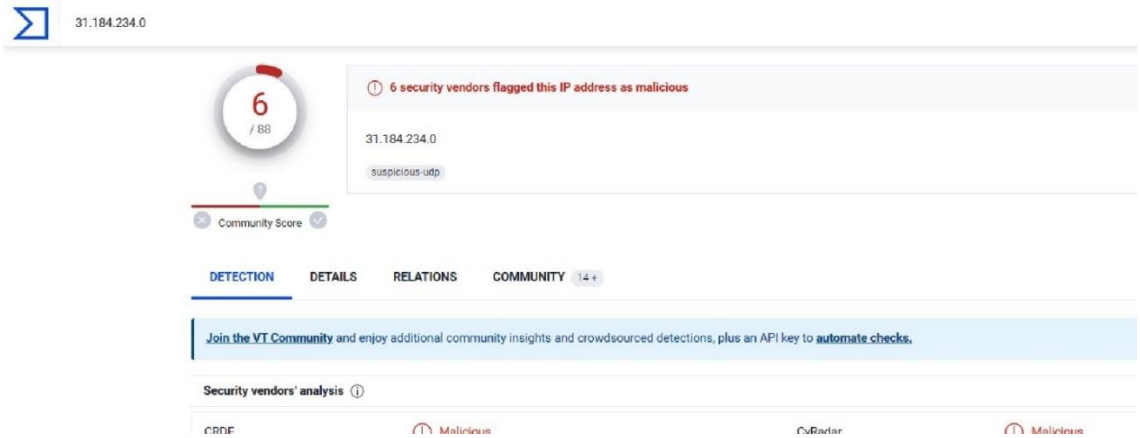


Figure 6.3 Checking one of the connected IP related to Cerber traffic

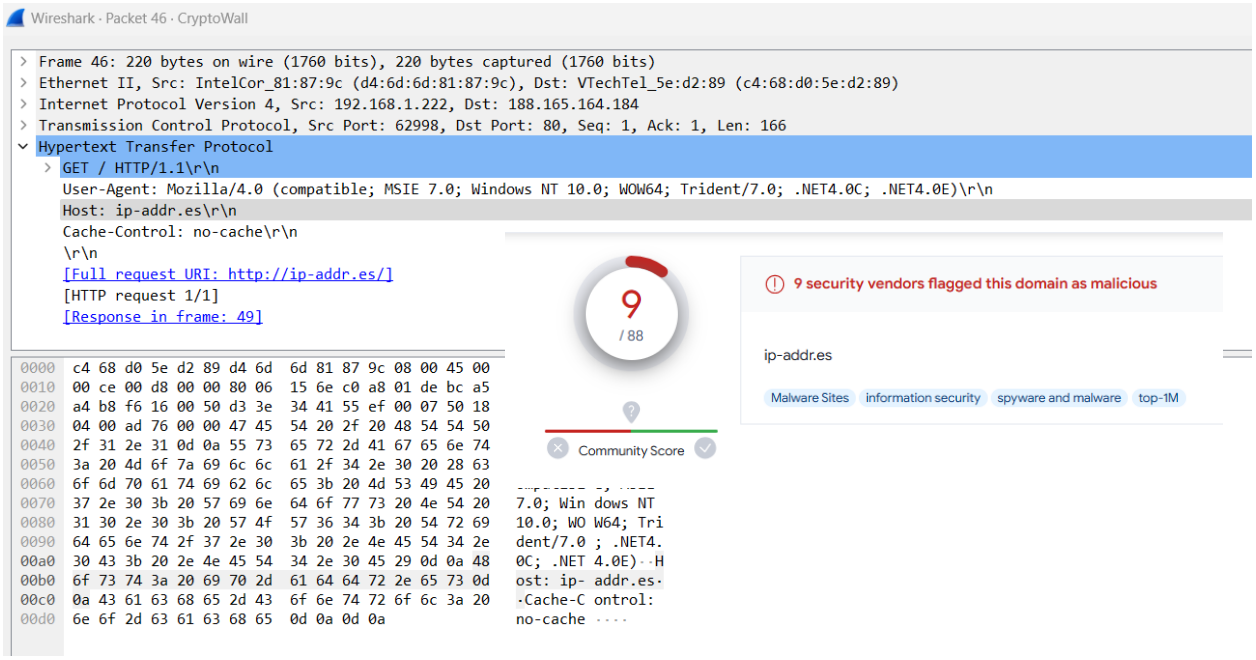


Figure 6.4 Checking one of the connected domains for CryptoWall

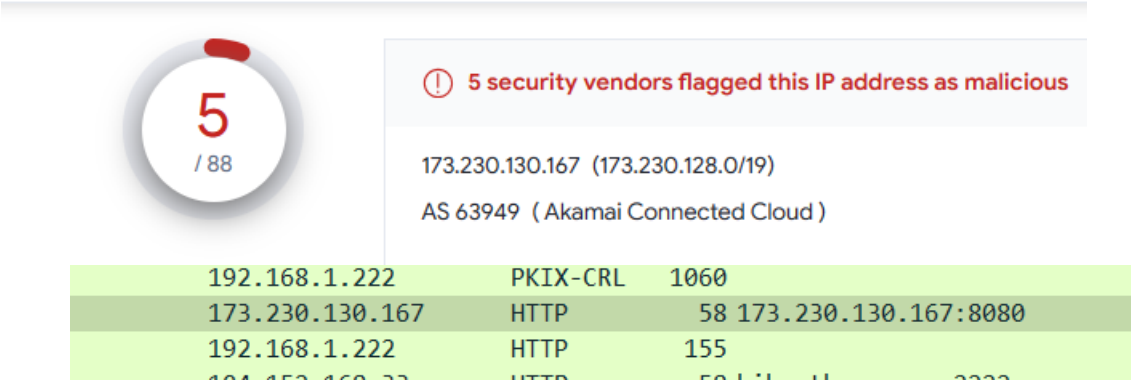


Figure 6.5 One of the connected Ip's used by CryptoWall

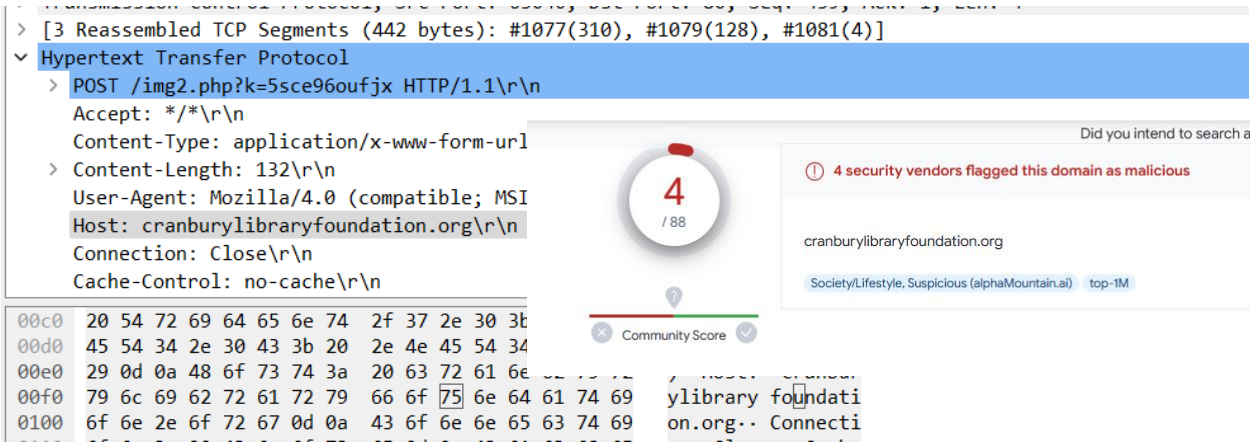


Figure 6.6 HTTP Post to malicious domain used by CryptoWall

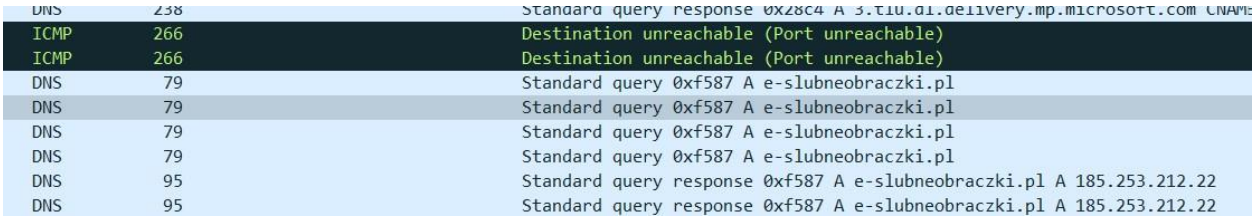


Figure 6.7 TeslaCrypt DNS traffic

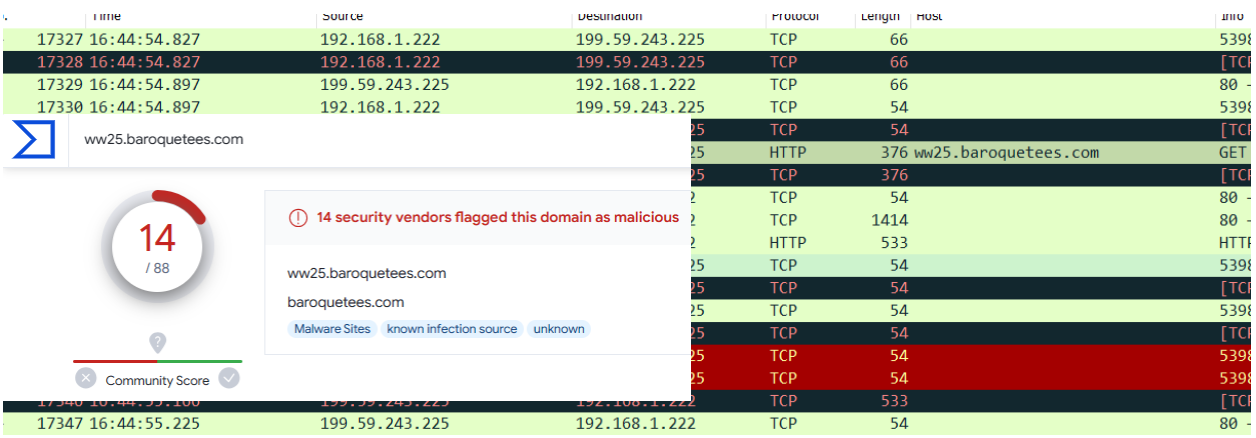


Figure 6.8 HTTP request for DarkSide ransomware

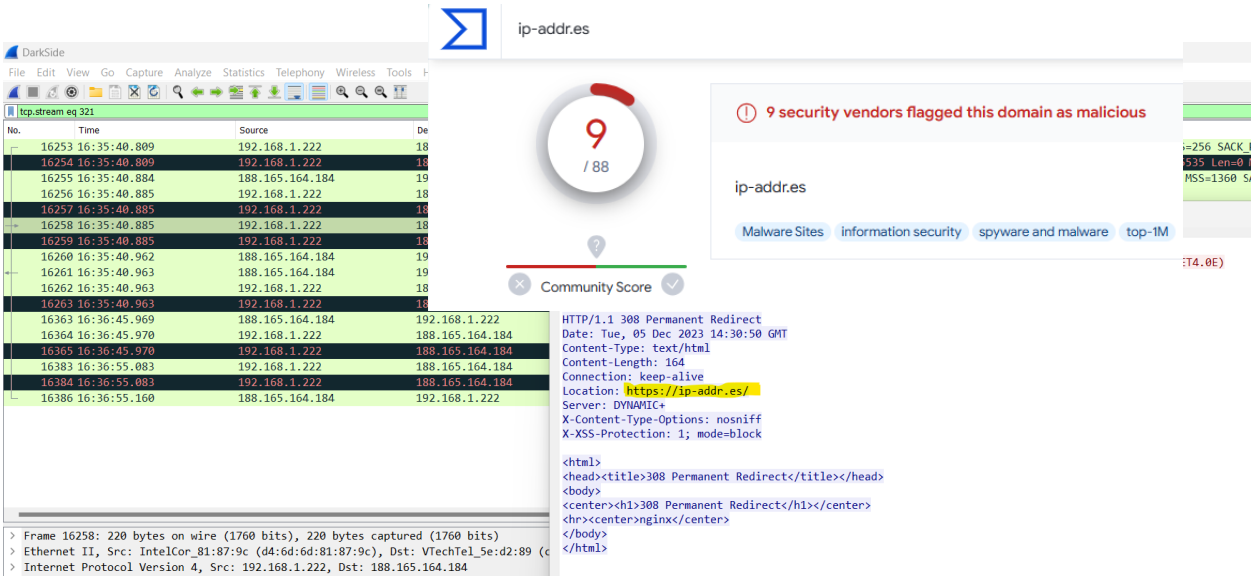


Figure 6.9 Malicious domain used by DarkSide ransomware

Appendix (2) Code

```
import pandas as pd

## Define the list of IPs that used by the ransomware
target_ips = [
"188.165.164.184",
"104.152.168.33",
"199.59.243.225",
"37.59.49.102",
"46.166.189.98",
"64.91.248.18",
"38.59.20.149",
"89.46.108.72",
"5.249.152.61",
"91.121.12.127",
"94.247.28.29",
"38.207.214.167"
]

## Load the CSV file into a DataFrame
csv_path = r'E:\Thesis\Collected Traffic\DarkSide\DarkSide_Flow.csv'
df = pd.read_csv(csv_path)

## Create a function to check if an IP is in the list of target IPs and lable it as 4
def check_ip(ip):
    for target_ip in target_ips:
        if ip.startswith(target_ip):
            return 4
    return 0

## Apply the check_ip function to the "Src IP" and "Dst IP" columns and create the "Label" column
df['Label'] = df.apply(lambda row: check_ip(row['Src IP']) or check_ip(row['Dst IP']), axis=1)

## Save the updated DataFrame to a new CSV file
output_csv_path = r'E:\Thesis\Collected Traffic\DarkSide\DarkSide_Labeled.csv'
df.to_csv(output_csv_path, index=False)

print("Label column added and DataFrame saved to", output_csv_path)
```

Code 1 Labeling code

```
## Select path of our datasat folder
filepath = 'E:/Thesis/Collected Traffic/DataSets & FinalCode'
## Read all files with .csv extention
csv_files = [file for file in os.listdir(filepath) if
file.endswith('.csv')]

dataframes = {}  ## Dictionary to store the dataset files

for i, file_name in enumerate(csv_files, start=1):
    file_path = os.path.join(filepath, file_name)

    ## Read the CSV file using Pandas
    df = pd.read_csv(file_path)

    ## Assign a specific name to the DataFrame
    df_name = 'df' + str(i)
    dataframes[df_name] = df

    ## Display the contents of the CSV file
    print("File:", file_name)
    print()  # Add an empty line between each file
```

Code 2 Reading the 4 datasets as 1 file

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

X = data.drop('Label', axis=1)  # Features
y = data['Label']  # Target variable

## Step 1: Binary Classification (Benign or Not Benign)
## Assign 0 to Benign and 1 to Not Benign
y_binary = (y != 0).astype(int)

## Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_binary,
test_size=0.2, random_state=42)

## Create a Random Forest classifier for binary classification
binary_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)
binary_classifier.fit(X_train, y_train)

## Make predictions on the test set
y_pred_binary = binary_classifier.predict(X_test)
```

```
## Evaluate the binary classifier
binary_accuracy = accuracy_score(y_test, y_pred_binary)
print(f'Binary Classification Accuracy: {binary_accuracy:.2f}')

## Step 2: If not Benign, classify the specific type of ransomware
## Filter the data for non-benign traffic
non_benign_indices = (y != 0)
X_non_benign = X[non_benign_indices]
y_non_benign = y[non_benign_indices]

## Split the non-benign data into training and testing sets
X_train_non_benign, X_test_non_benign, y_train_non_benign,
y_test_non_benign = train_test_split(
    X_non_benign, y_non_benign, test_size=0.2, random_state=42
)

## Create a Random Forest classifier for multi-class classification
ransomware_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)
ransomware_classifier.fit(X_train_non_benign, y_train_non_benign)

## Make predictions on the non-benign test set
y_pred_non_benign = ransomware_classifier.predict(X_test_non_benign)

## Evaluate the ransomware classifier
ransomware_accuracy = accuracy_score(y_test_non_benign,
y_pred_non_benign)
print(f'Ransomware Classification Accuracy: {ransomware_accuracy:.2f}')

print('\nClassification Report:')
print(classification_report(y_test_non_benign, y_pred_non_benign))
```

Code 3 Binary and multi-Classification code using random forest algorithm

```
from sklearn.ensemble import GradientBoostingClassifier  # Import the
Gradient Boosting classifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt

selected_features = ['Timestamp_numeric','Init Bwd Win Byts', 'Bwd Header
Len', 'Flow Pkts/s','Bwd Pkts/s']
X = data[selected_features]  # Features
y = data['Label']   # Target variable

start_time = time.time()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
gb_classifier = GradientBoostingClassifier(n_estimators=100,
random_state=42)
gb_classifier.fit(X_train, y_train)
end_time = time.time()
execution_time_gb = end_time - start_time

y_pred = gb_classifier.predict(X_test)

from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

## Assuming y_test and y_pred are arrays or lists representing the true
labels and predicted labels
accuracy_gb = accuracy_score(y_test, y_pred)
precision_gb = precision_score(y_test, y_pred, average='weighted')
recall_gb = recall_score(y_test, y_pred, average='weighted')
f1_gb = f1_score(y_test, y_pred, average='weighted')

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy_gb)
```

Code 4 multi-classification code using Gradient Boosting classifier

```
from sklearn.neighbors import KNeighborsClassifier  # Import the KNN
classifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt

selected_features = ['Timestamp_numeric','Init Bwd Win Byts', 'Bwd Header
Len', 'Flow Pkts/s','Bwd Pkts/s']
X = data[selected_features]  # Features
y = data['Label']   # Target variable


start_time = time.time()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


knn_classifier = KNeighborsClassifier(n_neighbors=5)
knn_classifier.fit(X_train, y_train)
end_time = time.time()
execution_time_knn = end_time - start_time


y_pred = knn_classifier.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

## Assuming y_test and y_pred are arrays or lists representing the true
labels and predicted labels
accuracy_knn = accuracy_score(y_test, y_pred)
precision_knn = precision_score(y_test, y_pred, average='weighted')
recall_knn = recall_score(y_test, y_pred, average='weighted')
f1_knn = f1_score(y_test, y_pred, average='weighted')

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy_knn)
```

Code 5 KNN Code

Appendix (3) Features Explanation

Feature Name	Description
Flow Duration	Duration of the flow in microseconds
Total Fwd Packet	Total packets in the forward direction
Total Bwd Packets	Total packets in the backward direction
Total Length of Fwd Packet	Total size of packets in the forward direction
Total Length of Bwd Packet	Total size of packets in the backward direction
Fwd Packet Length Min	Minimum size of packets in the forward direction
Fwd Packet Length Max	Maximum size of packets in the forward direction
Fwd Packet Length Mean	Mean size of packets in the forward direction
Fwd Packet Length Std	Standard deviation of packet sizes in the forward direction
Bwd Packet Length Min	Minimum size of packets in the backward direction
Bwd Packet Length Max	Maximum size of packets in the backward direction
Bwd Packet Length Mean	Mean size of packets in the backward direction
Bwd Packet Length Std	Standard deviation of packet sizes in the backward direction
Flow Bytes/s	Number of flow bytes per second
Flow Packets/s	Number of flow packets per second
Flow IAT Mean	Mean time between two packets sent in the flow
Flow IAT Std	Standard deviation of time between two packets sent in the flow
Flow IAT Max	Maximum time between two packets sent in the flow
Flow IAT Min	Minimum time between two packets sent in the flow
Fwd IAT Min	Minimum time between two packets sent in the forward direction
Fwd IAT Max	Maximum time between two packets sent in the forward direction
Fwd IAT Mean	Mean time between two packets sent in the forward direction
Fwd IAT Std	Standard deviation of time between two packets sent in the forward direction
Fwd IAT Total	Total time between two packets sent in the forward direction
Bwd IAT Min	Minimum time between two packets sent in the backward direction
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Bwd IAT Mean	Mean time between two packets sent in the backward direction
Bwd IAT Std	Standard deviation of time between two packets sent in the backward direction
Bwd IAT Total	Total time between two packets sent in the backward direction

Feature Name	Description
Fwd PSH Flags	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
Fwd Header Length	Total bytes used for headers in the forward direction
Bwd Header Length	Total bytes used for headers in the backward direction
Fwd Packets/s	Number of forward packets per second
Bwd Packets/s	Number of backward packets per second
Packet Length Min	Minimum length of a packet
Packet Length Max	Maximum length of a packet
Packet Length Mean	Mean length of a packet
Packet Length Std	Standard deviation of packet length
Packet Length Variance	Variance of packet length
FIN Flag Count	Number of packets with FIN
SYN Flag Count	Number of packets with SYN
RST Flag Count	Number of packets with RST
PSH Flag Count	Number of packets with PUSH
ACK Flag Count	Number of packets with ACK
URG Flag Count	Number of packets with URG
CWR Flag Count	Number of packets with CWR
ECE Flag Count	Number of packets with ECE
Down/Up Ratio	Download and upload ratio
Average Packet Size	Average size of a packet
Fwd Segment Size Avg	Average segment size observed in the forward direction
Bwd Segment Size Avg	Average segment size observed in the backward direction
Fwd Bytes/Bulk Avg	Average number of bytes in bulk rate in the forward direction
Fwd Packet/Bulk Avg	Average number of packets in bulk rate in the forward direction
Fwd Bulk Rate Avg	Average bulk rate in the forward direction
Bwd Bytes/Bulk Avg	Average number of bytes in bulk rate in the backward direction
Bwd Packet/Bulk Avg	Average number of packets in bulk rate in the backward direction

Feature Name	Description
Bwd Bulk Rate Avg	Average bulk rate in the backward direction
Subflow Fwd Packets	Average number of packets in a subflow in the forward direction
Subflow Fwd Bytes	Average number of bytes in a subflow in the forward direction
Subflow Bwd Packets	Average number of packets in a subflow in the backward direction
Subflow Bwd Bytes	Average number of bytes in a subflow in the backward direction
Fwd Init Win Bytes	Total number of bytes sent in initial window in the forward direction
Bwd Init Win Bytes	Total number of bytes sent in initial window in the backward direction
Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
Fwd Seg Size Min	Minimum segment size observed in the forward direction
Active Min	Minimum time a flow was active before becoming idle
Active Mean	Mean time a flow was active before becoming idle
Active Max	Maximum time a flow was active before becoming idle
Active Std	Standard deviation of time a flow was active before becoming idle
Idle Min	Minimum time a flow was idle before becoming active
Idle Mean	Mean time a flow was idle before becoming active
Idle Max	Maximum time a flow was idle before becoming active
Idle Std	Standard deviation of time a flow was idle before becoming active

الملخص

أصبح استخدام الانترنت أمر يومي يمارسه الشخص بكل سهولة من خلال عدة وسائل متاحة مثل اجهزة الحاسوب المكتبية والمحمولة, الهواتف النقالة, و العديد من الاجهزة المنزلية. ادى هذا الانتشار الواسع الى ازدياد كبير بالهجمات الالكترونية المختلفة و لعل اشهرها و اكثرها خطورة هي فايروسات الفدية. تكمن خطورة هذه الفايروسات بالاهداف التي تسعى لمهاجمتها حيث انها لا تقتصر على الاشخاص العاديين, بل اصبحت تهاجم المستشفيات, المؤسسات الخاصة و الحكومية, الشركات, المؤسسات التعليمية, و العديد من الاهداف, و بالتالي أصبح هجوم فايروس الفدية أحد اخطر الهجمات السيبرانية خلال السنوات القليلة الماضية. اظهرت الدراسات الماضية ادوات فعالة الى حد ما, البعض منها اعتمد على تحليل سلوك الجهاز المصاب من خلال مراقبة بعض العمليات التي تستخدم من قبل فايروس الفدية و الاخر اعتمد على سلوك الشبكة من خلال مراقبة و تحليل بروتوكول معين يقوم فايروس الفدية باستخدامه. مع التغيير المستمر بالاستراتيجية المستخدمة من قبل هذا النوع من الفايروسات اصبحت عملية اكتشافه تحد كبير لخبراء امن المعلومات.

و قد ركز هذا البحث على تحليل معمق لسلوك فايروس الفدية عبر الشبكة، و استخدام التحليل المختلط لمجموعة من البروتوكولات المستخدمة بواسطة خوارزميات التعلم الآلي. تم تهيئة منصة اختبار مكونة من 3 اجهزة لاستخراج بيانات الشبكة لـــ 145 عينة تعود لاربعة عائلات مشهورة. تم استخدام هذه البيانات لاختبار ثلاث خوارزميات لاكتشاف فايروس الفدية, و قد حققت الخوازمية الافضل نجاح بدقة 88% لاكتشاف فايروس الفدية و تحديد نوعه بوقت زمني مقداره 6.3 ثانية.