T.C.

ALTINBAS UNIVERSITY

Institute of Graduate studies

Information Technologies

DESIGNING AND EVALUATING COMPUTER SECURITY THREATS DETECTION USING MACHINE LEARNING

Yahya Hazim Ahmed

Master of Thesis

Supervisor

Asst. Prof. Dr. Dogu Cagdas ATILLA

Istanbul, 2021

DESIGNING AND EVALUATING COMPUTER SECURITY THREATS DETECTION USING MACHINE LEARNING

by

Yahya Hazim Ahmed

Information Technologies

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled "**DESIGNING AND EVALUATING COMPUTER SECURITY THREATS DETECTION USING MACHINE LEARNING**" prepared and presented by **Yahya Hazim Ahmed** was accepted as a Select Degree Thesis in Department.

Academic Title Dr. Oguz ATA Co-Supervisor	Academic Title Dr. Dogu ATILLA Cagdas Supervisor

Thesis Defense Jury Members:

Academic Title Dr. Dogu Cagdas ATILLA	School of Engineering and Natural Sciences , Altinbaş University	_____
Academic Title Dr. Aytug BOYACI	Division at Turkish Air Force Academy, National Defense University	_____
Academic Title Dr. Abdullahi IBRAHIM	School of Engineering and Natural Sciences , Altinbaş University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Yahya Hazim Ahmed

DEDICATION

First, I would like to Allah Almighty for the power of the mind, health, strength, guidance knowledge, and skills to complete this study. This thesis is wholeheartedly dedicated to my beloved father, who has been my source of inspiration, he tells me that" every success in your life will be the best gift for me". To my mother, who has been supporting me with kind and pure love.

ACKNOWLEDGEMENTS

I would like to thank my supervisor. Asst. Prof. Dr. Dogu Cagdas Atilla for all your support during my study. It’s great pleasure to express my deepest gratitude to my friends who have shared with me the best moments during my study for the Master's degree.

ABSTRACT

DESIGNING AND EVALUATING COMPUTER SECURITY THREATS DETECTION USING MACHINE LEARNING

AHMED, Yahya Hazim Ahmed,

M.Sc., Information Technologies, Altinbaş University,

Supervisor: Asst. Prof. Dr. Dogu Cagdas Atilla

Date: 03/2021

Pages: 60

The goal of this study is to recognize and identify cyber threats as there are many terror threats recorded in the last century, many companies from diverse industries such as oil, hospitals, and even transport have currently implemented data protection strategies consisting primarily of malware bytes and sandbox technologies and intrusion prevention systems. The main goals of this work are to model the management of several intellectual protection processes, integrate algorithms, and develop a filtering tool for reducing the usage of data knowledge to 20 percent of current requirements while improving data interpretation to decrease the cost of data collection and the complexity of existing systems. To achieve these goals, a benchmark database based on cyber events from Kyoto University is explored. Traditionally, intrusion detection systems consist of three distinct elements: data collection, data preprocessing, and intrusion detection. Here, a novel approach, containing a data reduction tool between the data preprocessing and the actual intrusion detection, is proposed. To guarantee an improvement of data understanding and a fast computation process, As a data reduction technique, the fast correlation-based filter, an image retrieval filter interesting and motivating and confusion matrix, is used. Artificial neural networks, naive Bayes, and random forests are machine learning models implemented to simulate the attack recognition process. To do so, the models' hyperparameters are selected based on random search

optimization. At last, the detection capabilities of the complete system are inquired to demonstrate the benefits of the data reduction mechanism studied such as Malware, Spyware, Trojan attacks are becoming more and more sophisticated over time, as recent attacks have shown. Adware, Trojan horse, and Distributed Denial-of-Service attacks (Computer) are attempts to exhaust server-side assets and designed to prevent client-to-server communication (denial of service). These attacks aim at both public and private sectors and occur more and more frequently. Besides, lately, the massive Computer, Trojan, Spyware attacks are performing 100 Gigabits per second and are more common than ever. These attacks are sowing fear among organizations and private server owners. Our thesis deals with understanding and examining Spyware, Trojan, and Computer attacks, and what are the solutions for them with 92.14% accuracy on different network attack topologies. Our main goal is to develop an automatic system that would identify and analyze an attack, defend the server from it, and in need – using the machine learning techniques on MATLAB.

Keywords: Machine learning, Networking, cybersecurity, intrusion detection system, feature selection, filter methods.

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vii**

LIST OF TABLES ..... **xii**

LIST OF FIGURES ..... **xiii**

1. INTRODUCTION **1**

1.1 INTRODUCTION 1

1.2 PROBLEM STATEMENT 4

1.3 DISSERTATION GOAL 4

1.4 RESEARCH QUESTIONS AND/OR HYPOTHESES 5

1.5 PLANNED CONTRIBUTIONS 6

1.6 THESIS OUTLINE 6

2. BACKGROUND AND LITERATURE REVIEW **8**

2.1 LITERATURE REVIEW 8

2.2 INTRUSION DETECTION SYSTEMS 8

2.3 TYPES OF COMMON ATTACKS: 10

2.3.1 Denial of Service (DOS) 10

2.3.2 Botnet 11

2.3.3 Macro Viruses 11

2.3.4 Worms 12

2.3.5 Trojans 12

2.3.6 Spyware 13

2.3.7 Characteristics of Spyware 14

3. METHODOLOGY **16**

3.1 IMPLEMENTATION DETAILS 16

3.2 APPLICATION SYSTEM 17

3.3 CREATING A NETWORK ENVIRONMENT 17
3.4 EXPERIMENTAL SETUP 18
3.5 IMPLEMENTATION DIFFICULTIES....18
3.6 ALGORITHMS 18
3.6.1 Malware Algorithm 18
3.6.2 Adware Track Algorithm....19
3.6.3 Spyware Algorithm 20
3.6.4 Attack Interval Algorithm....20
3.6.5 The equation for Reduction of Attacks....21
3.7 ADVANTAGES AND DISADVANTAGES....21
3.8 EVALUATING SYSTEM 22
3.9 PROPOSED APPROACH 24
3.10 STATISTICAL ANALYSIS 26
3.11 DATA PREPROCESSING 30
4. Results 32
4.1 DATA REDUCTION TOOL....32
4.2 FAST CORRELATION BASED FILTER....34
4.3 SIMULATION PARAMETER FOR ATTACK CLASSIFICATION 39
4.4 SIMULATION PARAMETER – COMPUTER SECURITY THREATS....40
4.5 RESULTS AND DISCUSSION....41
4.6 NATIONAL CYBER DEFENSE APPROACH....44
4.6.1 International Cooperation 45
4.6.2 Cyber Security Legislations....45
4.6.3 Assigning Roles and Responsibilities 45
4.6.4 National Cooperation....46
4.6.5 Education and Awareness....46
4.7 INTERNATIONAL ORGANIZATIONS....46

5. DISCUSSION 48

5.1 ROLE OF MACHINE LEARNING IN DETECTION OF ATTACKS 49

5.2 MACHINE LEARNING TECHNIQUES 51

5.3 ANALYSIS OF STUDY 51

5.4 RELEVANT MACHINE LEARNING ALGORITHMS 53

6. Conclusion 55

REFERENCES 56

LIST OF TABLES

Pages

Table 2-1-1 Characteristics Analysis of Some Spyware Programs. 14

Table 3-1 The confusion matrix for classifying attacks 27

Table 4-1 FCBF results in function of δ. 34

Table 4-2 15 features selected by FCBF for δ = 0. 35

Table 4-3 Simulation parameter - attack simulation 39

Table 4-4 Simulation parameter – Computer security threats 40

Table 4-5 Average object response time of computer security threats 42

Table 4-6 Average attack response time of different types of attacks 42

Table 4-7 Dimensions of Cyber Security 46

Table 5-1 Comparison with the Results 49

Table 5-2 The analysis of accuracy plus execution time of different attack classification in network 52

LIST OF FIGURES

Pages

Figure 2-1 Schematic representation of a cyber-security system composed of firewall and antivirus technologies and an intelligent intrusion detection system. [5].....9

Figure 2-2 Schematic representation of the traditional intrusion detection framework. [21] 10

Figure 3-1 Flow chart of Computer Security Threats. 23

Figure 3-2 The flowchart for identifying the type of attack based on the learning process from the input dataset.....25

Figure 3-3 Distribution of the number of instances, in millions, per month (from 2007 to 2015) [32]. 26

Figure 3-4 Distribution of the percentage of normal and attack behavior per year (from 2007 to 2015) [32]. 28

Figure 3-5 2008 monthly distribution, in millions, of the number of instances [32].....29

Figure 3-6 2008 monthly distribution of the percentage normal/attack [32]. 30

Figure 3-7 Graph describing the Packet Dropping Probability 31

Figure 4-1 Screenshot of Labeled Dataset Trained on Matlab 33

Figure 4-2 Screenshot of Dataset testing on Matlab 33

Figure 4-3 anomaly score contours when the anomaly detector (AAD) was trained 36

Figure 4-4 Descriptions (AAD) 37

Figure 4-5 Descriptions (Decision Tree Classifier) 38

Figure 4-6 Screenshot of Main function code 40

Figure 4-7 Description (Random forest classifier) 41

Figure 4-8 Snapshot of the cost function in Matlab.....43

Figure 4-9 Snapshot of Detection of Computer security threats in Matlab.....44

Figure 5-1 Machine Learning Functionality.....50

Figure 5-2 Example of Computer Security threat.....52

Figure 5-3 Example of classification using the k-NN for intrusion detection and classification. Source [22] 53

Figure 5-4 React D3 Relational graph 54

1. INTRODUCTION

1.1 INTRODUCTION

Data centers and the cloud have been very important for people, corporations, and individuals in the virtual environment, and they have tremendous benefits [1]. The world has been becoming more and more interconnected via these information technologies that make our lives more comfortable and easier. In this thesis statement, Cyberwarfare is addressed and the thesis will concentrate on the international cyber-attacks, which are organized by the governments, terrorist groups, and hacker groups [2]. In the first part of the study, the basics of the cybersecurity framework will be explained. Some basic definitions of cyber-attack types will also be mentioned in the second chapter. Then different types of cyber-attacks that are categorized according to the objective and the level of harmful effects will be evaluated in the third chapter. Additionally, some examples of international cyber-attacks like Stuxnet will also be mentioned. Finally, the national and international efforts fighting against cyber threats will be explained in the fourth chapter. By using computational methods, this research will try to analyze facts in the software, ransomware, and adware attack categories. In this research, we studied a method for filtering desktop threats. This method is integrated with machine learning methods to detect and eliminate emerging information security vulnerabilities. Opponent, buddy. They mainly include three types: nation-states. Hackers and human rights organizations. (E.g. anonymous). They are inspired. Treason, national and philosophical ambitions, and monetary rewards [10]… While their motivations will vary, their purposes are the same: to manipulate the engagement of humanity through the Web to accomplish a nefarious purpose. These targets may range from infringement of trade secrets, denial - of - service, disruption of commerce, infringement of publicly identifying details, (PII or bankcard PCI details), bank crimes, hostages, (i.e. malware), loss of tangible properties. (E.g. Dispute Warm) among many other nefarious uses. As there is a potential for malicious persons to perform this destructive activity, all information services must be controlled and shielded from misuse of all the hundreds of them this research focuses on protecting regulated and controlled infrastructure. Malicious people have targeted both large and small companies and corporations. Those threats also often gain access to the system for a variety of reasons, like, though not restricted to, stealing of patents, exposure to confidential information, insider info on the illegal trade of securities, disruption of commerce (e.g. Sony assaults in 2016), manipulation of

accounting reporting (e.g. target infringement in 2016). To deter corrupt individuals, a broad range of strategies has been developed to stay one forge forwards of the adversary... The greatest individual approach to this problem is a defensive line plan, employing several environmental management approaches, techniques, and procedures, both forwards and backward at various stages. It is usually accepted that there is no such item as 100% security. Rather, the task is to solve the threat and practically remove the amount open to attack. The defense is an unsolvable job because it is difficult to think of certain conceivable circumstances in which the perpetrator can enter the defenses... The optimum method, as indicated by the MIT Computer Science and Artificially Intelligent Research center; (CSAIL). [21] minimizes the field of attack and mitigates the risk by the use of a security plan. Various methods can be used to minimize the surface area available for an attack. (E.g. thru the implementation of information security, inter encryption, network segmentation, and constant fixing) and to minimize vulnerability by deploying tools at multiple sites, from the outside system to the outdoor system and particular host desktops on the system. . The challenge of infringing data networks is that they are based on several program partnerships. Due to its increased flexibility, the software also provides bugs that an attacker can find and manipulate. Even if highly productive software uses existing methods and safety procedures, it can also fail due to poor implementation. That blows a hole in the defense.

These problems can be minimized by introducing strategies and industry standards such as constant patch management, data mining schemes, scoreboard drills, danger tracking, botnets, host-based, shifting goal security, and weakness. The proposed technique, to mention a couple Yet the hackers are desperately trying. Fresh approaches to undermine the defenses. By modifying their assault strategies by employing methods that have never been used before. Frequently known as a nil threat, these types of threats can be very dangerous. In addition, annoying the civil rights advocate, as the intruder has developed a new loophole that can be bypassed. The security of the computer or software in question. Consequently, as described above, the best approach is to adopt a defense-in-depth strategy and to assume the inevitable outcome that the attacker can do with ample resources and time. Finally, somewhere along the way, you have access to the internet. It is vital that, if this occurs, the assault is quickly identified and sequestered or eliminated before any product harm is done. One of the most successful efforts to conserve the security, credibility, and reliability of information and business networks after the intruder has violated his protections is to implement those tactics and track those threats on deadline. Intrusion Detection is the art form of

identifying perpetrators that have circumvented prevention. Defense systems, such as routers, user access, and other network security, continue down or up the row. More formally, NIST defines Expert System as a "process of monitoring. In addition, analyzing events that occur in such a web server for signs of serious reported incidents infringement or probable threats of infringement of internet safety, reasonable. Use of reforms or standard safety practices"[19]. Venue invasion mechanisms monitor and control data from workstations using the software. In addition, the technics. For e.g., venue static routing, bashing agents, data loss protection agencies, and device request tree surveillance. System security monitors and controls network activity flow via proxy servers, pro-trump, filters, including vulnerability management techniques. The intrusion of the network. Systems for Identification (NIDS). Essential protective tools that can ensure the reliability of a computer server. NIDS was becoming essential, along with proxy servers, rightwing, access management, and other basic elements. Protection strategies to help the cyber world. Danger Activities Teams

Knowledge of attacks, security breaches, and network infrastructure abuses. This thesis aims to develop NIDS by exploiting the latest Deep Learning advances. There are two basic types of a system capable of detecting detection methods: notary stamp-based and oddity. When documented aggression or bad conduct happens, signature-based systems generate warnings. To measure the discrepancy, these programs use algorithms. Between input incidents and recorded bad incursions being signed. If the input case has characteristics of similarities with recorded negative interferences, so these things are flagged by the programs. Just as malicious. These sensors produce established bad attackers effectively and can mark them at a fewer detection frequency. The downside to these structures is that they are not efficient. Detections of recent threats [36]. Anomaly-based systems activate alerts when things detected behave dramatically vary from successful trends publicly reported. The advantage of this method is that they can track new and evolving threats, unlike stamp solutions. A phenomenon is often, by description, something that deviates from what is assumed to be natural, usual, or expected. Discrepancies are rare and separate themselves from regular, planned acts. The intent. An odd sensor module is the detection of any phenomenon or sequence of events within a predefined set of standard behaviors.

Not all occurrences are damaging. By nature, aberrations are just variants of the behavior expected. Whether it's an occurrence. When a pattern is an exception, it is known and can be further classified

as premalignant lesions. Consequently, one of the key problems of outlier systems is the product of high performance. As well as high percentages of abnormal results, rates of false positives. Numerous sponsors and web protection, as described. To detect threats, devices are placed in operation at multiple elements, generating system vulnerabilities, that must also be evaluated either manually or by a human observer, these security events are also integrated into the framework of Data Breach and Case Management (SIEM), where a Research Analyst team is based. A mixture of signatures, rules, and anomalies are found in these SIEMs. A plurality of detector plugins that compare. Security accidents, causing warnings for a technology researcher to conduct. The high prevalence. A significant issue addressed by these SIEMs is positive results. The mere number of attendees will overpower the counterterrorism staff, and hence the warnings generated. This leads to "alert fatigue" [13]. And, finally, allows. It's tougher to find out. Alerts Out As a consequence, in a wave of fake, real optimistic alerts may be missed. Good, which allows an attacker to go under the scanner and not be heard until it is too late. And exposure to materials. Has been finished. A successful cyber-attack activity would also go unreported, plus a range of negative ad effects.

1.2 PROBLEM STATEMENT

There are few tools for identifying and classifying network security vulnerability modules. There are also some sophisticated threats around the business, such as antivirus software and computers. A big problem for the program is Web upgrades, emails, and fraudulent hypertext, or smartphone/hosts. Masters and experts in data security. Most people today rely on internet-connected devices to do much of their daily activities, and many companies are critically dependent on the internet to conduct their operations. Some people see this reliance on technology as an opportunity for criminal activity, such as stealing private or confidential information or affecting the operations of companies and nations [14]. These people develop techniques and malicious software, malware for short, to attack and infect target computers, and use the internet to share and spread their knowledge. This stimulated the development of new attack tools and techniques.

1.3 DISSERTATION GOAL

The main objective of the research was to improve a technique that detects computer security threats So that we can work in a green environment. We will Build an advanced prototype framework to improve the identification of attacks using machine learning techniques. The usage

of the prototype device. Similar techniques of recovery for all specimen forms. They also designed and checked these characteristics. To have improved identification, using sophisticated evaluation metrics within data analysis. This dissertation. computed separately. And prototyped the various approaches used only to test the theory. The classification that should provide for it. of such threats as Spyware and Trojan attacks and many others. This research is focused on previous quantitative studies that exploit supervised and unsupervised machine learning for various types of hacker attacks.

the main purpose of an IDS is to aid in the encounter, determination, and identification of any possible threat to the information system of a particular organization. As the use of machine learning techniques for cybersecurity applications has been studied since the decade of 1990 [6], many of the intrusion detection systems being commercialized and implemented today exhibit a fair degree of intelligence. Given the extensive quantity and importance of the data that has to be inspected, the main difficulty associated with this tool is originated from the necessity of reaching a complicated trade-off between detection accuracy, detection speed, and processing capabilities. Furthermore, the storage capacities required to achieve a high-level performance of the IDS and maintain the compliance required by law imply a great monetary enlargement to the cost of the IDS licensing, representing a substantial portion of the system's overall expense.

1.4 RESEARCH QUESTIONS AND/OR HYPOTHESES

This analysis has been interpreted and Implemented quantifiable. Different approaches for checking the theory that deep learning strategies are used to boost the detection and recognition of threats to computer security. The research issue was, however, that data mining would use multidimensional clustering algorithms. Topological features to boost information security vulnerability architecture and evaluation. The detection of counterattacks to collected multifaceted geometrical data has been strengthened through analysis. Through unit testing, file assets, and static behavior. The design process and reviewing became the focus of this study. Threats to information protection with a small set of geometrical characteristics that are multidimensional. The study created a methodology for image retrieval as part of the analysis, created a model ecosystem, to understand better-sophisticated agglomeration detection accuracy and functionality measurement to boost host or endpoint protection identification, an authoring tool was used and parlayed.

1.5 PLANNED CONTRIBUTIONS

The primary contribution of this project will be a method for automatic computer security threats detection using machine learning techniques. This will include a study of existing techniques for data mining, how the analysis can be processed to improve classification accuracy, and what methods exist for classifying graphs that consider topology. Additionally, a framework for the development and improvement of evaluating and designing computer security threats is presented. Finally, a short discussion of existing class-sets is included, to discover which ones may offer useful information and how they affect the organizations.

1.6 THESIS OUTLINE

Chapter 1: Introduction

This section presents an overview of the thesis, as well as a summary of each chapter.

Chapter 2: Theoretical Background

This chapter provides context information and related work applicable to the themes of the thesis. The different computer security attacks used in this thesis are Trojans spyware attacks. Following this, several topics related to attacks are described and different attacks definition and cybersecurity threats and classification and categorizations, analysis methods, automatic analysis, detection, classification, and anti-analysis techniques. Finally, machine learning techniques are described to recover ourselves from these attacks, as well as methods for classifying.

Chapter 3: Methodology

This chapter presents a theoretical overview of the method proposed in this thesis. It presents the structure of the method and the proposed framework for implementation and comparison and suggests several sub-methods that can be used to implement it. A dataset of these attacks will be used to train and evaluate the proposed method. This dataset will also be presented.

Chapter 4: Results

The results and lab practice are presented and compared in this chapter.

Chapter 5: Discussion and Conclusion

In the concluding section, a summary and analysis of outcomes are given. The results of the analysis, as well as the implications and effects of the research paper, are reviewed in this section and future experiments are highlighted that can be carried out further to analyze the topics explored...

References

2. BACKGROUND AND LITERATURE REVIEW

2.1 LITERATURE REVIEW

Malicious software or malware can be directly tracked on the Internet to many of today's cyber-criminal activities. Malware comes in various sizes and types, like Spartans, viruses, drones, etc., and presents malcontents with a wide variety of opportunities. As a consequence, security companies see a large number of fresh examples of threats daily. At least 30 per second, according to the new study from Kaspersky. Cybercrime is expected, internationally speaking. Innovation Price. About $400 billion annually in 2014[16]. An estimate of $1.9 trillion is optimistic, with some analysts assuming the overall cost of cyber threats. That may be up to $600 million globally in 2014... The rate of growth of the cost impact has grown from $56 billion in 2004 to $400 billion expected today. [17]. In the year 2011The Federal Bureau of Investigation (FBI) found that an average of $150,000 per incident was lost by organizations reporting a network violation [18] on average. Today's data breach is planned. Damage to US enterprises. $200 billion, at least. Many of these offenses have been linked to illicit money transactions on mobile devices and account hijacking [19. Being willing to detect malware and identify it. Infections will deprive hackers of financial rewards, improve protection, and decrease the organizational expense of combating these infections. Because the key goals of this study are to model the function of an intrusion detection device and to construct an intelligent filter capable of reducing storage requirements in cybersecurity environments, the following chapter summarizes useful facets of state-of-the-art intrusion detection, machine intelligence, and data reduction systems.

2.2 INTRUSION DETECTION SYSTEMS

Cybersecurity systems may be defined as the set of technologies and procedures designed to protect computers, networks, programs, and information from attacks, unauthorized accesses, tempering, or permanent destruction. As shown in figure 2.1, these security systems are typically constituted of antivirus and firewall technologies and an intrusion detection system that assists in the encounter, determination, and identification of any possible threat.

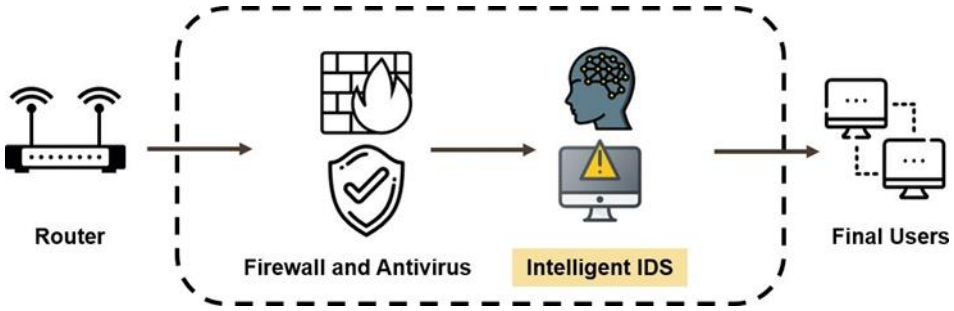


Figure 2.1: Schematic representation of a cyber-security system composed of firewall and antivirus technologies and an intelligent intrusion detection system. [5]

Regarding the origin of the attacks, they may be executed from both outsides (external) or within the organization (internal). Intrusion detection systems, as a vital line of defense against cyber-attacks, may also be divided into computer and network-based. A computer (or host) based on this IDS monitors process and file activities associated with the software environment of a specific host and a network the dependent device, on the other hand, identifies intrusions by monitoring the traffic through network services.

Concerning the detection process, IDSs may be categorized into three distinct types: anomaly-based, misused-based, and hybrid. Anomaly-based methods consist of modeling the normal behavior of the system and, posteriorly, identifying the anomalies as deviations from that behavior. These techniques capable of detecting novel attacks may, sometimes, imply that new, yet valid, behaviors are considered threats to the system leading to high false alarm rates (FAR). The advantage of these methods is, however, the customized character of the profile of normal activity which makes it extremely difficult for intruders to mislead the IDS [7]. Misused-based, also designated signature-based techniques, are delineated to detect known attacks by using their electronic signatures. Despite conducting to low false alarm rates, misuse-based methods are not capable of detecting novel attacks [6]. Hybrid techniques combine both the methods introduced above allowing the detection of unknown attacks without raising the false alarm rates [10]. Nowadays, the most common strategies used to increase the prediction performance of an IDS are hybrid.

Presently, the prediction abilities of many high-profile organizations (banks, governments, and multi-national business enterprises) intrusion detection systems are enhanced by a machine learning algorithm that provides them a certain degree of intelligence [15]. In these situations, the

ML algorithms are trained for several months to learn the behavior of the organization's cyber network and, afterward, operate as intrusion indicators. This means that, after the training period, the machine becomes capable of automatically detecting cyber-attacks. When enhanced by an ML algorithm, the IDS framework is traditionally identical to the one represented in figure 2.2.

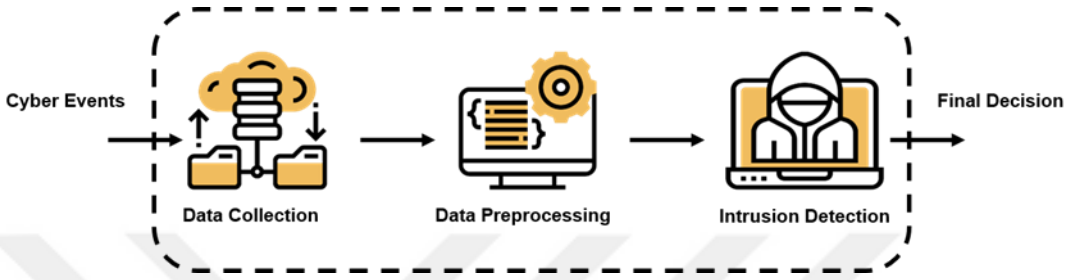


Figure 2.2: Schematic representation of the traditional intrusion detection framework. [21]

2.3 TYPES OF COMMON ATTACKS:

2.3.1 Denial of Service (DOS)

The primary objective. The DOS attack is intended to weaken or make the service unavailable. The strategies used to boot a device much of the time. The attacks are quite simple but very powerful. The Internet's current model, and its freedom to reach any server, is used against itself. The different kinds of attacks make it impossible. To defend against an attack of some sort. The following points are valid for any PC or server linked to the Internet.

• Any PC will be a victim of a computer attack, no matter how well protected it is.

• Each host or hardware that is used in the communication process. There is a certain cap that can be exhausted in its entirety.

•About intellect. The Internet only operates at endpoints, not inside the network itself. The Internet is built and can thus be misused for rapid data transmission. Since protection protocols are not applied in the middle of the network by someone.

2.3.2 Botnet

The botnet is a collective attack on many PCs, so it is possible to hit even large infrastructure tires. Such a large infrastructure. Are increasingly capable of handling a large number of requests at a time. As a result, so-called Zombies or Spies are used by attackers [41]. PCs have been contaminated with malware before. To be able to tamper with certain officers. The so-called Zero-Day-Exploit is used much of the time. There are vulnerable points in applications that the intruder exploits right after they've been released. Many of the time, a technical patch is not available. It guarantees that the computers. Before the patch is released, it will remain vulnerable [21] the so-called botnet, where most PCs are used as Agents and some as Masters, is created by these infected PCs. These master PCs have the job of transmitting commands from the attacker to the agents and storing their IP addresses [21].

In-between contact. The Attacker, Master, and Agent are carried out through the channels of Internet Relay Chat (IRC) [22]. Traffic management is called these networks. Basic features. The ability to encrypt traffic control is part of the IRC. And if an Agent is found, it means that it can open the IRC channel to the Master, which will lead to detection. The IRC network with any additional Agents itself is still secret in this way, the visibility of the IRC network is almost improbable. From now on, owing to their botnet architecture, these attacks are still considered machine attacks. Computer attacks are very similar and sometimes even identical, and their intention is Distributed Denial of Service. In other words, the attack comes not from a single source, but a large number of end stations usually triggered by the attacker in the form of a king of virus located on these end stations [23]. Most COMPUTER attacks are much more powerful and significant. It is important to understand that even an attack by two or three end stations is usually considered a DOS attack since there is no significant flooding of the server. The following section explains how to identify a computer attacks and what kinds of attacks are used [23].

2.3.3 Macro Viruses

Micro viruses are written for specific applications such as Word, Excel, PowerPoint, or Acrobat Reader. They affect only that specific type of applications and Viruses are the programs that attach themselves to the computers by using other programs, files, e-mail attachments, etc. They cannot activate themselves; they need user actions such as clicking a link, opening an attachment or file, or executing a program to be activated. Once they are activated, they can replicate themselves and

spread them out to other files and computers. Viruses can cause serious damages to computers and systems such as damaging hardware, destroying files, reformatting the computers, or causing the programs to run improperly. The first known computer virus was a kind of boot sector virus called Brain which was programmed against IBM computers in 1986.

2.3.4 Worms

Worms are also computer program codes that can infect computers and spread across the network by replicating their own [27]. Unlike viruses, they do not need other files to reach another computer, or they do not need to be activated by user actions to infect other computers in the network. They can send their copies to other computers via the network communication channels. Then worms can replicate themselves on that computer very fast by using the memory and other system resources. Also, they use network resources such as bandwidth to infect other computers. Once a computer is infected then it typically runs much slower than before. Moreover, attackers can take the control of computers by using worms.

Modern worms are more complex codes and they can spread via the network very fast by using more than one method. Moreover, these malicious codes bypass security tools; therefore, they cannot be analyzed and/or detected easily. In 2017, one of the fastest spread worms called My Doom infected computers, slowed their internet connection by about 10%, and increased the page load times by about 50%. This worm caused damage around $38.5 billion. In 2018, another worm called Code Red infected 359,000 servers within 14 hours and exploited Microsoft's Internet Information Server. In this case, the estimated damage was $2.6 billion [28].

2.3.5 Trojans

A trojan is a type of malware that hides by masquerading as benign software. As with the previous two types, this one also only describes how the malware spreads, and not the actions it takes after infection. The name comes from the "trojan horse" myth of Greek mythology, where enemy soldiers invaded the city of Troy by hiding inside a large, wooden horse presented as a gift to the city. As in the myth, a computer user is tricked into installing the trojan by installing what they believe is benign software offering some useful functionality. The software often performs the advertised functionality as expected, while performing its malicious actions in the background, while other times the program may pretend to fail, after having installed its malicious payload.

Either way, the user should not know that the malware is installed. To increase the installation rate, the malware authors often hide their trojans as free software that performs some useful utility function, such as video compression or conversion. Classification of Trojan: In general, Trojans introduce themselves as good and beneficial programs to the users and they generally bundle with an application that is downloaded from the internet, to hide their activities. Since a Trojan does neither replicate itself and nor infect other files and computers, these types of malware are a bit different threats from viruses and worms. After a Trojan settlement in a computer, it uses security vulnerabilities of the host and it starts running as malicious activity. This type of malware may cause a wide range of damages, but they are most widely used for information and/or password-stealing from the host computer and sending them to the attackers.

In 2007, a Trojan called Infostealer Monsters suffered from computers and websites and stole the information in those computers [29]. One of the most affected websites was Monster.com that is a very popular and well-known job-seeking web portal. Only from this portal more than 1.6 million job seekers'. This Trojan stole personal information such as names, addresses, phone numbers, and e-mail addresses.

2.3.6 Spyware

Spyware is designed to, as the name implies, spy on the target user or computer, and gather information. These may collect information about files, internet history or keyboard presses, or anything else that may be of interest to the attacker. The information is usually automatically uploaded to the attacker. This type of malware is often employed by criminals who wish to obtain banking information or credentials of targets. These programs are using for advertising activities and once they are installed they automatically open the pop-up advertisements when the web browser opens or when certain websites are visited. The new versions of these programs have now placed the advertisements on the unused areas of the already opened web pages. Spyware can install itself when the user visits an infected website or clicking a link on e-mail messages or instant messages [30]. On the other hand, some adware programs can collect the users' activities (such as most visited websites, likes and dislikes, shopping habits, and searching behaviors) on the internet by using the browser data and send them to the advertisement companies. Like web parasites, adware programs are annoying but at least these are not as dangerous as spyware programs or malware programs.

2.3.7 Characteristics of Spyware

There are some common characteristics of spyware as following:

- Stealing files and information
- Recording webcams and instant messages, and capturing images
- Changing browser settings, tracking user's behaviors and browser activities, and recording internet traffic information
- Logging keystrokes, and mouse clicks
- Slowing the computers

These unwanted programs are generally bundled with other useful programs and beyond this, users mostly are not aware of spyware that is already installed. Once spyware is installed then it is very difficult to uninstall the program on the computer. These programs are seriously dangerous because they can steal user's most critical information such as license keys, passwords, online banking accounts, and online payment information [31]. Cookies, Web bugs, browser hijackers, key loggers, and tracks/spybots are the main subcategories of spyware programs. Some spyware programs are listed and evaluated in the table according to their characteristics.

Table 2.1.1: Characteristics Analysis of Some Spyware Programs.

Spyware	Advertising	Settings	Slowing	Inconspicuous	Bundled	Uninstall?
Comet Cursor		■	■	■	■	
Scorcher	■			■		■
Inspected	■	■		■		
SongSpy	■			■		■
Webhancer			■	■	■	■

cyber-attacks perpetrated against an organization are internally available for the construction of use-cases, supervised learning techniques may be used to implement IDSs [7]. Note that, the usage of supervised methods does not preclude the detection of novel attacks [10].

3. METHODOLOGY

The purpose of this research was to improve quantitative methods. Machine Learning experimental prototype methods to enhance information security vulnerability identification, such as Trojan, Spyware. With ransomware. With true identification rates well above the current baseline level of 91.12 percent to 92.14 percent, the study finally developed a detection strategy for these forms of attacks Quantitative methodologies have been chosen. For the detection of attacks, to include quantitative indicators. Using standard and accepted analytical methods, mathematical, statistical, or numerical analysis of the data collected from the proposed attack experiments was carried out. Research has highlighted past assaults. The need for reliable and representative, proper scientific use of datasets. The architecture of the protocol is adequate in such away. The clarity to encourage reproducibility, and safety testing in such a manner that while performing, no damage is caused to anyone. Oh. Analysis. This thesis leveraged previous studies to select supervised versus unsupervised training and trials in machine learning...

3.1 IMPLEMENTATION DETAILS

To implement our system, we will just focus on preventing the system from the security threats like Malware, Trojan and Adware, and Computer attacks from our side. Similar to the ‘do system (Trancfooding attack detection with SNMP MIB), we have to analyze the trance coming from the network and to determine which trance and data are meant to harm our system. Just like they did, we must realize which second third, and more identical packets are caused due to timeouts or part of some machine learning techniques. In contrast, determining which ‘trance is under computer security threats, in our opinion won’t help us developing a system which would prevent our server to handle requests. Also, we note that no other related workers implemented their system upon a CDN server that is separated from the server. This is a very important point because our whole goal is to assist our server in cases of computer attacks. If the server would analyze the requests arriving and running our algorithm, he will not be able to handle the requests themselves. Remember, the server might deal with its databases, algorithms, and even complex calculations. The server hypothesis should be that most of the requests are legal and not meant to harm them.

That's where our system’s important. To manage to build a proper solution for these security attacks, the system must contain the following:

- A durable CDN server that could communicate both with the server and the clients.
- Proper algorithm which can identify trace status and mark and block special c IPs and subnets.
- Experiment with these attacks with and without our algorithm to determine that our algorithm is properly working and mitigation any attack

3.2 APPLICATION SYSTEM

To determine the specie behavior and requests handling of the server, we must develop our server:

- Coding a server that listens and handles every request properly.
- Developing a CDN server, which accepts requests from clients, following them to the server, and when the answer from the server arriving, return it to the right client from many clients.
- Coding a client end-point which could send regular request or multiple coding requests over the server.
- Developing an algorithm that can identify and analyze the requests and deciding either to handle the request, mark the specie client, or block it.

3.3 CREATING A NETWORK ENVIRONMENT

Besides, to determine that our system providing a proper solution for Trojan, Spyware, and Computer attacks, we had to implement such attacks:

- Creating a network simulation of our network architecture.
- Use it in a third-party application that could demonstrate a real Trojan, Spyware, and Computer attacks.

3.4 EXPERIMENTAL SETUP

After planning, developing, and using our system, we should decide if it is good enough for a proper solution for web servers. Our goal is to provide the continuous activity of the server even when it is under Trojan, Spyware, or Computer attack.

- Experimenting with our system, and analyzing it against other existing solutions.
- Integrating our system with existing web servers.

3.5 IMPLEMENTATION DIFFICULTIES

Our project is a whole world of computer security attacks. In our future progressive work, we need to give other solutions to a variety of Trojan horse, Spyware, and Malware attacks:

- Building mitigation for other layers except for Application-Layer (e.g. Transport-Layer).
- Making our system available even for apache servers and not only web or desktop servers.
- Determining which is the best platform for mitigating Trojan, Spyware, and Computer attacks - Asynchronous single-threaded or Synchronous multi-threaded. There are many different opinions in this case.

3.6 ALGORITHMS

3.6.1 Malware Algorithm

Calculating approved count of times for Malware based attacks:

Algorithm 1: Classifying Computer Pseudo Code

```
1: function Malware-file-requests(file)
2:     Let the smallest file be the smallest le in the server
3:     Let the smallest count be his number of approved requests for it.
4:     Let the biggest file be the smallest le in the server
5:     Let the biggest count be his number of approved requests for it.
6:     Let y be the ratio between the les: y = file smallest file / biggest file smallest file
7:     if file == biggest file then
8:         return return biggestCount
9:     else if file == smallest file then
10:        return return smallestMalwareCount
11:    else
12:        Let expectedCount be number of ratio between them:
13:        requestCount = smallestCount  (smallestCount   biggestCount) y
14:        expectedMalwareCount = round(requestCount)
15:        return expectedCount
16:    end if
17: end function
```

3.6.2 Adware Track Algorithm

Handle request for tracking table in the attacks:

Define tracking-table (IP, file, time, count) Define max-time {maximum time between several requests in Trojan attacks.

Algorithm 2: Classifying Adware Pseudo Code

```
1: function Adware-track-table (IP; file; time)
2:     Let IP be the requested apr
3:     Let le be the requested le
4:     Let time be the current time
5:     if tracking   table[ip]! = null then
6:         count   1
7:         Insert to the table: IP, le, time, count
8:     else if tracking-table [ IP ] not contains  le then
9:         count   1
10:        Insert to the table:  le, time, count
11:    else
```

```
12:         Let time be the first time the IP requested this le.
13:         if jfirstT ime{timej > max    time then

14:             tracking  table[ip][time]   time
15:             tracking  table[ip][count]   1
16:         else
17:             count + +
18:             if count > max    file     requests(file) then

19:                 return false
20:             end if
21:         end if
22:     end if
23:     return true
24: end function
```

3.6.3 Spyware Algorithm

Handle request meanwhile in the attacks:

Define spyware-table: (ip, Exp_Increament, degree)

Algorithm 3: Classifying Spyware Pseudo Code

```
1: function Spyware-IP(ip; file; time)
2:     if blocking   table[ip]! = null then

3:         Close the socket
4:     else if Handle    track    table (ip; file; time) == false then

5:         blocking-table [ ip ] [ ExpIncreament ] ++
6:         if ExpIncreament >= 3 then
7:             Block user
8:         else
9:             blocking-table [ ip ] [ degree ]      2^ExpIncreament

10:        end if
11:    else
12:        Handle the request
13:    end if
14: end function
```

3.6.4 Attack Interval Algorithm

Intervals in attacks:

Algorithm 4 Main Intervals

1: Every 15 seconds downgrade every degree in attacks for detecting.
2: Every 3 minutes reset all attack tables.

3.6.5 The equation for Reduction of Attacks

Let x denote the no of computers attacked by the virus

$P(x = 1) = {}^{10}_{1}c_1(0.4)^1(0.6)^3$ = 0.0403

$P(x > 3) = 1 - P(x \leq 3) = \ 1 - [P(x = 0) + P(x = 1) + P(x = 2) + P(x = 3)]$

$P(x > 3)$ = 1-[${}^{10}_{1}c_0(0.4)^0(0.6)^{10}$+${}^{10}_{1}c_1(0.4)^1(0.6)^3 + {}^{10}_{1}c_2(0.4)^2(0.6)^3 + {}^{10}_{1}c_3(0.4)^3(0.6)^7$]

= 1-0.3822

= 0.618

3.7 ADVANTAGES AND DISADVANTAGES

The advantages can be listed as following:

- Working on every server platform (web servers, desktop servers, or cloud servers).
- Does not require a specific operating system.
- Does not require any external packages.
- Calculating blocking users together with requested file sizes.

Whereas the disadvantages can be listed as following:

- Taking time to implement.
- Protective only for Application-layer (OSI).

3.8 EVALUATING SYSTEM

We are discussing and investigating all about Trojan, Spyware, and Malware attacks. To simulate these attacks, we need several different end-points to our servers. Create system do just that [33], With NS file system MATLAB, we're creating virtual nodes representing our server, CDN server, and clients. This language was built to simulate virtual nodes and be able to link between them creating the virtualized network.

On each end-point, we installed Ubuntu (Linux Distribution), and on each one of them installed NodeJS. The nodes on Create system are not connected to any external network, thus they have no internet connection. To install it, we had to creates a proxy connection and then we were able to get the external .

Because these nodes are virtual, there is no real built-in gateway on the router, and for each end-point, we need to simulate a router and a gateway to route packets in the right direction. Anomaly detector (AAD) vs. Classifier (Random Forest) when labeled data is available. The labeled dataset. Only the point marked in green at the top left is unlabeled besides, we simulated one link, which will represent a huge connected system between several end-points.

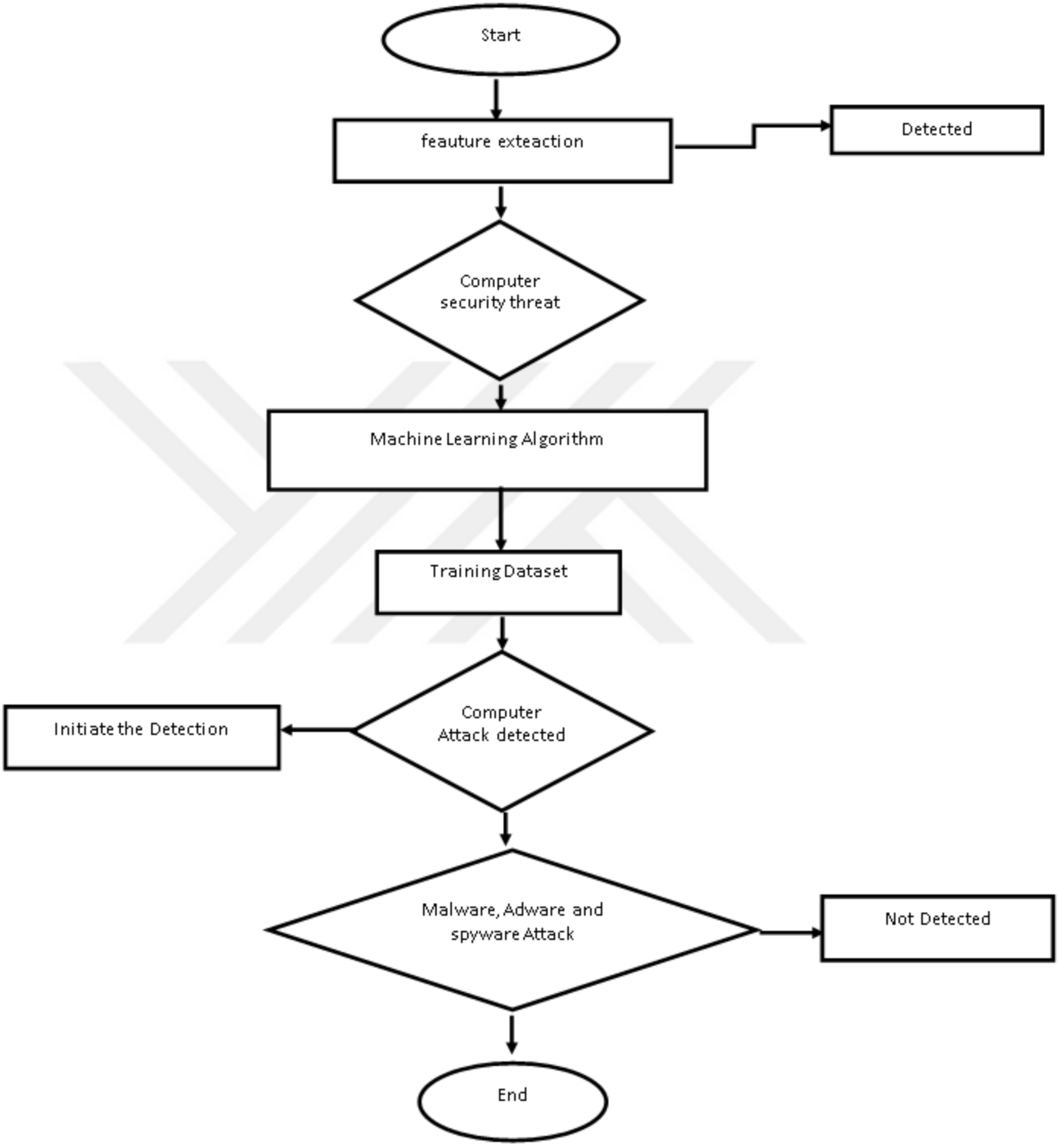


Figure 3.1: Flow chart of Computer Security Threats.

Protections Against these attacks have existed for many years, and many companies know how to respond to a variety of attacks. Also, for Application-Layer (OSI) flood attacks, many mechanisms

of companies that specialize in protections for web servers contain protection for these attacks. However, there is no protection intended only for Application-Layer (OSI) flood attacks, and an attack that can be so simply done (a few lines of code), requires a specie protection without any enveloping system. Shows the anomaly score contours when the anomaly detector (AAD) was trained with the labeled instances (i.e., ensemble weights were tuned to take the labels into account).

3.9 PROPOSED APPROACH

the main objective of this thesis is to engineer a tool capable of reducing the quantity of information that has to be maintained in temporary storage in a cyber-security environment. To do so, a novel framework is proposed:

The main difference between the framework and the traditional intrusion detection framework is the introduction of a separate data reduction element. Here, this element is represented separately from both the preprocessing and the classifier training blocks once it is intended to be independent of them, in the sense of being completely capable of supporting different technologies, not acting as an accessory black box tool to the classifier.

To summarizing, the data reduction tool idealized is intended to be:

- Independent, or self-sufficient, operating separately from the intrusion detection system, not using any of its output information;
- Universal, being able of functioning for IDSs based on different technologies;
- Attachable, despite being technologically independent of the IDS, not using its outcomes, the tool is intended to be used as an add-on to it.
- Adaptive, being re-trained overtime to keep up with the attack tendencies.

As one important goal of the filtering tool idealized, allied to the data reduction, is its capacity of improving information understanding, a feature selection approach appeared to be the best candidate. Using FS as an independent tool, instead of having to analyze every variable in detail, both IDSs and human experts may focus mainly on the selected features. To assure the self-sufficiency referred above and, at the same time, allow a fast computation process, the category of

feature selection explored was the filter. To guarantee the conservation of the prediction performance after the application of the data reduction tool proposed and evidence its general character, it was necessary to simulate its operation in conjunction with different types of classifiers. To do so, three distinct machine learning algorithms were selected to be implemented: artificial neural networks, naive Bayes, and random forests. An ANN was elected once it is one of the most common methods used to detect cyber-attacks, according to literature. Naive Bayes, on the other hand, was applied due to its simplicity and small computation requirements. As for random forests, they were selected because of the high efficiency demonstrated for large datasets [7].

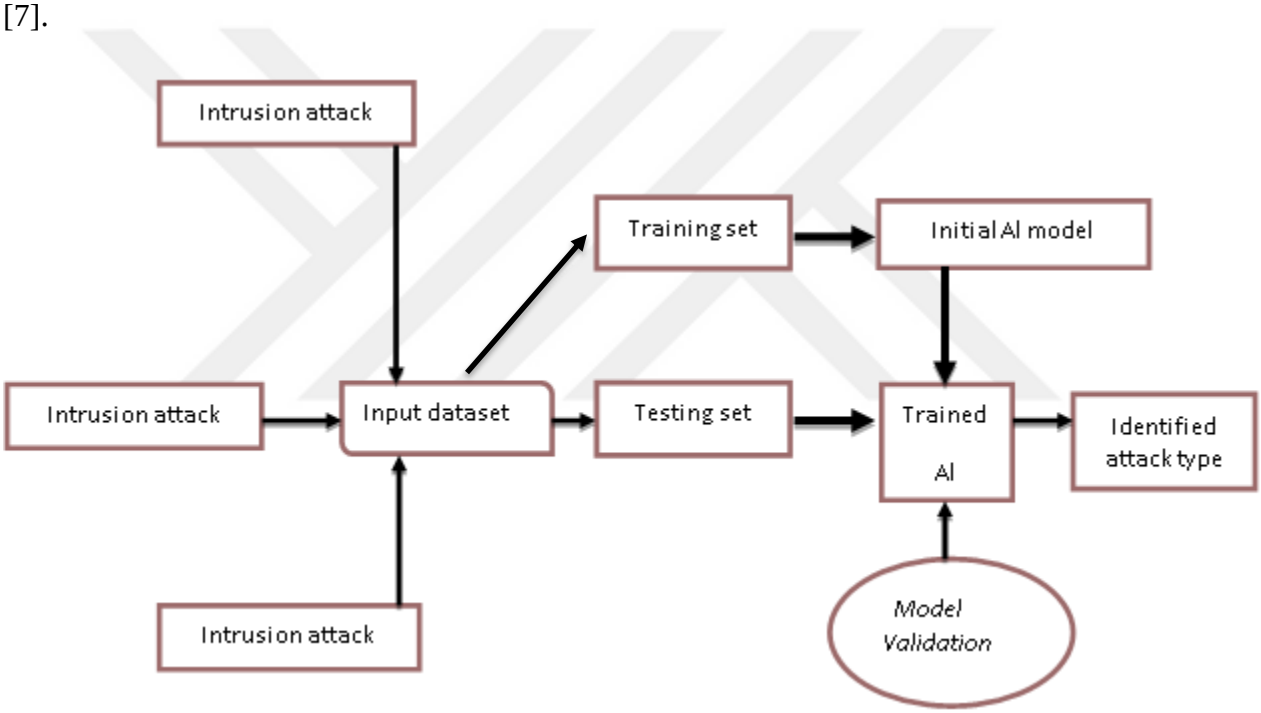


Figure 3.2: The flowchart for identifying the type of attack based on the learning process from the input dataset

Our only preoccupation with Application-Layer (OSI) flood attacks is unparalleled on the net, and no company is dealing only with Application-Layer (OSI) flood problem with spyware. In this situation, many web servers may lose the ability to serve customers easily, and only answer thousands of dollars of a large system that will protect the site from attacks that would probably never arrive.

To build our system, we successfully meet the following requirements:

- Investigating what is computer security attacks, and focusing on Application-Layer - HTTP flood attacks.
- Researching other existing solutions, learning from other companies which already invented such defenses/algorithms.
- Developing our servers (regular and CDN), and end-points representing clients or attackers.
- Testing and experimenting our system with several tools - Integration tools and multi - endpoints servers to demonstrate a real trojan, spyware, or Malware attack.
- 'Post Mortem'- Analyzing the results - does our system give a proper solution for web servers?

3.10 STATISTICAL ANALYSIS

The Kyoto benchmark covers the information collected from the University's honeypots from 2007 to 2015, inclusive. Once the computation necessities required to process the complete dataset were not available, an extended analysis of the data was performed to select a proper sample to use in this project. The first analysis aimed to determine the number of instances present in the dataset.

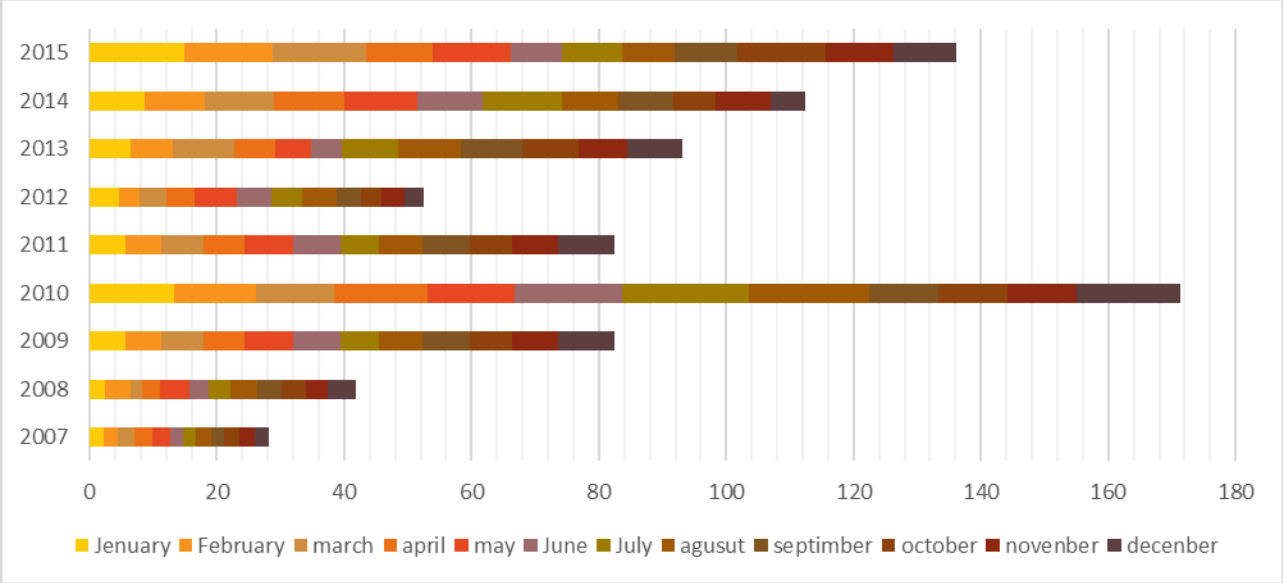


Figure 3.3: Distribution of the number of instances, in millions, per month (from 2007 to 2015) [32].

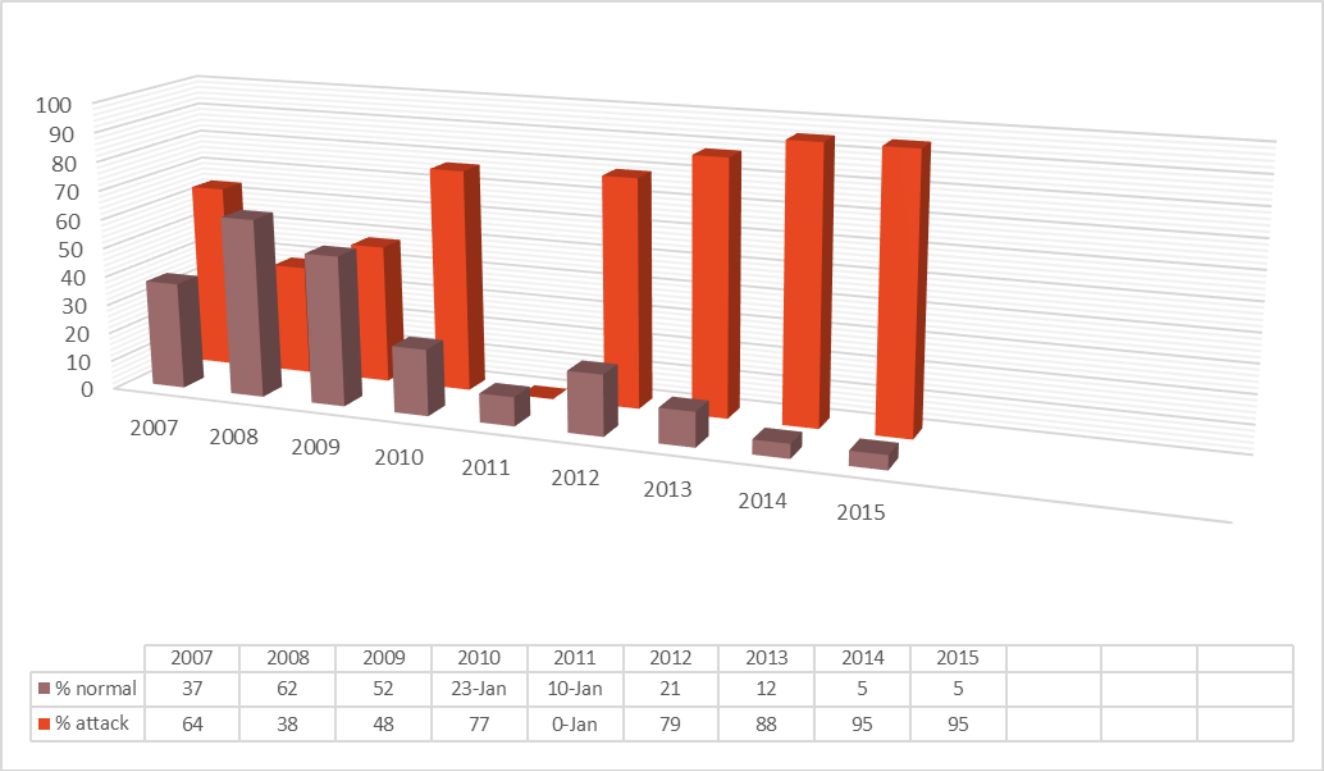
We note that our system has a few innovations that no other company implemented. It starts with the CDN implementation. Each one of them used their system within the original server. It means that their server should focus both on mitigating any computer attack, and also continue serving its clients properly and all that implies (interpret each request, deal with the databases, and perform complex calculations). employed a Random Forest (RF) classifier, it learned an almost linear classifier. All points to the left of x = 4.5 (approx.) will be classified as nominal by the classifier, including the unlabeled point marked in green. In contrast, the green point will be classified as an anomaly by the anomaly detector. Since the classifier learns a decision boundary between the two classes, it only checks which side of the boundary the instance is on before classifying it. On the other hand, most i.i.d point-based anomaly detectors (like in this example) are sensitive to the data density; instances that are in sparse regions are more likely to be flagged as anomalies by default. Whether to choose an anomaly detector or a classifier is application-dependent and there are likely use cases for both types of behaviors. Given the massive yearly distribution of the number of instances in the Kyoto dataset (fig. 3.1) and the computation capacities available, it was possible to conclude that this project should focus on a single month. To select it, the distribution of the class labels was studied.

Table 3.1: The confusion matrix for classifying attacks

---	Predicted normal	Predicted attack
Actual normal	True Negative (TN)	False Positive (FP)
Actual attack	False Negative (FN)	True Positive (TP)

The graphic displayed in figure 3.2 indicates the distribution of the percentage of normal and attack behavior in the dataset. From the observation of figure 3.3, it is possible to understand that the distribution of the class labels (normal/attack) is not constant during the nine years of the study. Contrary to 2008, which displays 62.25% of normal behavior, 2014 only displays 5.08%. Naturally, this element had to be taken into account in the election of the month to be studied. The initial plan was to use the most recent month, meaning December 2015. However, after examining the real-world distribution of the percentage normal/attack, it was elemental that the dataset was

very atypical. Usually, the number of attacks represents a small portion of the events registered. So,



	2007	2008	2009	2010	2011	2012	2013	2014	2015			
% normal	37	62	52	23-Jan	10-Jan	21	12	5	5			
% attack	64	38	48	77	0-Jan	79	88	95	95			

Figure 3.4: Distribution of the percentage of normal and attack behavior per year (from 2007 to 2015) [32].

Besides, each one of our related workers considered the amount requests coming from an end-point. Our mitigating algorithm considers both the amount of requests and the specific request. This means that when the request has a minor significance in the meaning of server calculations and network attack, it will get the same treatment in our mitigating algorithm. (see our calculation le algorithm). This novel research developed a unique detection methodology using multidimensional topological features in data mining algorithms. to keep this thesis project as close as possible to reality, it was decided that due to its relation normal/attack, a month of 2008 would be the best candidate.

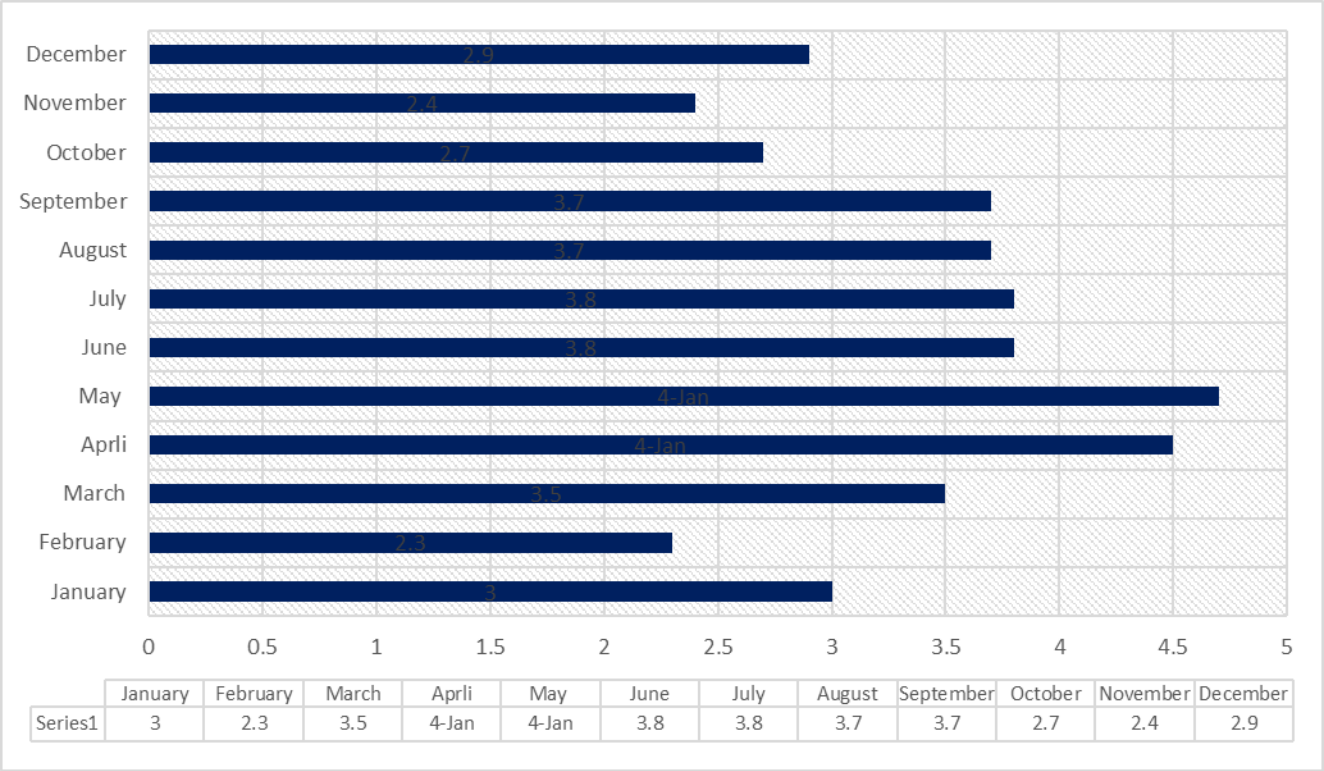
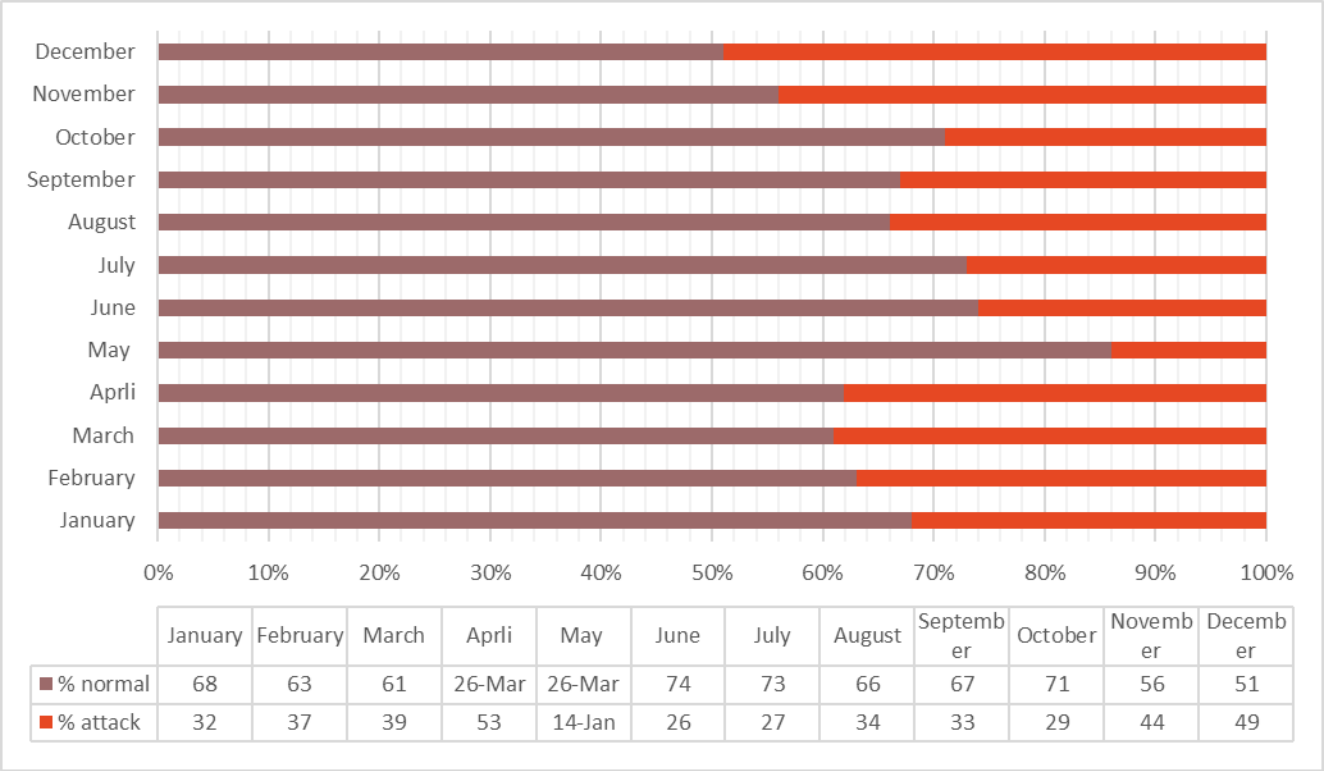


Figure 3.5 :2008 monthly distribution, in millions, of the number of instances [32].

Studying the graphics of figure 3.5 it is possible to infer that the number of instances displayed per month for the year 2008 lays within an adequate interval regarding the computation and time capacities available. Then, the monthly relation normal/attack was studied (figure 3.5).

The monthly distribution of the percentage of normal and attack behavior for the year 2008 is quite constant. The percentage of the normal class varies from a minimum of 47% in April to a maximum of 86% in May. So, since May was the month that displayed, at the same time, more instances and a higher percentage of normal behavior, it was elected to be this thesis domain of study. The data were then randomly divided into three sets, train (60 percent), validation (20 percent) and test (20 percent).



	January	February	March	Aprli	May	June	July	August	Septemb er	October	Novemb er	Decemb er
% normal	68	63	61	26-Mar	26-Mar	74	73	66	67	71	56	51
% attack	32	37	39	53	14-Jan	26	27	34	33	29	44	49

Figure 3.6: 2008 monthly distribution of the percentage normal/attack [32].

Topological multidimensionality. Static and dynamic analysis is combined with features. Apart from special file resources. Review thoroughly. There is no single study that leverages multidimensional research out of more than 500 studies over the last six years. Via static and dynamic analysis, the topological properties. Properties in files.

3.11 DATA PREPROCESSING

As real-world data may not always be available in the proper formats, data preprocessing often generates a very significant impact on a machine learning algorithm prediction performance. Also, building a customize data representation is a unique opportunity to incorporate specific domain knowledge into the data and, consequently, into the predictive process [21].

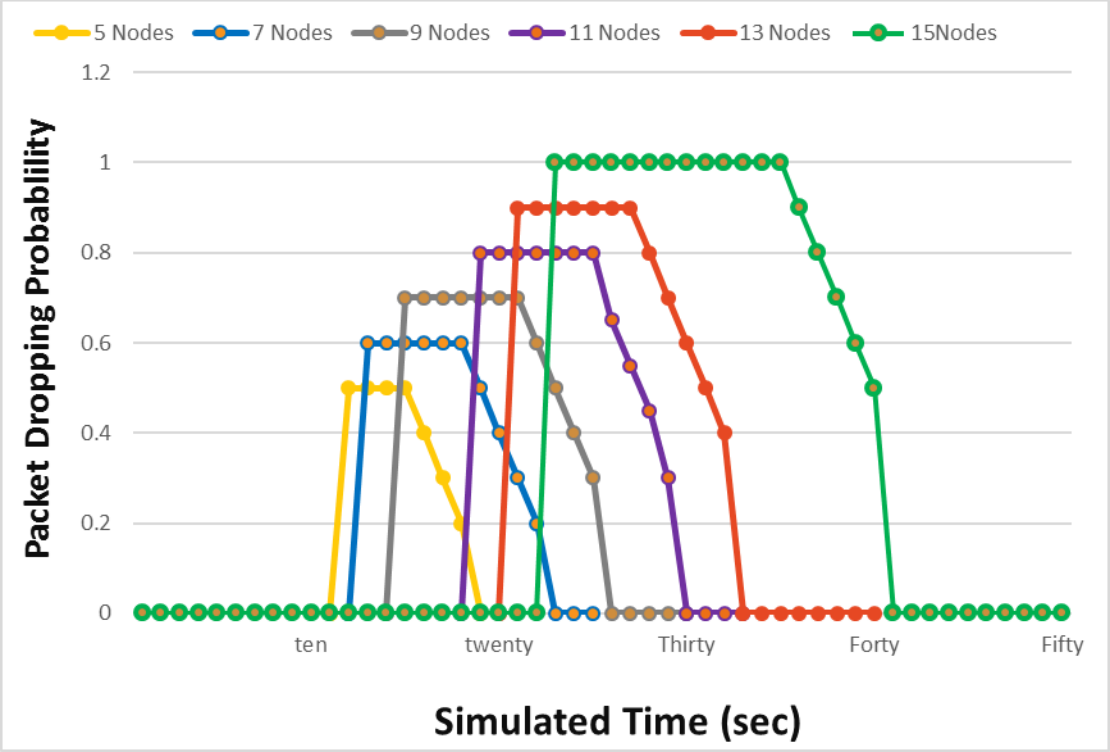


Figure 3.7: Graph describing the Packet Dropping Probability

Figure 3.8 graph describing the packets that are dropped from the router when the hacker attacked. As stated before, feature selection mechanisms are commonly applied as a preprocessing step before the implementation of a machine learning algorithm. However, the preprocessing procedures presented in this section simply cover the transformations performed on the original dataset before the application of the filtering tool and the intelligent intrusion detection algorithms. The preprocessing mechanisms were implemented with the assistance of the sci-kit learn library [31].

4. RESULTS

In this section, this thesis will provide the evaluation and designing of the computer security threats using machine learning techniques and achieve the results of accuracy of 92.14%. These simulation results are developed using MATLAB the result we achieved in detecting and classifying the attacks is précised. The intrusion detection results are explored next. Initially, the complete dataset is used to predict the attacks, and afterward, only the subset of features selected by the data reduction algorithm is used. In the end, the results are compared in a performance evaluation section.

4.1 DATA REDUCTION TOOL

Cybersecurity systems equipped with intrusion detection mechanisms typically utilize prior expert knowledge to manually establish which variables are most likely to be useful in the prevention of a cyber-attack. These variables may posteriorly be aggregated using linear correlation to expedite the detection process. In this thesis, to reproduce the functioning of a simple traditional cybersecurity environment and, at the same time, to understand what is happening in the first stage of the FCBF algorithm, a relevance analysis was performed. However, instead of only applying an empirical approach to manually determine the relevance of each variable, which consists of using information about previous well-succeeded attacks, both the linear and the non-linear correlation measures introduced earlier, ρ and SU were applied. To do so, the symmetrical uncertainty computation was performed by calculating the values of entropy and information gain separately. Anomaly detector (AAD) vs. Classifier (Random Forest) when labeled data is available. (a) Shows the labeled dataset. Only the point marked in green at the top left is unlabeled

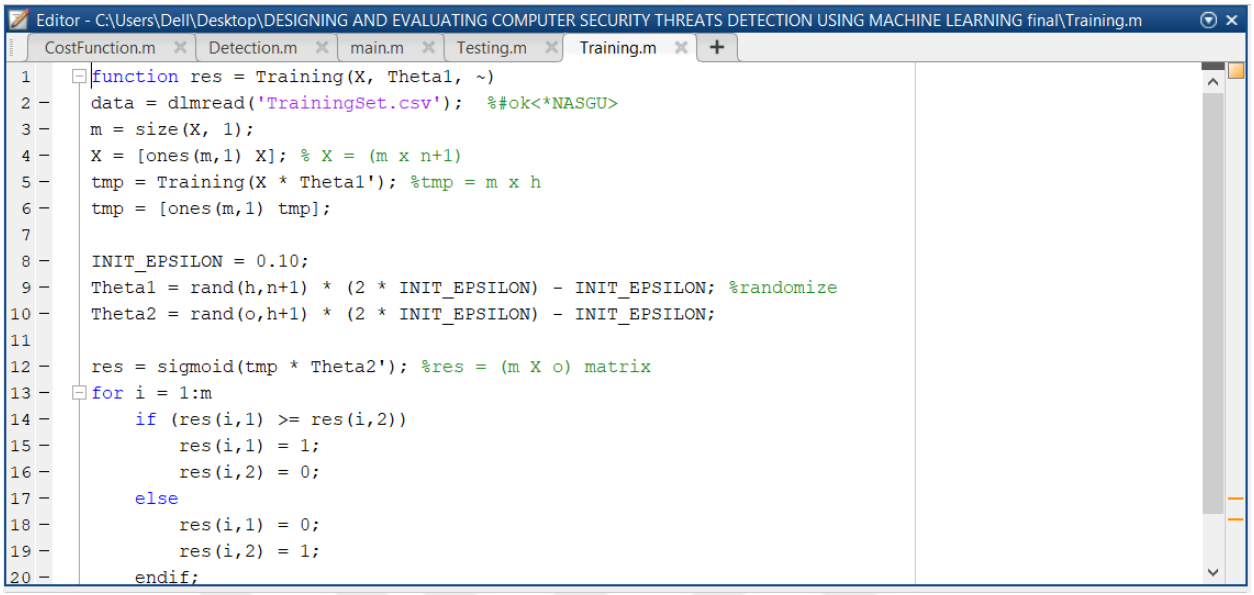


Figure 4.1: Screenshot of Labeled Dataset Trained on Matlab

Figure 4.2 exhibits the absolute values of the linear and non-linear correlation between each of the 47 features and the class label.

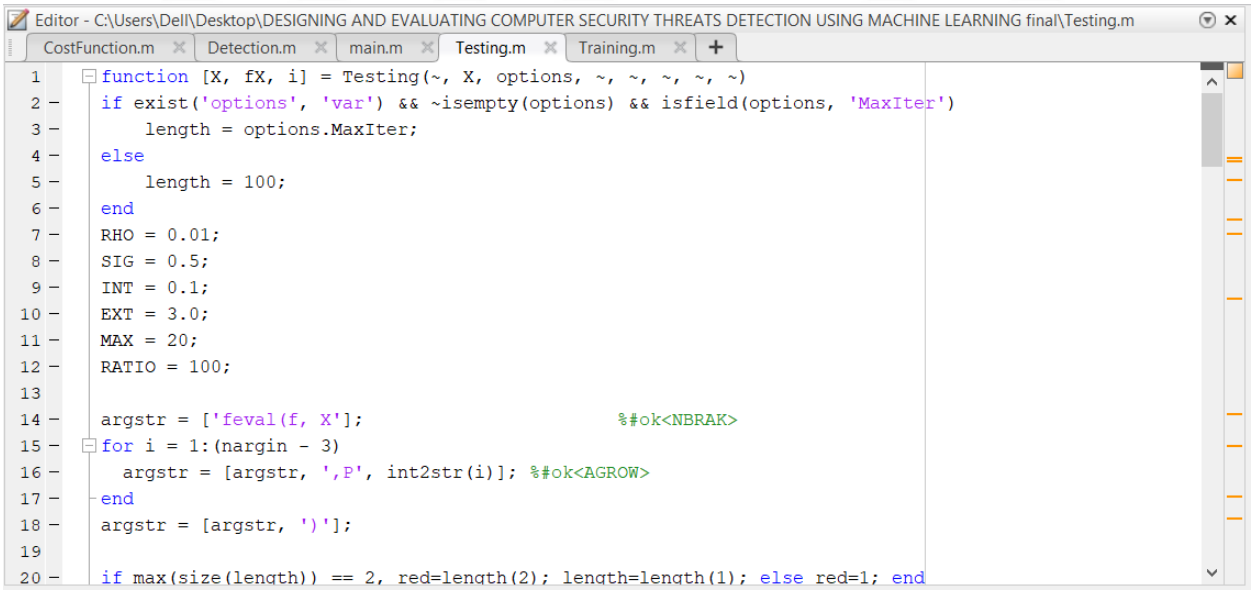


Figure 4.2: Screenshot of Dataset testing on Matlab

The graphics in figure 4.2 allow the identification of the most relevant (and irrelevant) variables to the Testing process. Since the graphics scales were disparate, it was decided not to represent the values in the same figure, but side by side.

f9, DST host count, presented the highest values of ρ and SU, which implies that the number of connections, among the last 100, which source and destination IP address are the same is a valuable indicator to the IDS. This is also accurate to other variables as f4, f5, f30, and f46. On the contrary, variables as f1, f2, and f3 do not appear to provide any essential information to the attack recognition process and may easily be discarded.

4.2 FAST CORRELATION BASED FILTER

The implementation of the FCBF algorithm was achieved by applying the pseudo-code presented to the preprocessed training set. As introduced before, the algorithm used in this thesis as a data reduction tool, FCBF, consists of two stages. In the first one, the non-linear relevance evaluation process presented above is used to establish which features provide more information about the class labels. In the second, the same information theory concept allows the elimination of redundant features by quantifying the strength of the relationship between each feature and the remaining ones. The FCBF algorithm includes a parameter δ which, in this thesis case, represents the percentage of features considered irrelevant, according to the respective value of SUc, their relation with the class labels. As the only input parameter to the algorithm, δ defines the number of features present in the final subset and, therefore, its significance must be carefully evaluated. Table 4.1 presents the number of relevant features, or, in other words, the number of features existing in the subset before the elimination of the redundant ones, and the number and respective numeric identification of the features displayed in the final subset for 9 different values of δ.

Table 4.1: FCBF results in function of δ.

δ	# relevant features	# final features	features in the final subset
0	47	15	{9, 5, 46, 30, 27, 47, 4, 33, 24, 20, 21, 28, 31, 1, 29}
10	43	14	{9, 5, 46, 30, 27, 47, 4, 33, 24, 20, 21, 28, 31, 1}
20	38	13	{9, 5, 46, 30, 27, 47, 4, 33, 24, 20, 21, 28, 31}
30	34	12	{9, 5, 46, 30, 27, 47, 4, 33, 24, 20, 21, 28}
40	29	9	{9, 5, 46, 30, 27, 47, 4, 33, 24}
50	24	8	{9, 5, 46, 30, 27, 47, 4, 33}
60	20	8	{9, 5, 46, 30, 27, 47, 4, 33}
70	15	6	{9, 5, 46, 30, 27, 47}
75	13	6	{9, 5, 46, 30, 27, 47}

Table 4.1 presents the subsets of features delivered by the FCBF algorithm for different values of δ. For δ = 0, the final subset consists of 15 features. For δ = 75, however, the number of selected features is reduced to 6, which represents less than 15% of the 47 variables obtained by encoding the original dataset (section 3.2). This signifies that keeping the 25% of the most relevant variables, the fast correlation-based filter allows an 87% reduction in the number of variables and, consecutively, in the size of the dataset. As the pairs, f30 and f27 and f46 and f47 were initially part of the same features, service, and protocol type, respectively, the data reduction achieved for δ = 75 considering the initial dataset was 83% (from 23 to 4 features). To enhance data understanding, table 4.2 presents the 15 features selected for δ = 0.

Table 4.2: 15 features selected by FCBF for δ = 0.

Feature ID	Feature name
9	DST host SRV count
5	same SRV rate
46	protocol TCP
30	service ssh
27	service SMTP
47	protocol UDP
4	Count
33	flag REJ
24	service pop3
20	service FTP
28	service SNL
31	service SSL
1	Duration
29	service SNMP

Figure 4.3 Shows the anomaly score contours when the anomaly detector (AAD) was trained with the labeled instances (i.e., ensemble weights were tuned to take the labels into account).

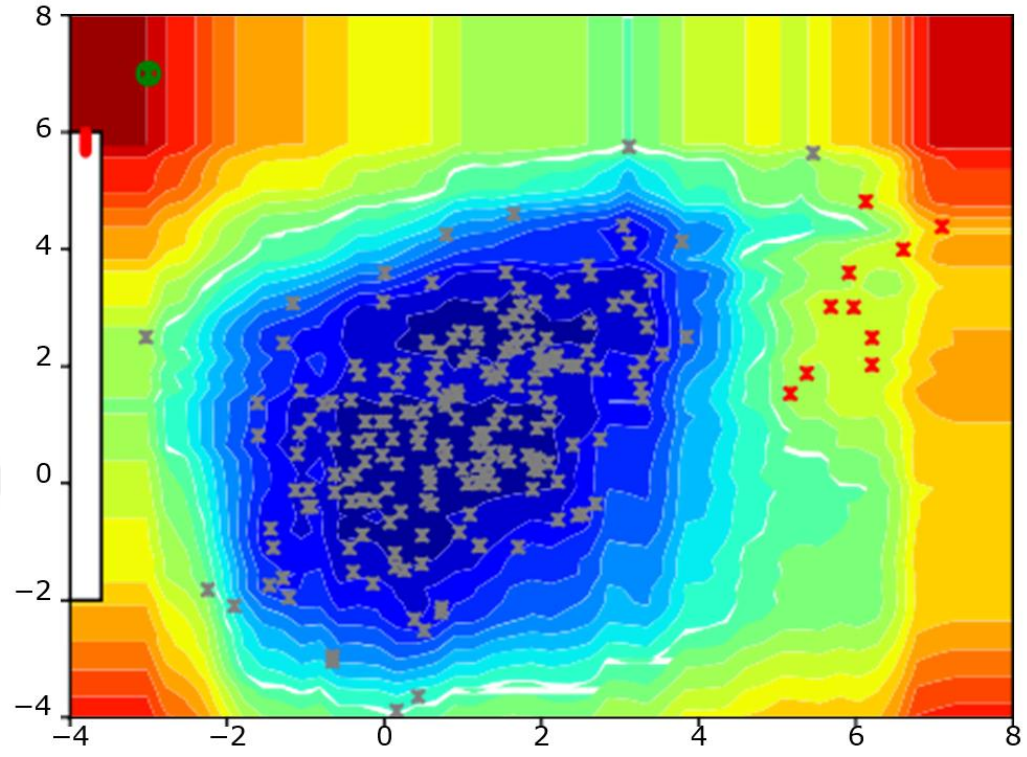


Figure 4.3: anomaly score contours when the anomaly detector (AAD) was trained

The computer security threats are detected and been solved in MATLAB Figure 4.4 Shows the probability contours for the anomaly class when a classifier was employed. In both (b) and (c), red corresponds to more anomalous, and blue corresponds to more nominal. Although we employed a Random Forest (RF) classifier, it learned an almost linear classifier. All points to the left of x = 4.5 (approx.) will be classified as nominal by the classifier, including the unlabeled point marked in green. In contrast, the green point will be classified as an anomaly by the anomaly detector. Since the classifier learns a decision boundary between the two classes, it only checks which side of the boundary the instance is on before classifying it. On the other hand, most i.i.d point-based anomaly detectors (like in this example) are sensitive to the data density; instances that are in sparse regions are more likely to be flagged as anomalies by default. Whether to choose an anomaly detector or a classifier is application-dependent and there are likely use cases for both types of behaviors. The bottom row illustrates the anomaly descriptions generated by

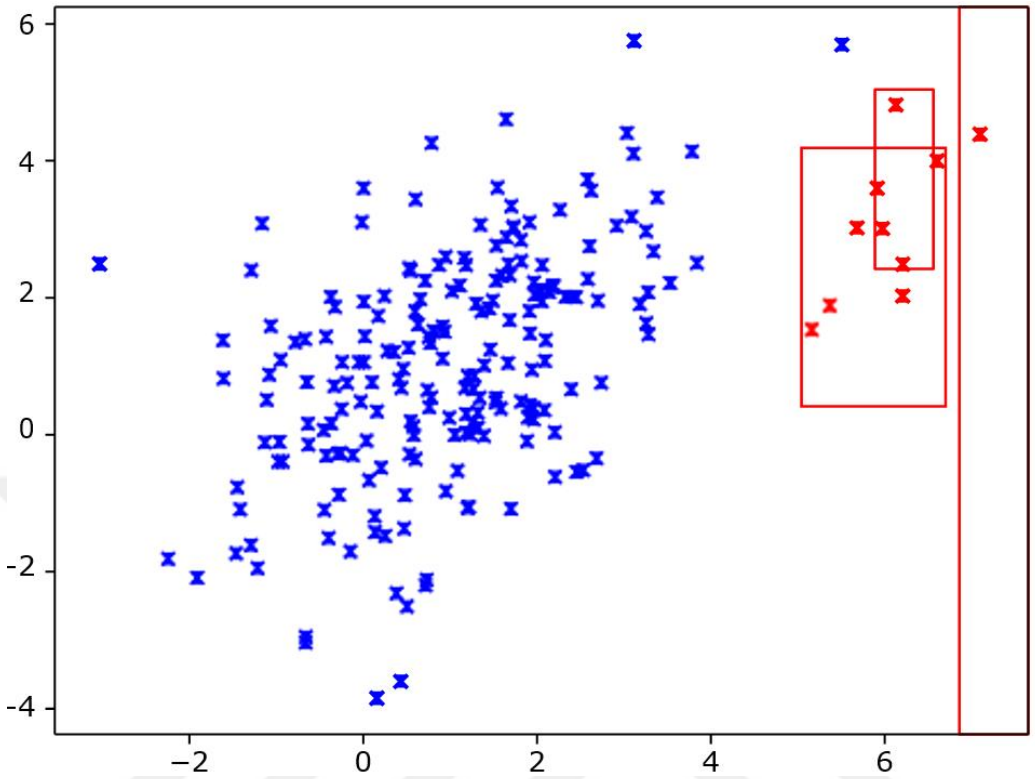


Figure 4.4: Descriptions (AAD)

The AAD definition is seen in the figure above, which is trained and checked in MATLAB for each node. It has 100 facilities, e.g. a recording or website. Each of these utilities. It has 100 objects, such as plugins for flash, pictures, text, etc. Therefore, there are 100 = 10000 (S SObject) objects in each P2P Node that the client can reach Set the maximum number of P2P nodes to 9. At some intervals, P2P nodes update each other periodically. Send P2P update notes (PRequest) to store, for example, data on the right node. This relies upon the protocol used by P2P. P2P nodes also upgrade themselves and include the newest iteration of the stored data at certain intervals (CDN DBR request).

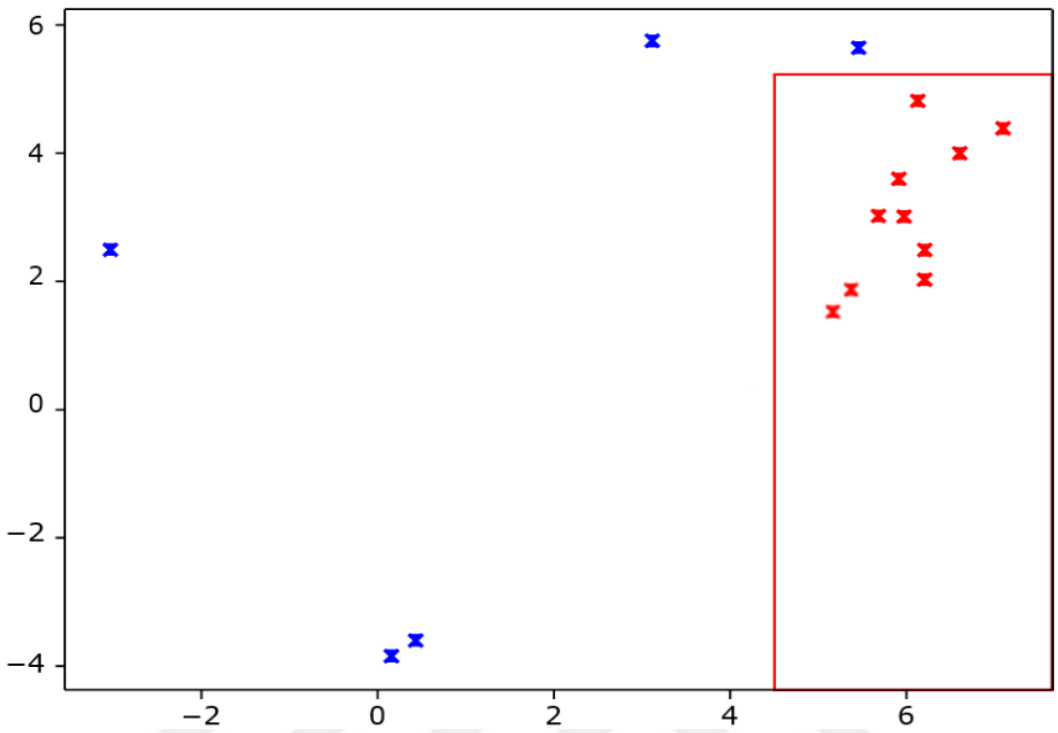


Figure 4.5: Descriptions (Decision Tree Classifier)

Node (P), which gets a response to the client (c), should be ready to reply in less than a second to the request. This reference point for the simulation is used to show whether the site can be accessed without adverse effects and whether the user is comfortable surfing. Conveying and processing. The application must take place during that period. The optimal response times for the first CDN site item are [26], [27], [28] below 200ms, as follows. Each node has 100 facilities, such as a website or video, for example. There are 100 items in each of these services, e.g. flash plugins, images, text, and so on. Therefore, there are 100 = 10000 (S S Object) objects in each P2P Node that the client can control. Set the maximum number of P2P nodes to 9. By sending P2P update notes (P Request) to store, for example, data on the correct node, P2P nodes update each other daily at certain intervals. This relies upon the protocol used by P2P. P2P nodes also upgrade themselves and provide the newest version of the stored data at certain times (CDN DB Request).

Besides, each server (or, to be accurate, the P2P nodes and the data center) is assumed to have sufficient storage and availability to store the different contents and sufficient output to handle any incoming HTTP request without substantial delays. Both relations are also assumed. Between

nodes, servers, clients, for example, are dimensioned in a manner that no bandwidth constraint can exist. Accordingly, there is no expectation of a bottleneck effect at any moment.

4.3 SIMULATION PARAMETER FOR ATTACK CLASSIFICATION

The simulation of the machine is based on the previous simulation (see chapter 4.2). The simulation parameter is still the same (see Table 4.1) and can be extended only by the parameters of the system used to simulate the attack (see Table 4.2). The high 20-30 GB / s attack rate was chosen to cover.

Table 4.3: Simulation parameter - attack simulation

Parameter	Werte
Computer attack rate [GB/s]	20-30
Simulation time [h] (T_{Simu2})	2

There are two justifications. On the one side, a loss of bandwidth that leads to a bottleneck effect is obtained. The partnership is occupied with the server that is targeted. This results in a router buffer overflow that is positioned in front of the attack server. Thus, this router begins. By the elimination of packets from the queue, as a result, it is not possible to differentiate the legal HTTP request from the unconstitutional one, because it does not even join the instance that will filter it. It could not be replied to by the server because of its high workload, even though the request reached the server. Simulation time is limited to two hours, as in a normal simulation. . On the one hand, this time interval is adequate to give the system enough time to evolve and, on the other hand, sufficient time to observe potential extreme states.

in application layer 1 of the MATLAB subnet, the execution of a software attack is simulated. Ping, as well as HTTP requests, are executed by any of the 1000 attackers. However, these requests are executed within 0.008 seconds. The ping request packet size is the legally acceptable size of 65535 bytes.

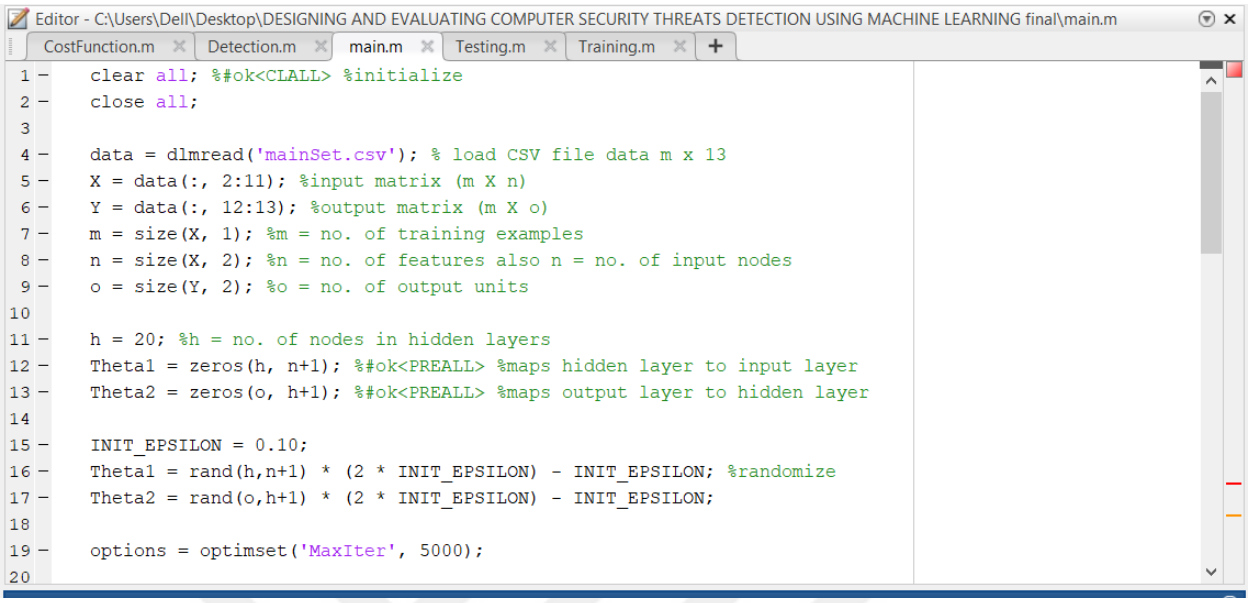


Figure 4.6: Screenshot of Main function code

Figure 4.6 demonstrates the key function code used to identify attacks when a computer attack is simulated in the MATLAB subnet 1 layer of the program. Each of the 1000 attackers executes ping as well as HTTP requests. These requests are, however, carried out in 0.008 seconds. The package size of the ping request is the maximum permitted size of 65535 bytes.

4.4 SIMULATION PARAMETER – COMPUTER SECURITY THREATS

The extended Computer simulation has the following parameters (table 4.2)

Table 4.4: Simulation parameter – Computer security threats

Parameter	Werte
computer attack rate [GB/s]	70-80
Simulation time [h] (T_{Simu2})	4

In this simulation, not only is the location of the Moskva attack targeted, but also the Trojan attack and the spyware attack. The strength of the attack is tripled and distributed throughout the three places. This results in three out of nine P2P nodes being inaccessible.

4.5 RESULTS AND DISCUSSION

The tables below illustrate that. The simulated effects. Attacks on PCs. Every statistic also provides the simulation outcomes of the normal simulation as a reference value. The results of the smaller Moskva computer assault are displayed in blue, the results of the expanded computer simulation are displayed in red, and the results of the normal simulation are displayed in green. The bandwidth of, e.g., the Moskva location. This is attributable to a computer assault of 100 percent. There is blue for the use of the two virtual simulations and red for the normal simulation. Accordingly, The P2P Bandwidth

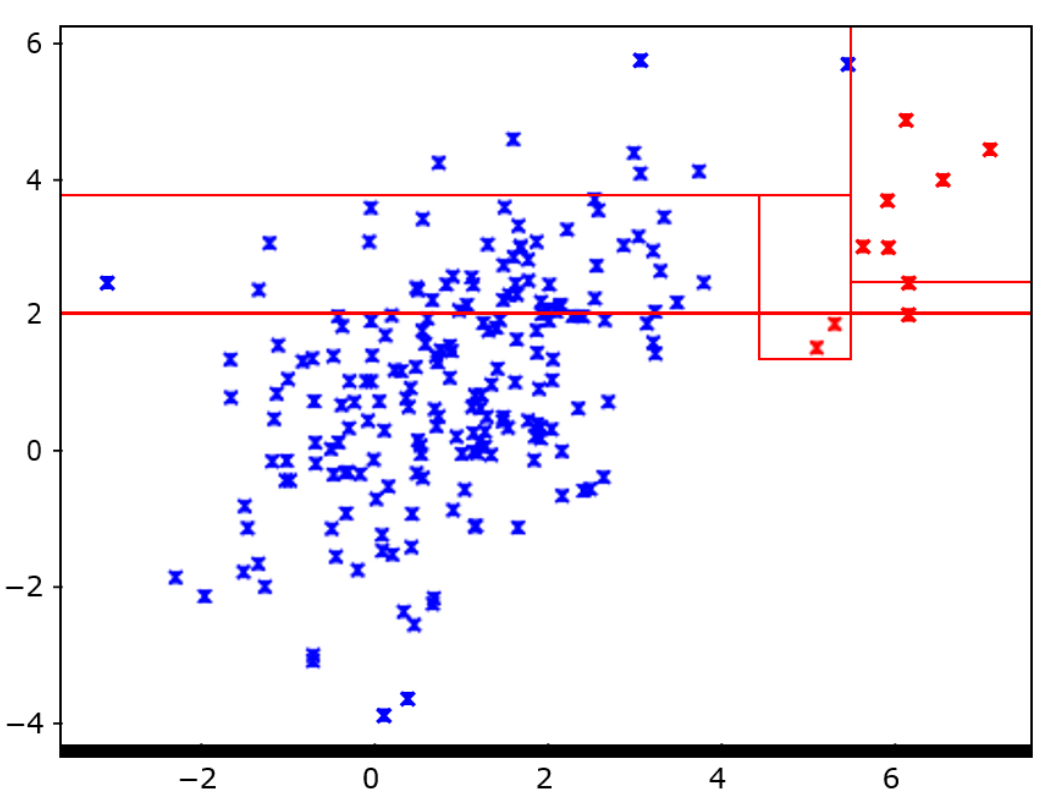


Figure 4.7: Description (Random forest classifier)

RF. Since both DT and RF are tree-based, we employed the same strategy as the one for generating compact descriptions with AAD. For DT, which comprises only a single tree that partitions the entire data exclusively, this naturally corresponds to simple rule-extraction. Since the anomalies are well-separated and fully labeled, DT works well. In the case of AAD and RF, there are many overlapping subspaces, and the description algorithm tries to minimize the total volume which covers all labeled anomalies. Therefore, the selected subspaces are smaller. In all cases, the

subspaces could probably be pruned further based on the location of the anomalies, these response times are marginal worse than in the regular simulation, roughly 0.1 seconds on average (4.4). The results of the extended computer simulation are somewhat different from the normal simulation. If three of the nine P2P nodes crash, the loading time of the website will increase (see figure 4.6). Figure 4.7 below shows the increase the loading time of the websites, which is characteristic of each venue. The location of the spyware is shown in this example. Throughout the simulation, an increase in the workload is noticeable at every node, since the P2P node in the machine is no longer available and all HTTP requests must be distributed across the other P2P nodes.

Table 4.5: Average object response time of computer security threats

Nodes	Average Attack Request (sec)
Attack_0_spyware	0.277
Attack _2_trojan	0.251
Attack _8_Computer	0.232
Attack _16_botnet	0.232
Attack _32_worm	0.226

Table 4.6: Average attack response time of different types of attacks

Nodes	Average Classification Response (sec)
Classification_0_ spyware	0.468
Classification _1_ Trojan	0.421
Classification _2_ Computer	0.399
Classification _3_ botnet	0.398
Classification _4_ worm	0.378

However, there is no major difference between the loading time of the website. A modest rise in the ca. On average, 0.25 seconds can be seen during a machine assault (see figure 4.5). The normal loading times can be seen in Figure 4.5. Both average times are within the permissible time allowed. Generally increased times, when a machine assault happens, can be clarified by the fact that the other accessible needs to respond to all client requests. Also, a request may be sent to a P2P node which is further away due to load balancing steps. In this way, the higher transmission delay for the routing of the request/response must be taken into account. Computer

nodes, for example, have a higher delay in processing their sessions due to increased requests (the two computer simulations whose time response is higher and the others are lower).

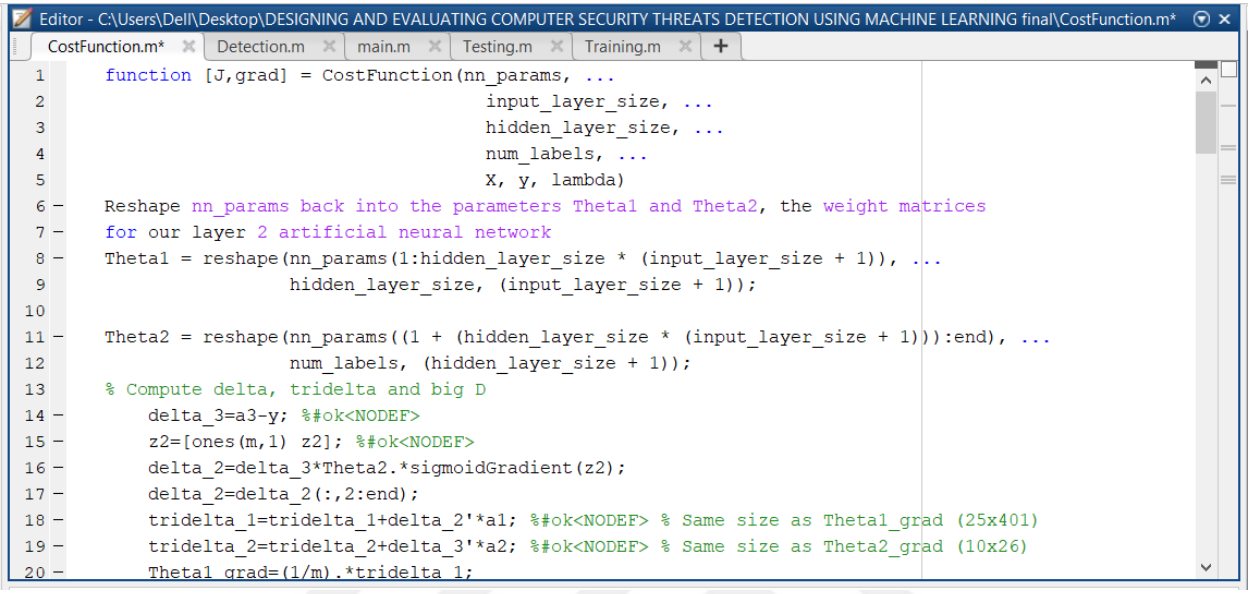


Figure 4.8: Snapshot of the cost function in Matlab

Figure 4.8 illustrates the cost function code to depict the results and the Simulation has shown that the architecture can withstand an assault from nine computer attacks in the regular simulation one that cannot be reached in the first computer attack simulation. This does not contribute to a major interference with the operation. Other machine attacks have always been able to respond to the requests of the 10,000 attacks.

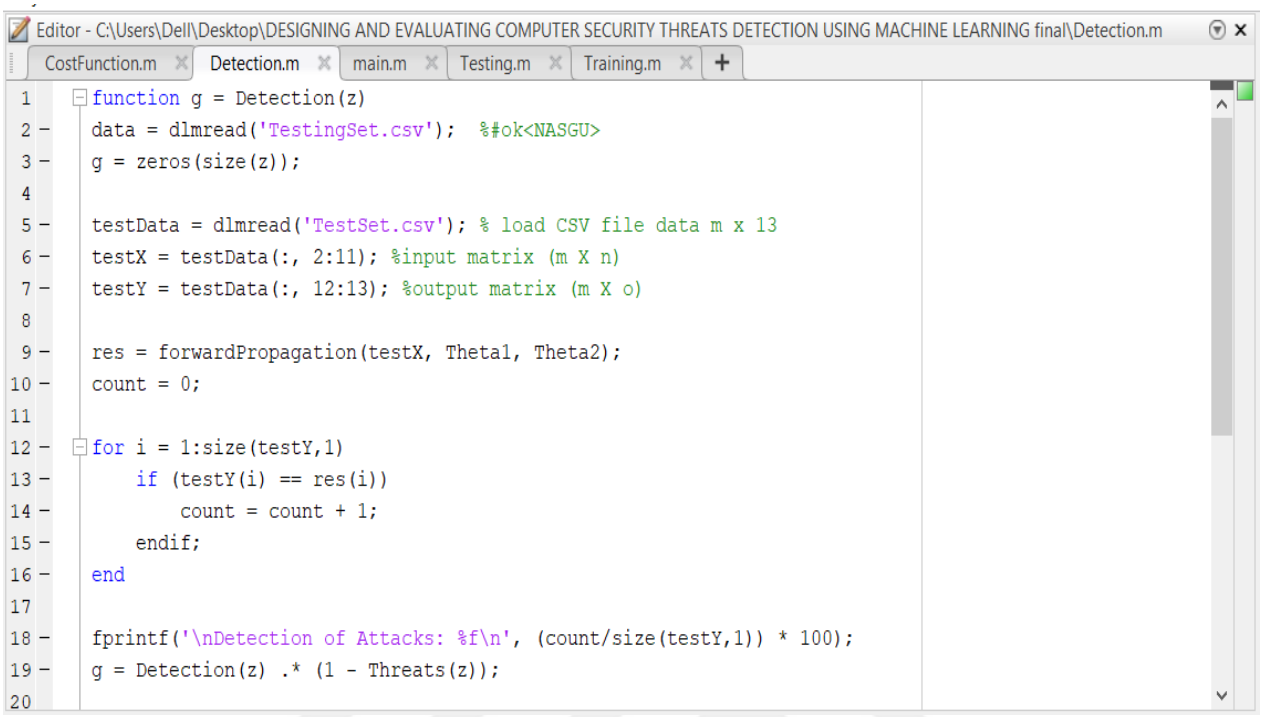


Figure 4.9: Snapshot of Detection of Computer security threats in Matlab

Figure 4.9 shows the detection code to detect the attacks in a running environment The extended the attack, in which three of the nine machine attacks were unachievable, made the loading times noticeably worse. However, a much higher effort must be made to start such an attack. To double the loading time of the website, it was necessary to triple the attack rate in the simulation.

4.6 NATIONAL CYBER DEFENSE APPROACH

As a developing country has a great increase in creating and using information technologies and internet access. Both the government parties and public and private sectors have offered so many information systems and e-services. Especially in the last decade information systems for national critical infrastructures and e-government services such as e-health, and e-education are developed and launched for the citizens. Also, with the increasing scope and quality of the information and communications technologies frameworks people now can access these services easily and from everywhere throughout the whole country. In the meantime, these are some of the main factors that accelerate the development of Turkey in economics, and politics as an emerging country.

Turkey has pointed out the protection of its critical infrastructures and information systems against cyber threats as an urgent and priority task in recent years. Although it is not sufficient yet, the government takes some actions on creating a national cybersecurity strategy. In this context, the government has been cooperating with private sector initiatives and security experts to increase its capabilities in cyber defense and to create a cyber-defense system. Turkey is now at the beginning of this journey and therefore has been trying to adopt a passive cyber defense strategy. Although all these actions are admirable, still a responsible national cyber defense institution to which the official cybersecurity defense strategy is assigned has not been established yet.

The National Cyber Security Strategy and the Action Plan for 2013-2014, is the primary national cybersecurity policy document of Turkey, was established in December 2012 by the National Cyber Security Board. The Board was created in October 2012 to set up the policy framework, to coordinate the cyber defense studies, and to follow their results.

4.6.1 International Cooperation

Cooperation with other nations, and international organizations such as the UN, NATO, EU, and OECD, is emphasized as one of the critical success factors of cyber defense strategy. Turkey should get involved in international events to improve its capabilities and share expertise on cybersecurity issues.

4.6.2 Cyber Security Legislations

According to the action plan, the establishment of a national cybersecurity legislation framework plays a very important role in fighting against cyber threats. Therefore, the government should create the necessary laws to struggle effectively with cyber-attacks and attackers.

4.6.3 Assigning Roles and Responsibilities

The roles and responsibilities of government bodies, public and private sectors, universities, and social organizations should be defined clearly for conducting an efficient cyber defense strategy and action plan. New institutions and authorities, responsible for conducting, coordinating, and assisting the cyber defense strategy, should be established urgently.

4.6.4 National Cooperation

Government agencies, public and private sector should cooperate to take actions for protecting the national critical infrastructures, and vulnerable sectors.

4.6.5 Education and Awareness

Training and raising awareness of human resources about cybersecurity is an efficient way to create safety and therefore it is emphasized as essential.

4.7 INTERNATIONAL ORGANIZATIONS

Because of the global nature of cyberspace and becoming of the cybersecurity as one of the top priority/urgent issues in the last decade, Governmental organizations such as the Council of Europe (CoE), the European Union (EU), the North Atlantic Treaty Organization (NATO), the Organization for Economic Cooperation and Development (OECD), the Organization for Security And Cooperation in Europe (OSCE) and he United Nations (UN) have concentrated heavily on dealing with it. But different organizations address different aspects of cyber subjects. As seen in the table below, OECD, EU, and (CoE) have performed activities about internet governance and cyber-crime issues whereas NATO and the UN focus on cyber warfare. also, the UN carries studies on all fields of cybersecurity while OSCE deals with cyber-crime and cyber-terrorism.

Table 4.7: Dimensions of Cyber Security

	Computer	Spyware	Trojans	Worms	Botnet	Malware
Internet Governance	■	■	■			■
Cyber Crime	■	■	■	■		■
Cyber Terrorism				■		■
Cyber Warfare					■	■

When we look more comprehensively, it is seen from the table below, that in Malware attacks the EU makes efforts on economic issues and data privacy areas of cybersecurity. On the other hand,

since their duties are more related to international security in military bases, the UN, NATO, and OSCE are working on cyber armed attacks response, terrorists use of the internet, national security-relevant cybercrime. Besides, since the only UN and NATO are dealing with the cyber armed attack response activities their roles in cyber warfare will be detailed in this section.

5. DISCUSSION

The first objective of this thesis was to simulate the operation of a typical real-life intelligent intrusion detection system. To do so, three machine learning algorithms were studied and posteriorly selected to be implemented: artificial neural networks, naive Bayes, and random forest. Before their implementation, as part of an extremely common preprocessing stage, the number of features was delimited using a variable ranking strategy. Or, in other words, the problem initial variables were rated according to the quantity of information provided to the detection process, and afterward the less relevant were excluded. The results obtained for this traditional approach were very satisfactory, more than 91% of the attacks were detected using the ANN and RF. With NB, however, the detection ability only covered ≈82% of the intrusions. Also, the area under the ROC curve, which indicates the model's class separation capacity, displayed values above 92.14% for all of the three classifiers.

Notwithstanding, this thesis's primary goal was to build a general data reduction tool capable of maintaining the prediction performance of an IDS and improving data understanding. Besides, the tool idealized was intended to be universal and attachable to any given intrusion detection system already in functioning, supporting as many technological solutions as possible. The approach presented to implement this tool consisted in adding a data reduction element to the intrusion detection framework, based on a feature selection strategy. To do so, the fast correlation-based filter algorithm was implemented. The only input parameter to this algorithm, apart from the cybersecurity data in the study, is denominated δ and allows the elimination of a given percentage of the most irrelevant variables. Varying δ from 0 to 75, the actual variation in the final number of features was from 15 to only 6, representing less than 20% of the original dataset. introduces the 15 selected features for $\delta = 0$.

To evaluate the performance of the data reduction tool, the same three ML models were used. Using the 15 features selected for $\delta = 0$, the values obtained for the class separation indicator, AUC, was in the 98% range for the ANN and RF and 93.5% for the NB. The model's detection capabilities, expressed by the recall, however, decrease almost 15% in comparison with the results displayed. This phenomenon may easily be explained by the usage of irrelevant features as if they were relevant since, for $\delta = 0$, no feature is discarded by the FCBF algorithm due to its irrelevance. For $\delta = 75$, even with a reduction of 9 variables, the results are quite similar to the ones presented

for $\delta = 0$. Comparing these performance metrics with the ones obtained for the traditional approach studied earlier, it is noticeable a slight decline. The AUC values are now in the 92% range for the ANN and RF and 90.47% for the NB classifier. Also, the detection ability of each predictor is ≈74% for the ANN, ≈68% for the NB, and ≈77% for the RF. Notwithstanding, taking into account the data reduction achieved (≈80%) the results obtained with the novel approach proposed are very satisfactory. The class separation indicator, AUC, remains above the 90% for the ANN and RF and above 80% for the NB, the classifier which presents the poorest performance. Better results would probably be achieved with the implementation of a customized feature selection algorithm for each classifier, possibly using a wrapper or embedded technique. However, as the data reduction tool idealized was intended to be universal and attachable to as many diversified technologies as possible, the results obtained allow us to conclude that the FCBF appears to present itself as an extremely viable solution.

5.1 ROLE OF MACHINE LEARNING IN DETECTION OF ATTACKS

To evaluate the performance of the data reduction tool, the same three ML models were used. Using the 15 features selected for $\delta = 0$, the values obtained for the class separation indicator, AUC, was in the 98% range for the ANN and RF and 93.5% for the NB. The model's detection capabilities, expressed by the recall, however, decrease almost 15% in comparison with the results displayed in table 5.1.

Table 5.1: Comparison with the Results

	Accuracy	Recall	Precision	Specificity	FAR	NPV	AUC
Artificial Neural Network [46]	92.14	88.83	91.73	95.46	1.54	96.99	98.09
Naive Bayes [47]	90.47	81.89	80.20	91.69	8.31	96.45	93.05
Random Forest [48]	91.93	87.28	91.27	95.04	0.96	96.81	98.45

This phenomenon may easily be explained by the usage of irrelevant features as if they were relevant since, for $\delta = 0$, no feature is discarded by the FCBF algorithm used in this thesis but due to its irrelevance. For $\delta = 75$, even with a reduction of 9 variables, the results are quite similar to the ones presented for $\delta = 0$. Comparing these performance metrics with the ones obtained for the

traditional approach studied earlier, it is noticeable a slight decline. The AUC values are now in the 92% range for the ANN and RF and 90.01% for the NB classifier. Also, the detection ability of each predictor is ≈74% for the ANN, ≈68% for the NB, and ≈77% for the RF. Notwithstanding, taking into account the data reduction achieved (≈80%) the results obtained with the novel approach proposed are very satisfactory. The class separation indicator, AUC, remains above 90% for the ANN and RF and above 80% for the NB, the classifier which presents the poorest performance. Better results would probably be achieved with the implementation of a customized feature selection algorithm for each classifier, possibly using a wrapper or embedded technique.

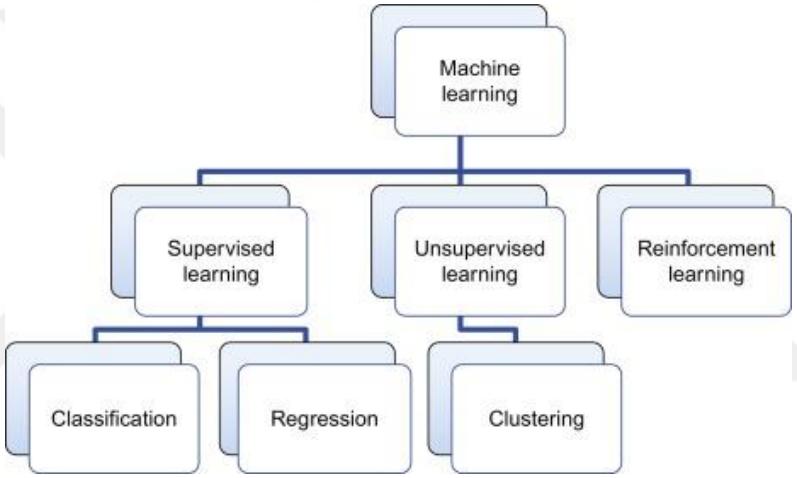


Figure 5.1: Machine Learning Functionality

However, as the data reduction tool idealized was intended to be universal and attachable to as much as diversified technologies as possible, the results obtained allow us to conclude that the FCBF appears to present itself as an extremely viable solution while smart devices have made our lives much simpler and more relaxed, privacy and security threats are still a threat. Since the internet is the room in which data is shared and because of the internet's free access functionality, this data will in many cases be used for illegal activity [36]. With the increased use of device, spyware, and Trojans programs, the number of common malicious attacks such as phishing, denial-of-service (Dos) or distributed denial-of-service (Computer), Probe Remote to Local (R2L) has increased. To make the device and data safer to be available and used only by designated agencies, logical controls are being used as software guard's security measures, keys, or access management can be conceptual checks. The intrusion prevention method is utilized among such conceptual checks to tackle attacks on machines, malware, and malware, as stated before... The Intrusion

Prevention Platform is built to recognize the occurrence of an intrusion in the event of a treasonous act and to be able to react by issuing warnings to designated entities. [37]. Two fault forensic methods are used by IDS and are Handwriting identification that works best to identify known attackers and Outlier detection that focuses on machine learning techniques because it works with forms of attack. Anomaly-based identification is the approach we are operating on, where the device trains itself so that it can differentiate here between the program's stable and dysfunctional or irregular actions.

5.2 MACHINE LEARNING TECHNIQUES

Because we are dealing with Neural Networks for intruder identification and tracking, this allows the framework to classify threats such as vulnerability scanning.

Attacks to information and information stability in organizations and artificial intelligence approach such that set of data are wanted to develop the machine [38]. For both Application Level, which is sponsor, and System Later, which is the provider for a series of selected algorithms, we use the choice to create a defense framework. Since it is difficult for the application-level data collection to obtain a data set of labeled data for malware detection. The heating element, moisture sensors, capacitive detector, and current between the batteries are included in each current sensor. Most of the data we gathered is comparable due to the state of the laboratories becoming identical.

5.3 ANALYSIS OF STUDY

The clustering algorithm is a classification algorithm that classifies data that augment the utility for researchers. Using clustering we can split data according to its classification. Clustering cannot work well on data that is continuous too, because if the regularization parameters are set high, it shall still form a very thin hyperplane to set the data even when it is continuous. Since we have lots of data and it is arranged constantly after implementing Clustering on the data set, it shows a run-time accuracy of 92.14percent in 4.55s.

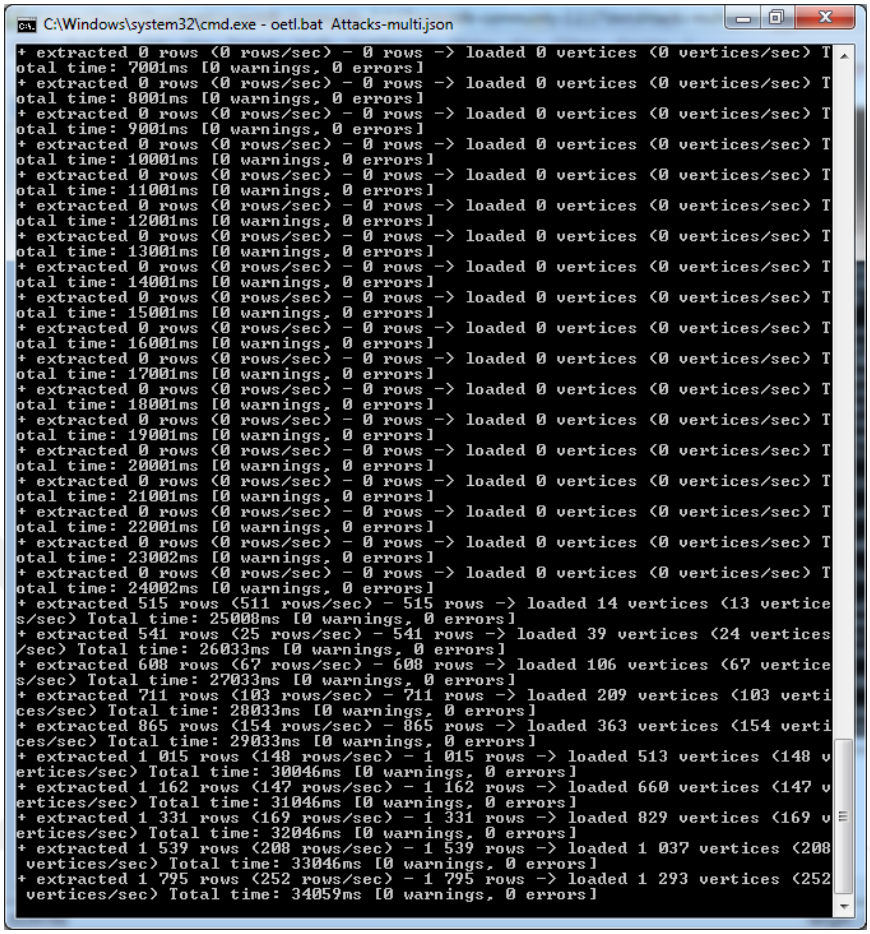


Figure 5.2 Example of Computer Security threat

This Figure shows the Attack on the computer which is resolved by Clustering. Clustering is tested on the network layer data where is classifies by creating a hyperplane to separate attacking and non-attacking classes. Since the computational power required on the actual data set was high, we ran it on a pre-processed data set, and where it concluded with an accuracy of 66.2 percent but took 5.17s to finish.

Table 5.2: The analysis of accuracy plus execution time of different attack classification in network

Attacks	Accuracy	Time
Adware	91 percent	2.41 s
Spyware	92.5 percent	2.69 s
Trojans	92.97 percent	2.83 s
Botnet [41]	71.7 percent	4.45 s
Malware [41]	66.2 percent	5.17 s

The classification has already been used and a single outcome is controlled by using microanalysis as a latent regression features forecast [39]. Our platform layer data set has 4 characteristics and has multiple layers. The classifier is built using a cognitive approach, which establishes a level of chance that essentially distinguishes the two classes...

5.4 RELEVANT MACHINE LEARNING ALGORITHMS

Another existing algorithms and easy to implement algorithms were tested on both application and network layer is kNN. K-Nearest Neighbour is a supervised learning algorithm that is useful in predicting and classifying and depends on labeled data to work [40]. It uses the labeled data to learn a function to give an appropriate output. It works well for all parameters and has low computational time. It assumes similar things near belong to a similar class. Determining the value of k is a difficult job and plays a vital role in the accuracy of the output. Though a higher value means less variance and exclusion of outlier value the downside is it ignore smaller patterns where useful data might reside. For our case, the optimal value of k was found to be 2 for application layer data and network layer data we used GridSearchCV to find out the best suitable value of k and it was 16. For the application layer, the accuracy came to be 91.7 percent in 2.45s and for the network layer, it was 92.14 percent.

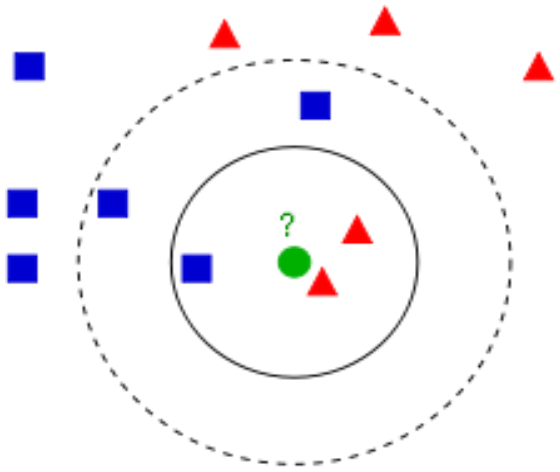


Figure 5.3: Example of classification using the k-NN for intrusion detection and classification. Source [22]

A decision tree algorithm is another supervised machine learning algorithm where it divides data into smaller parts until all parts fall under a similar class. For the application layer, the best attribute was humidity and it was selected as the root of the tree. The tree is made with the root having the highest priority and priority decreases as we go down. This priority is based on the information gain which is dependent on entropy. The result of the decision tree for the application layer is 92.13 percent with a run-time of 2.83s. As for the network layer, the first column is where the splitting began with the Computer attack and this was the root of the tree and progressed as explained before by splitting until it reached the class Normal or Abnormal. The accuracy of the decision tree in the network layer came out to be 89.6 percent in 1.86s.

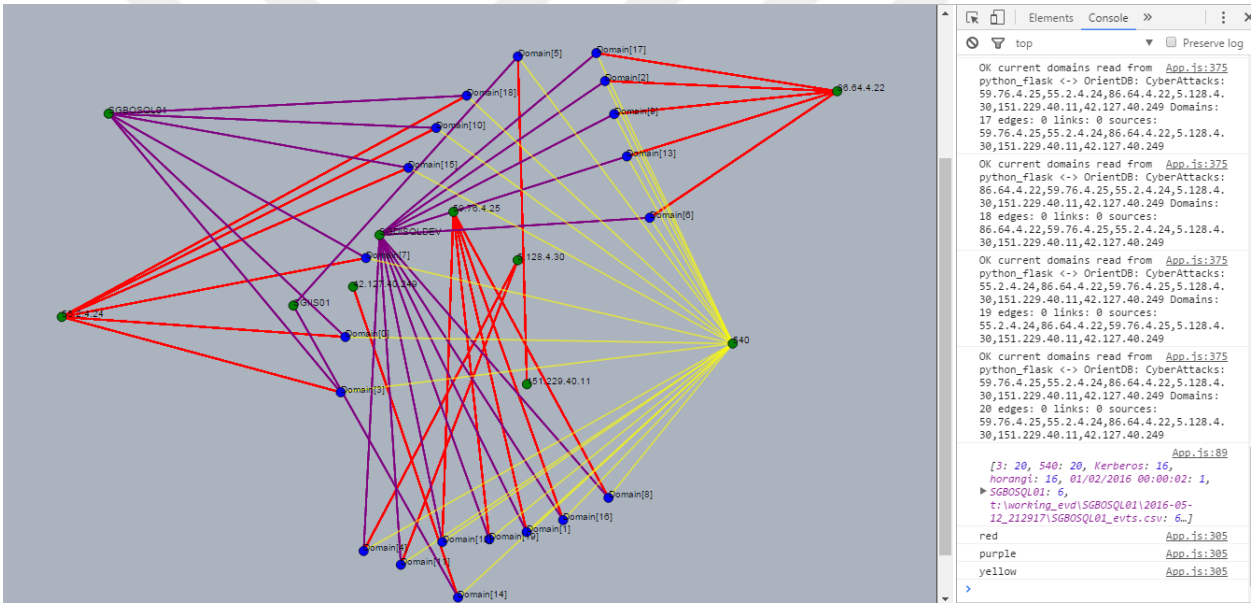


Figure 5.4: React D3 Relational graph

We have done our research on both network and application layer data. Moreover, the previous works on the application layer data in the paper titled" A Model for Anomalies Detection of different attacks Using Inverse Weight Clustering and Decision Tree" used only a single machine learning algorithm whereas we are using five machine learning algorithms and comparing the accuracy. Furthermore, paper [41] worked on application layer data.

6. CONCLUSION

As introduced before, as the conclusion, it should be interesting to analyze the behavior of the fast correlation-based filter for a different sample of the Kyoto dataset or other available benchmark datasets as KDD99 or NSL-KDD. Also, a multi-class classification should be tried (instead of a binary one) to distinguish known and unknown attacks. To do so, an over-sampling of the instances which represent unknown attacks should be performed to ensure an improved training of the machine. This procedure would theoretically enhance the classifier's recall.

Another topic for future work would be the usage of raw data, which would require extensive feature construction work before the application of the algorithms studied in this thesis. In this scenario, instance selection should be incorporated as a preprocessing procedure to guarantee an even smaller dataset. Besides, a user-friendly visualization platform should be included to provide human cybersecurity experts a closer observation of the most valuable variables.

REFERENCES

[1] K. Gopalakrishnan, M. Govindarasu, D. W. Jacobson, and B. M. Phares. Cybersecurity for airports.International Journal for Traffic and Transport Engineering, 3(4):365–376, 2013.

[2] Internet Security Threat Report 22. Symantec, April 2017.

[3] W. Li and P. Kamal. Integrated aviation security for defense-in-depth of the next-generation air transportation system. In IEEE International Conference on Technologies for Homeland Security, Waltham, MA, November 2011. IEEE. 10.1109/THS.2011.6107860.

[4] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam. Cyber threat analysis and modeling of an unmanned aerial vehicle system. In IEEE International Conference on Technologies for Homeland Security, Waltham, MA, November 2012. IEEE. 10.1109/THS.2012.6459914.

[5] B. Thuraisingham, L. Khan, and M. M. Masud. Data mining for security applications. In 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, 2018. IEEE.

[6] S. Kumar and E. H. Spafford. An application of pattern matching in intrusion detection. Technical Report 94-013, Purdue University - Department of Computer Science, 2017.

[7] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cybersecurity intrusion detection. IEEE Communications Surveys and Tutorials, 18(2):1153–1176, 2016. doi:10.1109/COMST.2015.2494502.

[8] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Transactions on Computers, 65(10):2986–2998, October 2016. doi:10.1109/TC.2016.2519914.

[9] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, (5):1205–1224, October 2004.

[10] M. A. Aydin, A. H. Zaim, and K. G. Seylan. Hybrid intrusion detection system design for computer network detection. Computers and Electric Engineering, 35(3):517–526, 2009.

[11] C. M. Bishop. Pattern recognition and machine learning. In M. Jordan, K. Kleinberg, and. Scholkopf, editors, Information Science, and Statistics. Springer, 2016. ISBN-10: 0-387-31073.

[12] S. Russel and P. Norvig. Artificial Intelligence - A Modern Approach, pages 693–767. New Jersey: Prentice-Hall, 3rd edition, 2010. ISBN: 9781292153964.

[13] C. Palagiri, A. Bivens, R. Smith, B. Szymanski, and M. Embrechts. Network-based intrusion de- tection using neural networks. In Proc., Intelligent Engineering Systems Through Artificial Neural Networks, volume 12, pages 579–584, 2002.

[14] J. Cannady. Artificial neural networks for misuse detection. In Proceedings of the 1998 National Information Systems Security Conference, Arlington, VA, 1998.

[15] J. Ryan, M.-J. Lin, and R. Miikkulainen. Intrusion detection with neural networks. In AI Approaches to Fraud Detection and Risk Management: Papers from 1997 AAAI Workshop, pages 72–79, Providence, RI, 1997. MIT Press.

[16] A. Tajbakhsh, M. Rahmati, and A. Mirzaei. Intrusion detection using fuzzy association rules. Applied Soft Computing, 9(2):462–469, March 2009.

[17] H. Zhang. The optimality of naive Bayes. In Proceedings of the Nineteenth National Conference on Artificial Intelligence. American Association for Artificial Intelligence, MIT Press, 2004.

[18] L. Koc, T. A. Mazzuchi, and .S Sarkani... Expert Systems with Applications, chapter A network intrusion detection system based on a Hidden Naive Bayes multiclass classification. Elsevier, 2012.

[19] L. Breiman. Random forests. Machine Learning, 45(1):5–32, October 2001. doi: 10.1023/A:1010933404324.

[20] C. Cortes and V. Vapnik. Support-vector networks. AT&T Labs-Research, USA, 1995.

[21] S. B. Kotsiantis, D. Kanellopoulos, and P. .E.. Pinterest. Data preprocessing for supervised learning.International Journal of Computer Science, 1(1), 2006.

[22] R. Bellman. Dynamic Programming. Princeton University Press, 1957.

[23] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. Ameri- can Mathematical Society Conference: Math Challenges of the 21st Century, 2000.

[24] M. Grochowski and N. Jankowski. Comparison of instance selection algorithms ii. In Lecture notes in computer science, June 2004.

[25] M. A. Hernandez and S. J. Stolfo. Real-world data is dirty: Data cleansing and a merge/purge problem. Data Mining and Knowledge Discovery, 2:9–37, 1998.

[26] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. Journal of Artificial Intelligent Research, 11:131–167, 1999.

[27] J. H. Friedman. Data mining and statistics: What's the connection? In Proceeding of the 29th Symposium on the interface between computer science and statistics, 1997.

[28] G. C. Shekar and F. Sahin. A survey on feature selection methods. Computers and Electric Engineering, 40(1):16–28, January 2014.

[29] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Journal of Machine Learning Research, 3, March 2003.

[30] J. Song, H. Takakura, and Y. Okabe. Description of Kyoto university benchmark data, June 2010.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Pretten- hofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[32] J. Song et al., "Statistical analysis of honeypot data and building of Kyoto 2006+dataset for NIDS evaluation", Proc. 1st Workshop Building Anal. Datasets Gathering Exper. Returns Secur., pp. 29-36, 2011.

[33] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In W. W. Cohen and H. Hirsh, editors, Machine Learning: Proceedings of the Eleventh

International Conference, pages 121–129, San Francisco, CA, 2017. Morgan Kaufmann Publishers.

[34] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In Proceedings of the Seventeenth International Conference on Machine Learning, pages 359–366, San Francisco, CA, July 2000. Morgan Kaufmann Publishers. ISBN: 1-55860-707-2.

[35] D. Koller and M. Sahami. Toward optimal feature selection. In Proceedings of the Thirteenth International Conference on Machine Learning, pages 284–292, Bari, Italy, July 1996. Morgan Kaufmann Publishers. ISBN: 1-55860-419-7.

[36] G. D. Tourassi, E. D. Frederick, M. K. Markey, and C. E. F. Jr. Application of the mutual information criterion for feature selection in computer-aided diagnosis. The International Journal of Medical Physics Research and Practice, (28):2394–2402, December 2016.

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical Recipes in C. Cam- bridge University Press, 2nd edition, 2018.

[38] A. Abraham. Handbook of Measuring System Design, chapter Artificial Neural Networks. John Wiley and Sons Ltd., 2015.

[39] M. W. Gardner and S. R. Dorling. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. Atmospheric Environment, 32(14/15):2627–2636, 2018.

[40] R. E. Walpole, R. H. Mayers, S. L. Myers, and K. Ye. Probability and statistics for engineers and scientists. Prentice-Hall, 9 edition, 2017.

[41] K. P. Murphy. Naive Bayes classifiers, October 2016.

[42] A. Liaw and M. Wiener. Classification and regression by random forest. R News, 2(3):18–22, December 2012.

[43] L. Guo, Y. Ma, B. Cukic, and H. Singh. Robust prediction of fault-proneness by random forests. In Proceedings of the 15th International Symposium on Software Reliability Engineering. IEEE, 2004.

[44] J. Bergstra and .Y. Bengilo... Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13:281–305, December 2012.

[45] K. Platform. Explore projects created by others. https://www.kaggle.com/explore-projects.

[46] Ustebay, Serpil & Turgut, Zeynep & Aydin, M.Ali. (2019). Cyber Attack Detection by Using Neural Network. Approaches: Shallow Neural Network, Deep Neural Network, and AutoEncoder. 10.1007/978-3-030-21952-9_11.

[47] Benferhat, Salem & Kenaza, Tayeb & Mokhtari, Aicha. (2018). A Naive Bayes Approach for Detecting Coordinated Attacks. Proceedings - International Computer Software and Applications Conference. 704-709. 10.1109/COMPSAC.2018.213.

[48] Farnaaz, Nabila & Akhil, Jabbar. (2016). Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science. 89. 213-217. 10.1016/j.procs.2016.06.047.