REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical And Computer Engineering

TRAFFIC CLASSIFICATION FOR DDOS ATTACK DETECTION USING MACHINE LEARNING

Ali Hameed Qasim ALMUFADHL

Master's Thesis

Supervisor

Asst. Prof. Dr. Muhammad ILYAS

Istanbul, 2022

TRAFFIC CLASSIFICATION FOR DDOS ATTACK DETECTION USING MACHINE LEARNING

Ali Hameed Qasim ALMUFADHL

Electrical And Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled TRAFFIC CLASSIFICATION FOR DDOS ATTACK DETECTION USING MACHINE LEARNING prepared by Ali ALMUFADHL and submitted on 25/7/2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

.

Asst. Prof. Dr. Muhammad ILYAS

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Muhammad ILYAS	Faculty of Engineering and Architecture, Altınbaş University	
Asst. Prof. Dr. Hasan ABDULKADER	Faculty of Engineering and Architecture, Altınbaş University	
Asst. Prof. Dr. Ali HAMITOGLU	Faculty of Engineering and Natura Sciences, Istinye University	

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information and data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ali Hameed Qasim ALMUFADHL

DEDICATION

This thesis is dedicated to the people who have supported me throughout my education. Thanks for making me see this adventure through to the end.

ABSTRACT

TRAFFIC CLASSIFICATION FOR DDOS ATTACK DETECTION USING MACHINE LEARNING

ALMUFADHL, Ali Hameed

M.Sc. Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Muhammad ILYAS

Date: 08/2022

Pages: 69

Anomaly detection remains a major concern for many Internet players. It is a complex problem, having attracted the interest of both industrial players, such as Internet Service Providers (ISPs), and members of the scientific community. These anomalies, defined as events deviating from the usual behaviour of traffic, can have severe consequences on the services provided. While some anomalies may be the result of configuration errors, it is scans and attacks that are of most interest to digital players We present our problem here, divided into three parts. First, we cover all the things to consider when building a detector. We then address several challenges related to current traffic. Finally, we discuss the specificity of distributed denial of service attacks Thus, this thesis seeks to propose new solutions, taking into account all the components of this problem. we first present the motivations of project, for which our work was carried out. The problem to which we answer is then exposed. then, we introduce our different contributions to answer this problemWe present our problem here, divided into three parts. First, we cover all the things to consider when building a detector. We then address several challenges related to current traffic. Finally, we discuss the specificity of distributed denial of service attacksThus, this thesis seeks to propose new solutions, taking into account all the components of this problem. we first present the motivations of project, for which our work was carried out. The problem to which we answer is then exposed. then, we introduce our different contributions to answer this problem.

Keywords: IDS, ANN, SVM, DDOS

TABLE OF CONTENTS

Pages

ABSTRACT ..... **vi**

LIST OF TABLES ..... **ix**

LIST OF FIGURES ..... **x**

ABBREVIATIONS ..... **xii**

1. INTRODUCTION **1**

1.1 BACKGROUND 1

1.2 PLANNING THE PROBLEM 2

1.3 THESIS OBJECTIVES 4

1.4 THESIS CONTRIBUTION 5

1.5 THESIS ORGANIZATION 5

2. LITERATURE REVIEW **6**

2.1 CHAPTER OVERVIEW 6

2.2 PREVIOUS WORKS 6

2.3 CHAPTER CONCLUSION 12

3. MATERIALS AND METHODS **13**

3.1 BASIC STRUCTURE OF A DDOS ATTACK 13

3.1.1 Security Threats 14

3.1.2 Intrusion Detection System 17

3.2. DATA BASE 19

3.2.1 UNB CIC-IDS 2021 19

3.2.2. UNB CIC-DDoS 2019 20

3.3 ENTROPY OF A DATASET 21

3.3.1 Algorithm for establishing thresholds 23

3.3.2 Time series forecast using the ARIMA algorithm 23

3.4 PRINCIPAL COMPONENT ANALYSIS 24
3.5 ARTIFICIAL NEURAL NETWORKS 25
3.5.1 Parameterization of an Artificial Neural Network 26
3.6 SUPPORT VECTOR MACHINE 28
3.6.1. Parameterization of a Support Vector Machine 29
3.7 INTERNET OF THINGS 31
3.7.2 Intrusion Detection System 32
3.7.3 Evaluation of IDS 34
3.7.4 The different types of IDS 35
3.8 MACHINE LEARNING 40
4. PROPOSED METHODOLOGY 41
4.1 DATA PROCESSING 41
4.2 APPROACH BASED ON THE ARIMA METHOD 43
4.3 SUPERVISED APPROACH 44
4.4 COMPUTATIONAL RESOURCES 45
5. RESULTS AND DISCUSSIONS 47
5.1 ARIMA METHOD 47
5.2 SUPERVISED LEARNING METHOD 51
6. COCNLUSIONS AND FUTURE WORK 54
6.1 CONCLUSION 54
6.2 FUTURE WORK 54
REFERENCES 55

LIST OF TABLES

Pages

Table 3.1: The data streams obtained through the collection system provide 78 attributes for each row of data. 20

Table 3.2: The data streams obtained through the collection system provide 87 attributes for each row of data. 21

LIST OF FIGURES

Pages

Figure 1.1: Structure of DDoS attacks [1] 2

Figure 1. 2: DDoS attacks growth by year [5] 3

Figure 3.1: shows the basic topology of a DDoS attack where the attack is distributed to various electronic devices (*laptops, desktops, smartphones, servers*) whose purpose is to harm the services provided by the server. 13

Figure 3.2: Entropy filtering for the DDoS attack 22

Figure 3.3: Diagram of the ARMA algorithm 24

Figure 3.4: Data reduction using PCA 25

Figure 3. 5: represents only one neuron, with activation function *f* 26

Figure 3. 6: ANN classification 28

Figure 3.7: shows a simple example of classifying geometric shapes into two distinct classes 29

Figure 3.8: RBF kernel with different gamma values 31

Figure 3.9: Types of evaluation criteria for IDS 34

Figure 3.10: types of intrusion detection system 36

Figure 3.11: types of IDS attack prevented by the system 38

Figure 4.1: Data preprocessing step in the IDS system 42

Figure 4.2: shows the flowchart of the proposed algorithm 43

Figure 4.3: shows the data flowchart capable of summarizing the processes implemented in this approach 45

Figure 5.1: shows the threshold just after the tenth window. That is, from this window the system considers it as an attack, which ends after the thirty-seventh 48

Figure 5.2: Window Entropy Graph and threshold (3,3,1) as a function of time On the other hand 49

Figure 5.3: shows that the system considered attack situations, starting from the twelfth window, which extended to the thirty-sixth window..... 50

Figure 5.4: Window Entropy Graph and threshold (2,3,3) as a function of time 50

Figure 5.5: shows the execution of the scripts. The ML block refers to the different learning methods applied 52

ABBREVIATIONS

AI	:	Artificial Intelligence
FSK	:	Frequency Shift Keying
IoT	:	Internet of Things
PAN	:	Personal Area Network
OFDM	:	Orthogonal Frequency Division Multiplexing
TDMA	:	Time Division Multiple Access

1.INTRODUCTION

1.1 BACKGROUND

Anomaly detection remains a major concern for many Internet players. It is a complex problem, having attracted the interest of both industrial players, such as Internet Service Providers (ISPs), and members of the scientific community. These anomalies, defined as events deviating from the usual behaviour of traffic, can have severe consequences on the services provided. While some anomalies may be the result of configuration errors, it is scans and attacks that are of most interest to digital players. These anomalies are generally much more difficult to eliminate since they are the result of a real desire to harm. They then depend directly on the motivation and resources of the attacker. Among these attacks is the famous Distributed Denial of Service (DDoS) attack.

This type of attack consists of many sources of traffic, distributed in several places on the Internet, sending a very large number of fake requests to a target system. Exhausted by these requests, the target system can no longer provide an efficient service. These attacks can have serious consequences. Thus, it is then necessary to guard against such anomalies. Especially since such approaches make it possible to pool resources and bring together detection tools at a single point. Nevertheless, network anomaly detection is a problem with many components. It requires tools capable of operating in real time, autonomously, on high-speed, disparate and constantly changing traffic. Added to this is the changing nature of the anomalies to be detected, which can defeat the detection tools. To all these considerations are added industrial constraints, due to the technological capabilities of current equipment.

In particular, when the detection must be performed at the network level, the equipment used (routers or switches) use their resources first and foremost for traffic management. The remaining capacities are therefore limited, forcing the design of detection methods that do not consume a lot of computing resources. Also, current surveillance architectures operate on sampled traffic which certainly limits the use of resources necessary for detector operations, but can lead to degraded detection. [2].

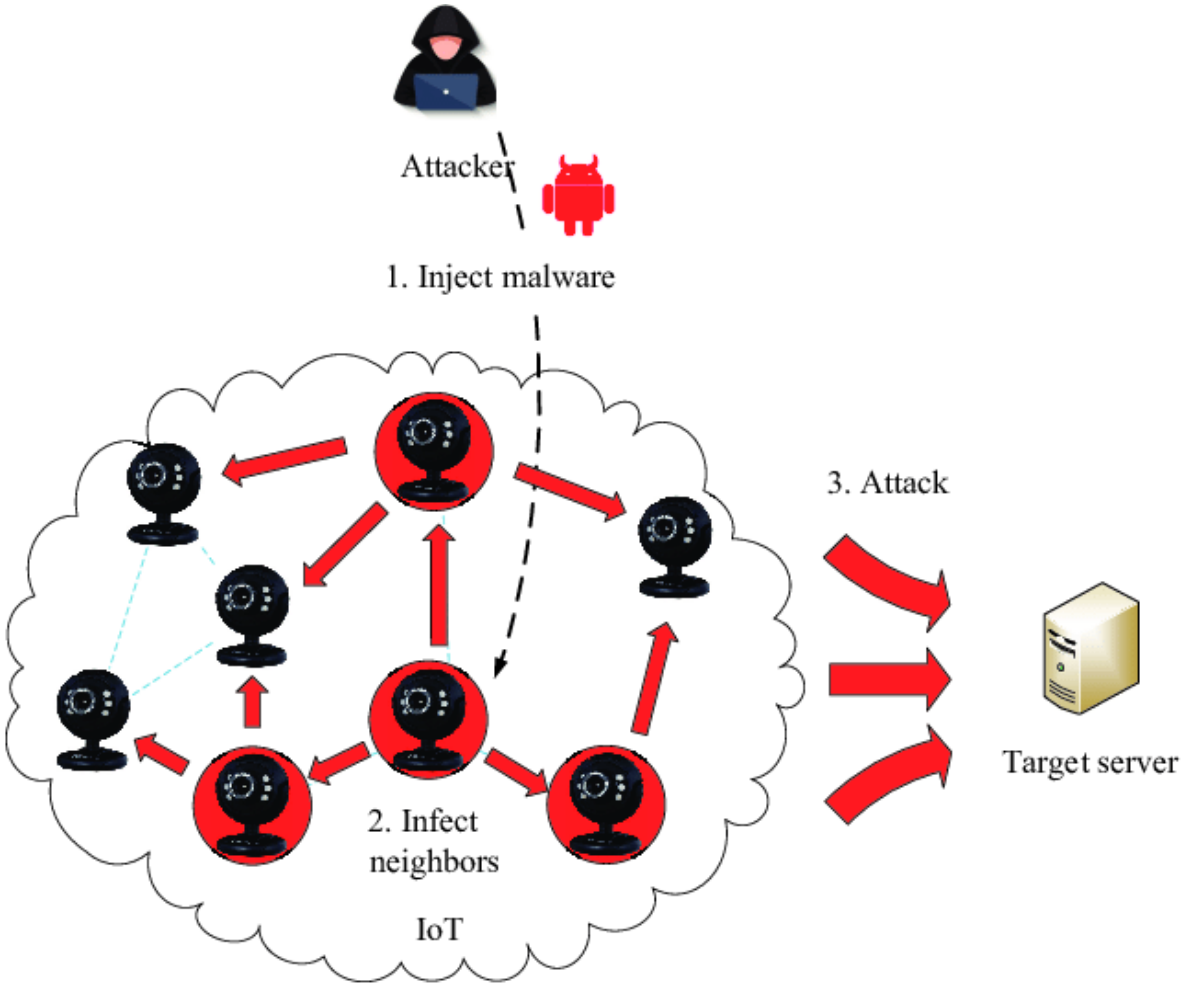


Figure 1.1: Structure of DDoS attacks [1]

1.2 PLANNING THE PROBLEM

We present our problem here, divided into three parts. First, we cover all the things to consider when building a detector. We then address several challenges related to current traffic. Finally, we discuss the specificity of distributed denial of service attacks [8].

The architecture of the Internet was originally designed with openness and scalability in mind. Despite its success in these areas, several security problems were not originally foreseen [9] such as the Distributed Denial of Service (DDoS), which is a type of attack that aims to disrupt a service provided by a host, causing users to not be able to access the offered service. With the extreme ease of access to automated tools that perform the construction of networks of compromised machines and the tasks required by the attacker, the number of attacks belonging to this category

has increased significantly. Software-Defined Networking (SDN) is an emerging networking architecture that physically separates the data plane from the control plane, which is directly programmable [10].

Network Function Virtualization (NFV) transforms how operators design their networks by implementing network functions, such as firewalls, CDNs and message routers, in software that runs in virtualized environments running on general-purpose hardware [11] With the emergence of SDN/NFV based infrastructures, it has become possible to monitor and manage the network in a logically centralized way, migrate network functions, obtain easier modification of network functions and more flexible management of network functions compared to to the use of dedicated physical devices (known as middleboxes). Consequently, networks based on SDN and NFV infrastructures allow a refactoring of mitigation techniques, improving aspects of flexibility and elasticity of defense mechanisms [12] This work will investigate approaches that allow the improvement of such aspects.

Many parameters must be taken into account when choosing an anomaly detector. If the quality of the detection is of course to be taken into account, the use of a detector also requires the mobilization of resources, whether they are related to the work of an administrator or to the infrastructures necessary for the execution of the detector. Each solution then requires a true cost-benefit analysis. Completeness and robustness. The usefulness of a detector is first of all conditioned by its ability to detect anomalies, efficiently and unconditionally. The quality of the detection is therefore one of the parameters to be taken into account when choosing a solution.

The sensitivity (or rate of true positives) of a detector corresponds to its ability to raise an alert when an anomaly is present in the traffic. An ideal detector is then very sensitive, detecting all the anomalies present in the traffic. Also, a usable detector must be able to operate in real time on the traffic while detecting anomalies at the earliest, before they cause irreparable damage to the targeted system. Finally, it must be robust, able to operate in all situations. In particular, a peak in use must not disturb the proper functioning of the detector. In the event of an attack, the detector must remain operational as much as possible, at least until a countermeasure can be put in place.

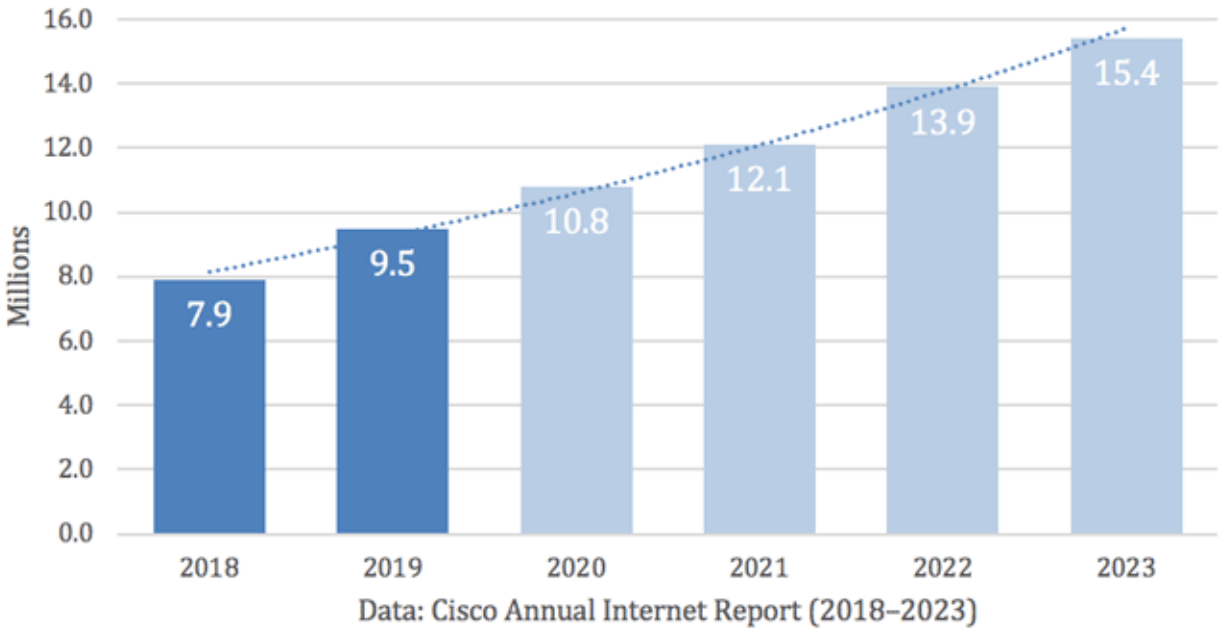


Figure 1. 2: DDoS attacks growth by year [5]

1.3 THESIS OBJECTIVES

Given the limitations presented in the literature and the observed technical need, the objectives of this work are detailed below.

- • The work intends to act in the approach of mitigation mechanisms, which specifically explore IOT characteristics. To carry out these tasks, the following items will be carried out
- a literature review with the objective of understanding and enumerating the main types of attacks, as well as understanding the fundamental characteristics present in the chain of a DDoS attack
- the development of a taxonomy regarding DDoS attack mitigation mechanisms that illustrates the main techniques and strategies currently used
- the elaboration of a prototype that will allow the accomplishment of experiments and analysis of the results of the proposed strategies
- comparison between a subset of evaluated strategies focusing on mitigation strategy response time metrics and service availability for non-malicious users;

1.4 THESIS CONTRIBUTION

Thus, this thesis seeks to propose new solutions, taking into account all the components of this problem. we first present the motivations of project, for which our work was carried out. The problem to which we answer is then exposed. then, we introduce our different contributions to answer this problem.

This requires their detection and characterization using effective tools. Some detectors operate at a host level, an ideal approach for detecting anomalies with an impact limited to a single machine. However, some attacks, including Distributed Denial of Service attacks, can completely saturate the targeted system. In some cases, saturation can occur downstream, at the entrance of the network for example, which makes host-level mitigation unnecessary. This is why detection at the network level becomes necessary

1.5 THESIS ORGANIZATION

In Chapter 2, a survey of past works and the chain of a cyber-attack, the chain of a DDoS attack, and the classification criteria for DDoS attacks are presented. It also introduces the IOT infrastructure and details of how the OpenFlow protocol works. Finally, the advantages, architectural characteristics and the relationship of IOT are presented. Chapter 3 describes how mitigation strategies based on Rate Limiting, Filtering, and Reconfiguration tactics can be used in networks with IOT-based infrastructures. In Chapter 4, the strategies proposed for mitigating DDoS attacks are implemented and the results obtained are analysed. In Chapter 5, related works are presented. Finally, Chapter 6 presents the conclusions reached through the development of the work, with a summary of contributions and future work.

2.LITERATURE REVIEW

2.1 CHAPTER OVERVIEW

In this Chapter, state-of-the-art works on the detection and mitigation of denial-of-service attacks on Internet-connected networks are presented and discussed. More specifically, detection techniques, data processing approaches, deployment strategies and reaction time to attacks are analyzed. In the end, the most recent works are put into perspective with this thesis in order to highlight the most relevant contributions

2.2 PREVIOUS WORKS

- Anderson (1980), in the course of his research on the subject of computer security monitoring and surveillance, published an early report on the monitoring and surveillance of threats to computer systems. On the basis of the needs and definitions that have been provided by this research, it is now possible to design tools that can detect anomalies in this kind of system.
- A few years later, in 1987, Denning expanded these definitions and developed the first known model of an Intrusion Detection System, which was referred to as SDI. Denning (1987) developed a framework for an expert system that can detect general purpose intrusions that is independent of the system, application environment, vulnerability, or information kind. In spite of the fact that this paradigm continues to inspire modern SDIs to varying degrees, the rapid advancement of technology has altered the behavior of threats and systems alike, which necessitates ongoing research in the field of information security. Despite the fact that this paradigm continues to inspire modern SDIs to varying degrees, Studies on denial of service began to emerge at the same time as vulnerabilities in computer systems and networks became more commonplace.
- Gligor (1984) established the definition of "denial of service" in operating systems by basing it on the amount of time that groups of users spent using available resources. The prevention of DoS was the subject of a later study written by Yu and Gligor (1990). In addition, Schuba et al. (1997) investigated the network-based denial-of-service attack known as SYN flooding. They discussed and provided preventive strategies to combat it, such as firewalls and other methods. Axelsson (2000) forecasted that denial of service

(DoS) attacks would grow increasingly widespread; nonetheless, the frequency of these kinds of assaults has consistently outpaced predictions over the years. Because of the growth of networks, the introduction of new applications, and the increase in bandwidth, there has been an increase in the number of new kinds of assaults. These findings led to the development of specialized approaches for the detection and mitigation of distributed denial-of-service attacks in computer networks.

-
- For instance, Chang (2002) emphasized the complexity of DDoS attack methods and tools, which not only proved more effective but also made it more difficult to track down the actual attackers. In his essays, the author expressed concern about the difficulty of defending against attacks on a broad scale. Nevertheless, in order to be considered comprehensive, a solution needs to incorporate not only prevention and preemption (which take place before the assault), but also detection and filtering (which take place during the assault), tracking, and source identification (during and following the attack). According to the author, who discovered that the solutions examined in his study were clearly insufficient to protect Internet nodes against DDoS attacks, this problem may never be solved. He found that the remedies studied in his study were clearly insufficient to protect Internet nodes against DDoS attacks.
-
- Prior to the work of Mirkovic and Reiher, there was already a taxonomy of DDoS attacks and defense measures (2004). According to the authors, given that the Internet is managed in a distributed form, distributed denial-of-service attacks raise a variety of security vulnerabilities that need to be addressed.
-
- When it comes to attack mechanisms, Mirkovic and Reiher (2004) came up with eight main categories, ranging from the type of victim (an application or device), the impact on the victim (disruptive or degrading), and three categories of mechanisms. of defense: by activity (preventive or reactive), by cooperation, and by the place of emplacement. These categories are broken down as follows: the type of victim (an application or device); the impact on the victim (disruptive or degrading); and the
-

- In order to augment the work that Mirkovic and Reiher (2004) did, they consulted material that had been examined by experts, and they conducted an in-depth investigation of protection systems against DoS and DDoS attacks. Peng et al. (2007) suggest that the global network infrastructure should be made more reliable in order to protect against denial-of-service attacks (DoS). According to the authors, malicious users have to be hunted down and penalised. The writers are of the opinion that more stringent measures are required to validate the provenance of Internet communications. However, this is a tough task to accomplish because of the nature of the Internet and the tools that attackers employ to mask and falsify origin. Concerns about the privacy of network users also make this a challenging endeavor. The authors believe that preventing distributed denial of service attacks requires collaboration amongst people all around the world. On the other hand, accomplishing this objective is difficult since there are not enough financial incentives for users to participate in security measures, in particular those that protect third-party networks. Research conducted by demonstrates that efforts to find solutions continued for many years after the fact.
-
- • Cambiaso et al. (2012) in their taxonomy of slow application layer attacks, in addition to Zargar et al. (2013) and Wang et al. (2013)'s systematic reviews of DDoS Flooding attacks; (2015).
-
- According to Chang (2002), the most important challenge in thwarting distributed denial of service attacks is the early identification and thwarting of attacks as close as is practical to their point of origin. This highlights the necessity of systems cooperation. This was demonstrated by Zargar et al. (2013). However, a comprehensive solution that takes into account each of these concerns has not yet been put into effect. The Internet of Things (IoT) is becoming increasingly widespread, which means that it is exposing an increasing number of devices, services, and security holes in the networks that we use.
-
- according to Al-Fuqaha and other authorities (2015). As is the case in many other Internet-related scenarios, these devices can serve as a jumping off point for assaults

directed at particular targets. Multiple studies have been conducted on the topic of denial-of-service attacks and have been published in the academic literature; nevertheless, when looking at the distribution and characteristics of these studies across time, it is clear that this research field is not yet fully studied. It is abundantly obvious that advancements in technology and societal norms have an effect on the dynamics of assaults and the defense systems, both of which need to be continuously improved and adapted to accommodate for new circumstances. Our study contributes to the development of safer networks as a consequence of the presentation of a system to combat potential threats posed by denial of service attacks.

-
- Detecting flood assaults on web servers through the use of data sampling that is based on HTTP was a proposal that was made in 2017 by Jazi et al (Hypertext Transfer Protocol). The authors employed the variables "number of application layer requests" and "number of packets with a payload size of zero" to mathematically establish the legitimate traffic profiles as well as the DoS attack profiles. After criteria were established, the CUMSUM technique was put to use in order to categorize incoming traffic as either normal or a DoS attack. This system was evaluated using both real and simulated traffic, and the results were based on data from CIC-DOS. The researchers were able to identify 80–88 percent of the risks by using a sample rate that was only thirty percent of the total traffic. The variables that are used to characterize traffic are susceptible to attack since attackers can easily alter packets with a payload of zero size notwithstanding the encouraging findings of the proposed method. This leaves the variables open to manipulation. When the industry's typical sample methods were used, such as Random packet sampling and Random n-out-of-N packet sampling, there was an increase in the number of false alarms that were generated.
-
- Using metrics like generalized entropy (GE) and generalized information distance (GID), a collaborative defense system known as D-FACE (Behal et al., 2018) has been built to detect a variety of DDoS attacks and Flash Events (FE). Take, for instance, a volumetric distributed denial of service attack, which occurs when hundreds of legitimate users attempt to access the same computing resource, like a website, at the

same time. A combination of real and simulated systems was used in the experiments that were carried out. According to the findings, D-FACE was successful in identifying both online DDoS attacks and FEs. In order to validate the results, both the datasets from MIT Lincoln Laboratory and data from the real world were employed. Despite the fact that it contains relevant contributions, it does not appear that any strategy to sampling network traffic was used in this work. As a direct consequence of this, some of the datasets that were utilized in the past have been found to be irrelevant and removed from use. The proposed plan has been received with criticism from the company for these reasons, in particular because it requires a high degree of commitment from ISPs, particularly in terms of local privacy and security regulations.

-
- The SkyShield system was proposed by Wang et al. (2018) for the detection and mitigation of DDoS assaults at the application layer. SkyShield is able to identify irregularities on attacking hosts by exploiting the difference between the sketch tables S1 and S2 of two different sketch tables (Sketch). In each discovery cycle, the input keys that are used are the source IP addresses of any and all requests that are received. The findings of S1 are written down in a backup sketch for S2 before the conclusion of the typical detection cycle. Following the conclusion of a detection cycle, the d(S1, S2) divergence is calculated. If the difference between S1 and S2 is greater than a specified threshold known as t, then an assault warning will be triggered. As a direct result of this, S2 will no longer receive any updates. Protective methods taken during the mitigation phase include filtering, whitelisting, blacklisting, and the usage of CAPTCHA. In order to conduct an analysis of the system, we compiled our own sets of data. The proposed solution from SkyShield is susceptible to flooding at both the network and transport layers because it is dependent on HTTP. Due to the fact that they process all network data, detection systems have the potential to create a bottleneck in large-scale attacks.
-
- A patent has been issued for a method of communication that is centered on BGP (Border Gateway Protocol) communications and was proposed by Dousti and others (2018). In this conception, the controller router is not responsible for providing detection; rather, it is responsible for establishing a protocol for the communication between routers. As

soon as it obtains the detection information, the controller starts the mitigation process and immediately begins notifying the other routers about the attack. Even though it is impossible to assess the success of this proposal in operation or how communication with the detection system occurs in practice, this idea is important because it offers the possibility of mitigating attacks at the network's core. Even though it is impossible to assess the success of this proposal in operation or how communication with the detection system occurs in practice.

-
- • The patent application that was submitted by Jalan et al. includes a description of a local TCP SYN flood attack mitigation system (2018). These cyberattacks on telecommunications networks have led to the implementation of a tool for mitigating their effects called TCP RESET messages. Due to the fact that it is only concerned with one type of intrusion, this strategy has extremely limited capabilities for both detection and defense against intrusions. An attacker might or might not process a TCP RESET message that was sent by the victim. This could happen either way. Hosseini and Azizi (2019) classified DDoS assaults with the application of a supervised machine learning algorithm. The authors put out a hybrid framework that contains a variety of classifiers, and it was tested using the NSL-KDD (Tavallaee et al., 2019).
-
- • Aamir and Zaidi (2019) conducted their research using a methodology known as semi-supervised. In both instances, offline analysis and evaluation of datasets were carried out with the use of machine learning. In order to protect the Umbrella system from a wide range of DDoS assaults, multi-layered defense architecture has been incorporated into the system by Liu et al. (2020). The writers offered a technique that was purely based on the safeguarding of the victim and determining who they were. The algorithm for the system decides what constitutes acceptable rates of packet loss and what constitutes fair sharing for each individual user. Mitigation is carried out by utilizing the filters of the firewall and the QoS rules. On a specialized test platform, the Umbrella system was put through its paces in terms of traffic control in order to evaluate its effectiveness.
-

- The authors assert that the system is capable of withstanding assaults on a massive scale. The problem with this method is that despite the fact that it is widely used in the industry, there is little evidence to suggest that it is effective against truly massive DDoS attacks. Research on denial-of-service attacks is now being conducted in many different types of network environments, including cloud computing, software-defined networking (SDN), and the Internet of Things (IoT) (Mohamad Noor & Hassan 2019). (Phan & Park 2021).).

2.3 CHAPTER CONCLUSION

Early detection, prevention and mitigation of attacks as close to the source as possible remain the main goals of researchers. However, new challenges were added, among them: the sophistication of the attacks and the large volume of data that needs to be processed. In addition, the proposed solution must have a viable implementation, considering both current and future technology. To this end, researchers investigate strategies such as machine learning-based detection, sampled traffic analysis, SDN-based systems, and cloud computing as ways to solve the DDoS problem.

3.MATERIALS AND METHODS

This chapter presents the main theoretical concepts for the development of the two proposed algorithms, as well as the concepts necessary for a fair comparison between them. First, it is important to understand the structure of a DDoS attack and study the structure of the databases that will be used to test the algorithms. Then, some important topics for the development of the algorithm will be presented: entropy in a dataset, use of the ARIMA model for thresholding processes, dimensionality reduction techniques and, finally, classifiers for supervised learning.

3.1 BASIC STRUCTURE OF A DDOS ATTACK

Although this is not the scope of this work, it is important to have a basic understanding of the taxonomy of a DDoS attack. A typical infrastructure for launching DDoS attacks is shown in Figure 3.1 where the criminal uses compromised machines as a mirror for the attack, increasing the efficiency of the attack by repeating requests to the victim of the attack [12].

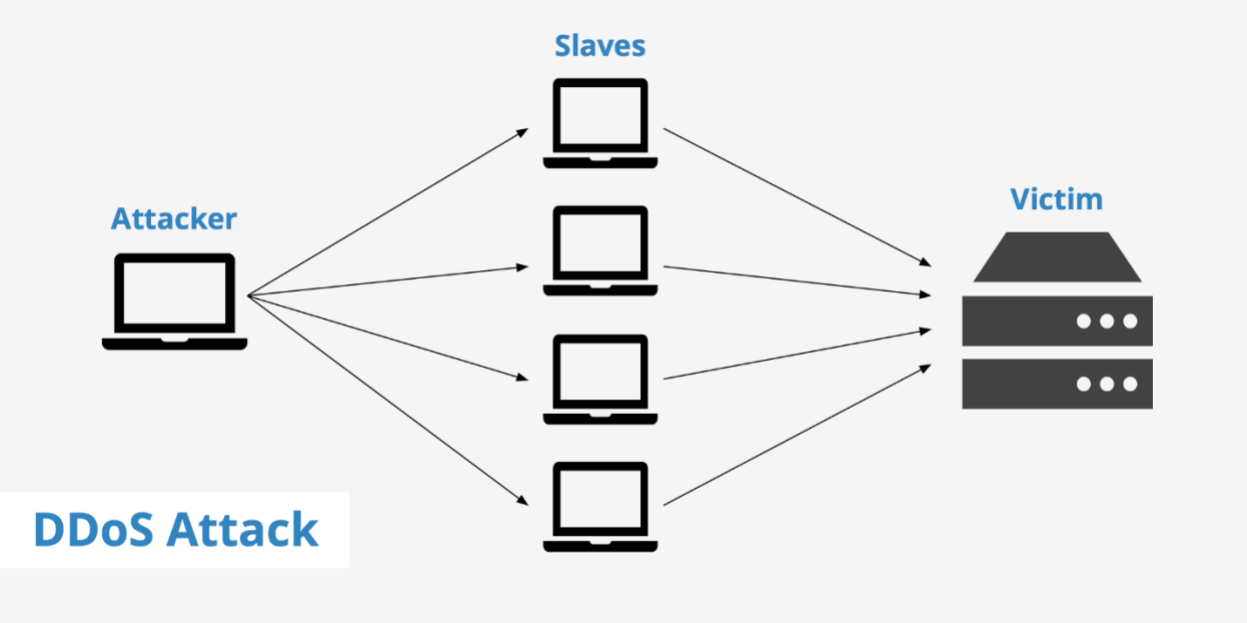


Figure 3.1: shows the basic topology of a DDoS attack where the attack is distributed to various electronic devices (*laptops, desktops, smartphones, servers*) whose purpose is to harm the services provided by the server.

These types of attacks have several motivations: competition between companies, politics, extortion and even cyber wars.

3.1.1 Security Threats

The rapid development of the connected objects market is forcing manufacturers to quickly offer consumers many innovative products. Thus, the security considerations of new objects placed on the market are not sufficiently explored, which creates greater risks for users. Moreover, with the heterogeneity of the proposed solutions, it is complicated for security researchers to address security issues for all smart objects. Conti distinguishes between security challenges related to connected objects and those related to their digital investigation Thus, the author explains that it is necessary to ensure authentication, authorization and access control. It is also necessary to secure user privacy. According to him, this must be done with a secure architecture. Regarding digital forensics, he asserts that the issues related to the identification, collection and preservation of evidence are different from traditional computer systems. It is necessary to reflect on these issues in the context of connected objects in order to be able to analyze the evidence, correlate it and be able to trace the attacks that have occurred. Thus, many authors are concerned about the general lack of security of objects placed on the market and the threats that this introduces to users. the security of objects and respect for the privacy of users appear to be fundamental criteria for the adoption of these objects in homes. The last author identifies the dangers associated with the use of these objects, while Babar makes a taxonomy of threats associated with the emergence of the Internet of Objects (Babar et al., 2010). It identifies five areas of danger for smart objects:

- Identification management: this area includes all the security techniques that must be implemented to uniquely identify objects, users or sessions. It is necessary, according to the author, to be able to guarantee a high level of security of authentication, authorization and account management methods.

- Communications security: this principle includes all the measures aimed at ensuring the availability of communications.

- We could have added to this the integrity of these communications, which should not be able to be modified, as well as the confidentiality of these to avoid putting the privacy of users at risk.

- Physical threats: in this category, the author explains that physical access to objects is most of the time very easy, and that this makes it easier for attackers to perform attacks such as reverse engineering of the system or microprobing attacks to find encrypted files

The author does not explain that connected objects can compromise the physical security of their users, this point being detailed below.

- Security of embedded systems: the author claims that embedded systems will be subject to attacks similar to traditional systems at the physical and MAC levels. For example, attacks by side channels or against secure environments can be conducted against these objects.

- Management of information storage: this principle is essential in the management of cryptographic keys. It is also important to ensure the integrity and confidentiality of the data present on the objects.

Various works have been proposed to ensure the security of data storage in the case of connected objects, such as that of Babar which proposes a global framework for securing such systems. Few articles deal with the risks introduced by connected objects on users, whether it is their physical security or respect for their privacy. Thus, the interconnection of more and more objects such as connected ovens, intelligent alarm or smoke detection systems, or even gas control systems capable of being operated remotely are becoming more democratic. However, if an attacker obtains sufficient control over these objects, for example by acquiring full access to the home automation controller, it would then be possible for him to open the gas supply and then suddenly turn on the oven, thus possibly causing a explosion and physical damage to the habitat or its inhabitants. Similarly, respecting the privacy of users of smart objects involves securing personal information, such as audio tracks or photos recorded by cameras. It is also necessary to protect the information emitted by the various sensors that may be found in the houses (temperature, movement, light sensors) since they could allow an inhabitant to know the habits of the inhabitants of the house or even to determine their activities. These threats are based on the different waves of attacks involving connected objects. Saxena thus identifies several attacks targeting smart objects while

Sikder is interested in attacks specifically targeting sensors in networks of connected objects Similarly, Babar performs a taxonomy of the attacks encountered and Kozlov shows different attack scenarios depending on the architectures developed We can summarize the different works of these authors according to the following different principles:

- data exfiltration, with the aim of spying on users, stealing information from residents or modifying it. For example, the attacker might want to blackmail the victim, steal secret documents to resell them, or alter documents to deceive his target.

- attacks on network messages: the attacker wishes to modify the behavior of the home automation system or even destroy it, to do this he can send false information to connected objects (whether it is erroneous sensor data or orders to actuators), perform replay attacks where it will capture commands sent over the network and replay them or use person-in-the-middle techniques.

the unavailability of systems: the malicious user aims here to make a system inaccessible to its users. This system can be a connected object in itself or a remote server. An example of such an attack where the target was a smart object is the which aims to delete all files of the object it infected and corrupt the memory so that it is not able to be repaired. A famous infection wave targeting a remote server is the botnet campaign which infected smart objects by attempting default passwords. Once the objects were infected, they could be controlled via a remote server to launch distributed denial of service attempts against remote targets, such as the servers of the DNS provider DYN or the host OVH. It can even create financial risks for owners of connected objects, with waves of ransomware forcing consumers to pay a ransom to be able to decrypt the software of their infected objects and thus be able to use them again intrusion into a network: whether it is a home or business network, the connected objects belonging to it are often less protected than traditional systems, allowing attackers to have under their control a first piece of equipment of such a network so that others can then be targeted. Thus, pivoting techniques can be performed once an object is controlled, sometimes allowing network protections such as external firewalls to be bypassed.

3.1.2 Intrusion Detection System

To overcome all these threats, many works have tried to find effective ways to detect intrusions as soon as they occur. Such devices are called IDS, for Intrusion Detection System, and can identify attacks using network traces, it is then a NIDS (Network-based IDS), or using information on the monitored system is then a HIDS (Host-based IDS). IDSs have been studied for a long time, as Debar explains in his taxonomy on intrusion detection systems and these systems have been deployed in all types of networks as shown by Anandvalli with mobile ad hoc networks says traditional IDSs are not suitable for the IoT because objects have limited resources that make it impossible to use heavy-duty intrusion detection technologies. In addition, the protocols used by these objects are numerous and specific to them, and the objects often evolve in mesh networks, being both a source of data and a system capable of redirecting the data it receives. It is these constraints that have led to new research on IDS in the context of smart objects. However, most of these are based on network information (NIDS), as explained by Sabahi in his literature review on the subject can be explained by the fact that this information can be easily obtained without having to modify the monitored object, which can be complicated with embedded systems and even reduce their performance. Sabahi explains that there are two types of intrusion detectors: those based on rules about the system, and those based on the discovery of system anomalies The use of an expert system, which is one of the means of carrying out intrusion detection with rules, can be based on known signatures of threats, on rules concerning the data collected or on the state transitions of the system. This type of intrusion detector has the advantage of being precise and working very well on known threats,but it is unable to detect new threats, unlike the second technique. Indeed, anomaly-based IDSs monitor the behavior of the system they are protecting, and attempt to detect any suspicious variations in that behavior. They therefore require the ability to precisely characterize the behavior of the system they are monitoring, which can sometimes be complicated and require a lot of resources. The various anomaly detection techniques are explained in the following subsection. Sabahi also points out that hybrid methods, using both data from network traffic and characteristics of the monitored host, have been proposed. The architecture of intrusion detectors can also vary greatly, depending on the type of system monitored, the resources needed to detect the intrusion or the data used. Thus, Sabahi explains that centralized IDSs are the most common, but distributed and hybrid architectures have been proposed. In a centralized architecture, the system that is monitored is the one responsible for detecting anomalies, for

example in the case of SNORT where the analysis of network frames is done on one or more nodes of the network, which are the systems that will collect the information necessary for intrusion detection. On the other hand, distributed architectures are essentially based on agents that will collect information on various hosts and route this information to another system that will take care of detection. An example of such a system is DIDMA, the solution offered by Kannadiga, which consists of static agents on the hosts to be monitored, which transmit information to mobile agents using an agent dispatcher These mobile agents then transmit the information to alert agents on the IDS console who will then raise the alerts when necessary. As the various articles have pointed out, intrusion detectors are only as effective as the means of collecting the information they need. Thus, several works on HIDS have relied on system calls to detect malicious behavior, However, several articles have shown that this information is not sufficient to prevent attackers from carrying out intrusions without being noticed. Thus, Wagner's work shows how it is possible to evade intrusion detectors based on system calls, and particularly when the arguments of these kernel functions are not taken into account by the intrusion detector. Its method is to imitate the normal behavior of a system by performing illegitimate actions on it. Similarly, Ma points out that it is possible to evade detection from most HIDS by using child processes that will perform few actions, but that the sum of these actions can lead to advanced attacks

One of the means used to acquire the information necessary for detection is the use of tracing techniques, which not only allow to have the list of system calls made, but also their arguments and information at different levels, when the we trace the kernel, such as network information, the sequence of the actions of the scheduler or even information on the use of the processor. Moreover, Murtaza explains that traces at the kernel level make it possible to acquire much more useful information for intrusion detection Tracing techniques are detailed in the corresponding section. However, the uses of intrusion detection systems have evolved with the arrival of new computer applications. With regard to cloud computing, very precisely the different intrusion detection techniques that can be used in this context The author explains that the traditional systems (HIDS and NIDS) can be used, but that new systems have appeared. This is the case of distributed intrusion detection systems (DIDS) which are made up of several IDSs placed in a large network and which communicate with each other (whether with a central server or directly with each other). These systems can combine both HIDS and NIDS. In addition, intrusion detection systems based on virtual machine hypervisors have also been developed Such systems

thus make it possible to monitor the exchanges of information between the virtual machines, between the virtual machines and the hypervisor and even with the hypervisor of the virtual network, according to the author. Likewise, the democratization of smart phones, and particularly of the Android operating system, has created new threats for users. This has justified various works on intrusion detection on such devices. Thus, Borek listed the various intrusion detection techniques employed for objects running Android. He thus explains that several works detect illegitimate applications according to their source code.

3.2. DATA BASE

The main object of study of learning algorithms is data. There is no point in developing the perfect algorithm for pattern recognition if the database provided for its operation does not provide enough information for the application.

As a result, one of the main steps for the development of any learning algorithm is the study of the data that will be used to train the learning models. In this way, two approaches are possible to choose this very important resource, namely: the generation of data is also the scope of the algorithm development or working with existing databases, focusing on the development in the subsequent stages of capturing information.

As this work aims to develop prediction models and not capture the network data that are necessary for the development of the model, we chose to use the databases, already used in academia, UNB CIC-IDS 2021 [13] and UNB CIC-DDoS 2019 [14].

It is important to note that data collection for analysis and development of intrusion detection systems (IDS)*Intrusion Detection System*) needs to happen constantly. This need is the result of the change in traffic data over time and the new types of attacks that are developed. In addition, old data, in addition to not translating the current reality, may suffer from the lack of important attributes.

3.2.1 UNB CIC-IDS 2021

This database contains benign data, from common traffic, and data from the most different types of attacks, gathered in files in PCAP format. In addition, this database includes an analysis file

in CSV format, with data streams classified as benign or malignant, sending and destination IP and even flow time, obtained from the software. *CICFlowMeter* [15] [16].

This database was generated from the system*B-profile*[17] proposed by Sharalfadin *et al.*in order to characterize human behavior generating benign traffic data. The database is composed of the behavior of 25 common users based on HTTP, HTTPS, FTP, SSH and e-mail protocols.

The system was implemented from July 3, 2021 to July 7, 2021. In this time frame, several attacks were implemented:*Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS*. However, as this work seeks to detect only DDoS attacks, only the network flow file of this type of attack was used. Table 1 shows the characteristics of the file used in the implementation of the algorithm.

Table 3.1: The data streams obtained through the collection system provide 78 attributes for each row of data.

Dataset/Network Flows	#Feature	Training		Testing	
ISCX-UNB Saturday	8	85,222	1353	45,889	1353
ISCX-UNB Monday	8	108,945	2451	58,664	1320
ISCX-UNB Tuesday	8	347,308	24,295	187,012	13,083
ISCX-UNB Wednesday	8	339,470	0	182,793	0
ISCX-UNB Thursday	8	255,054	3381	137,338	1822

Note: For both set: left—Benign, right—Malicious.

3.2.2. UNB CIC-DDoS 2019

This database is an exclusive database for DDoS attacks and which even explores the most different types of approaches within this class of attacks. The basis used in the studies by Sharalfadin et al. [14] was assembled together with the proposal of a new vision for the taxonomy of DDoS attacks, where attacks are divided into two classes: DDoS based on exploitation and DDoS based on reflection, where both exploit a third component in order to use it as a “reflector” for launching the attack.

This database was also generated from the system*B-profile* [17] for simulating human behavior, thus obtaining benign natural traffic data. The base contains

the abstract behavior of 25 users based on HTTP, HTTPS, FTP, SSH and email protocols. Data flows were simulated for two days and captured in PCAP form and the information was extracted by the platform*CICFlowMeter-V3*[15] [16] and exported to CSV format. The attacks carried out are shown in Table 3.2.

Table 3.2: The data streams obtained through the collection system provide 87 attributes for each row of data.

Land	Service	Protocol	Flag	TTL	Class
0	http	TCP	SYN	TTL≥128	Normal
0	http	TCP	SYN	TTL≤128	Attack
1	ftp	UDP		TTL≥128	Attack
0	smtp	ICMP		TTL≥128	Normal
0	http	UDP		TTL≥128	Normal
1	http	TCP	SYN	TTL≥128	Attack
0	http	UDP		TTL≥128	Normal
1	ftp	ICMP		TTL≥128	Normal
1	ftp	ICMP		TTL≥128	Attack
0	ftp	ICMP		TTL≤128	Attack

It is important to note that the types of attacks are broken down according to the protocol or service exploited for the attack. For example, the SNMP-type attack is based on the SNMP application layer protocol. *Simple Network Management Protocol*), used for network management. Another example is the DNS attack. *Domain Name System*) that seeks to exploit the DNS service of servers to launch the attack.

3.3 ENTROPY OF A DATASET

A widely used measure in the detection and analysis of network flows is the entropy measure. Monk*et al.*[8] used the entropy concept to create an algorithm capable of detecting EDoS attacks. Following the same concept, Idhammad*et al.*[1] applied the concept together with machine learning techniques to increase cyber-attack detection performance.

Entropy is defined, for a continuous and memoryless information source, as a measure of the average information content per source symbol [18] (1). In this work, this measure will have as main objective to indicate the amount of information contained in an event from a subset of the attributes of the network data. The event in question is a time window*w*within the training dataset on which the entropy calculation algorithm will be applied.

This step, based on the amount of information contained within a dataset, is applied only to the most relevant attributes for the problem and which, in fact, represent the most sensitive information within a DDoS attack.

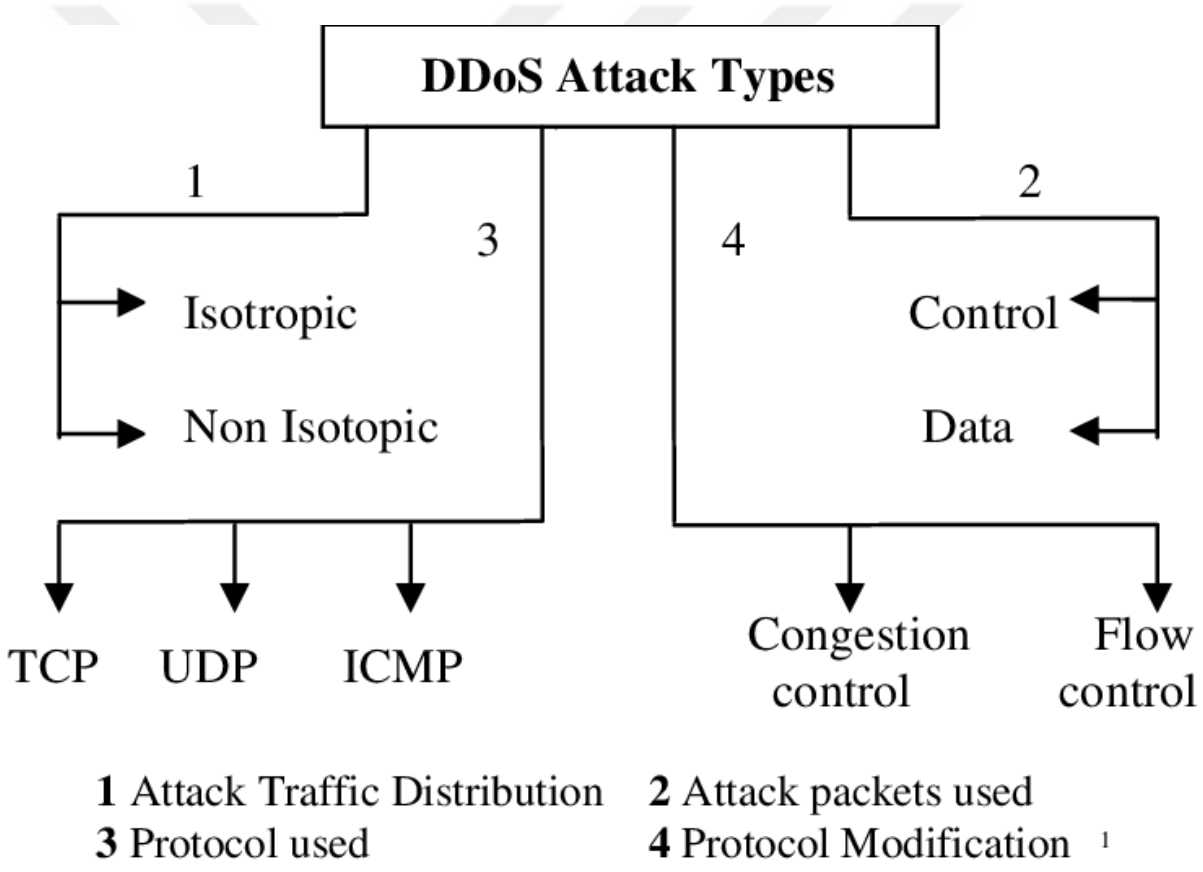


Figure 3.2: Entropy filtering for the DDoS attack

The attributes used were the amount of bytes and packets, sent and received, per data flow, totaling four attributes for the entropy analysis. Importantly, the motivation behind choosing these attributes is due to the intrinsic characteristics of a DDoS attack that aims to overload the victim with service requests. For this reason, the attributes chosen for analysis need to be the ones that

most change at the time of an attack, directly impacting the entropy of the dataset. The implementation of entropy analysis, in conjunction with a lower and upper threshold data selection algorithm aims to detect attack situations.

3.3.1 Algorithm for establishing thresholds

Predictive algorithms based on time series are widely used for various applications [19] Its use is possible both for forecasting future variables linked to time series as well as for the process of establishing maximum and minimum thresholds for these future variables. The two thresholds (*T*uand*T*d), established refer to the upper and lower selection thresholds, respectively, for the selection of data for the later stages of the problem. the constant*k*, depending on the type of probability distribution used to calculate the forecast error, represents the confidence index of the forecast. The variable ? represents the forecast of a future value of this time series and refers to the actual value of that forecast. The next topic of this work will be destined to the development of the necessary basis for the understanding of the algorithm used to predict the future data of the time series used in this work.

3.3.2 Time series forecast using the ARIMA algorithm

One possible solution for predicting future variables within a time series is using the integrated autoregressive moving average (ARIMA) model. *autoregressive integrated moving average*) proposed by Box and Jenkins [10]. This solution model for predicting future data was proposed for data whose nature presents non-stationary characteristics, and cannot be expressed through AR processes.*autoregressive*), MA (from English*moving average*) and ARMA (*autoregressive moving average)* good understanding of the ARIMA process, it is necessary to understand the other two processes that, when integrated, shape it. Following the order proposed by the acronym of the complete process, the first to be explained will be the AR process, commonly applied

Thus, can be represented as a sum of the stationary process b y *d* times, giving rise to the name *integrated*, giving meaning to integration.

From this process, it is possible to make future predictions from the variables *because e* and *d*.

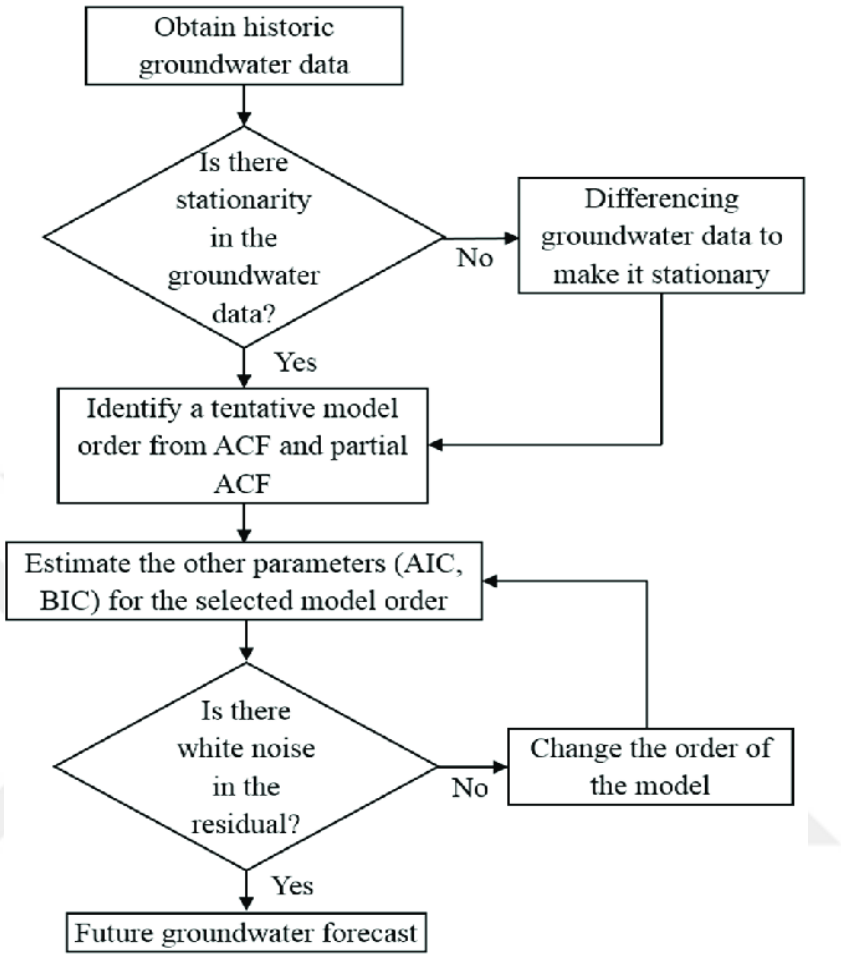


Figure 3.3: Diagram of the ARMA algorithm

3.4 PRINCIPAL COMPONENT ANALYSIS

A topic of extreme relevance within pattern recognition refers to the analysis of data attributes. From this study it is possible to take some situations as ideal for the correct analysis of a pattern. Among these situations, one that stands out is the use of data that do not have redundant information. That is, taking*x*as the input data of an attribute transformation system, having*and*as the output of this same system, it can be said that optimally unrelated data are data that respect the symmetry of*Rx*guarantees mutual orthonormality between the eigenvectors associated with it, the eigenvectors being the columns of the matrix*THE*. Thus, the relationship established by guarantees that*Rand*be a diagonal matrix with the diagonal elements being the associated eigenvalues. Assuming, further, that *Rx*is a definitely positive matrix, that is, if the final equality is different from a null matrix, we have the basic definition of the Karhunen-Loève transform

(KL), generating mutually unrelated attributes. Thus, from the choice of eigenvectors associated with the highest eigenvalues is minimized and represented by the sum of the lowest eigenvalues. This is why the KL transform is also known as Principal Component Analysis. *Main Component Analysis*or PCA). In general, the PCA method uses linear algebra principles in order to remove redundancies present in data sets, in order to improve the performance of certain machine learning tasks.

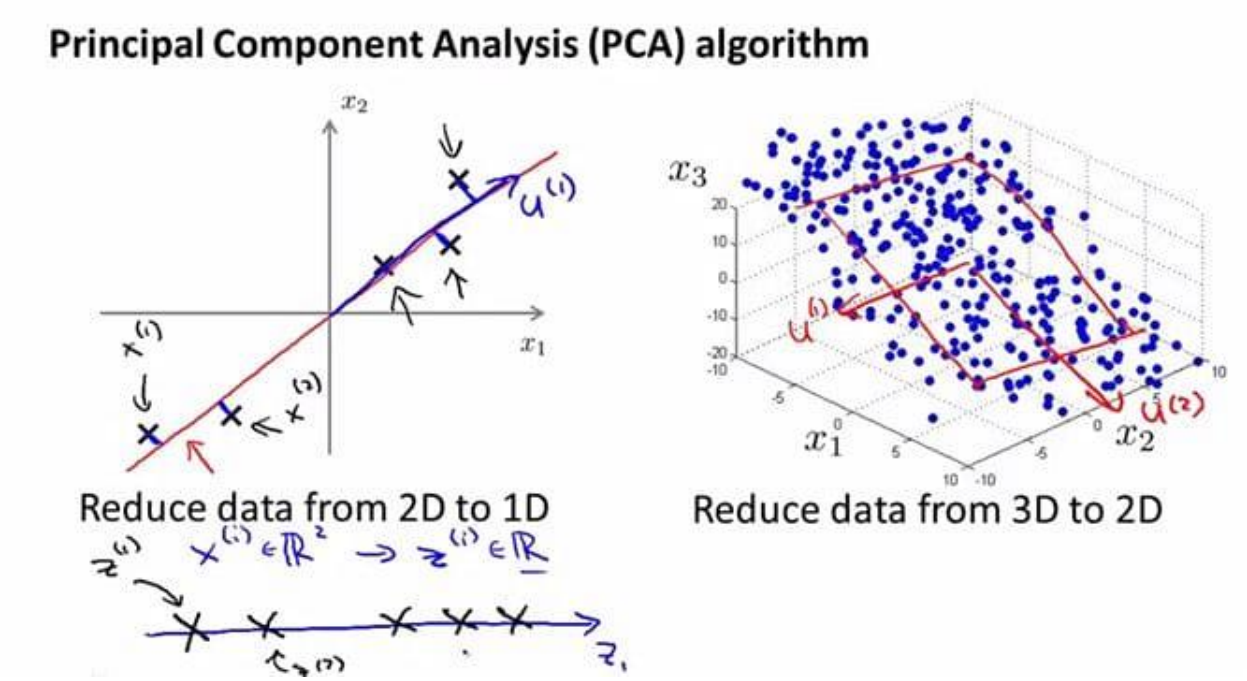


Figure 3.4: Data reduction using PCA

3.5 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) has the graph model as its main feature. In this model, the nodes are the artificial neurons and the edges are the connections between the input and output of the neurons [23]. Figure 3.2 represents the neuron model proposed by this classifier. it can be said that = [1, 2, 3, . . .,] is the vector that represents the feature vector,*E1*is the system standard input, = [0, 1, 2, 3, . . . ,]are the weights assigned to each entry,*f*is the activation function and, finally, [ ]is the system output [24].

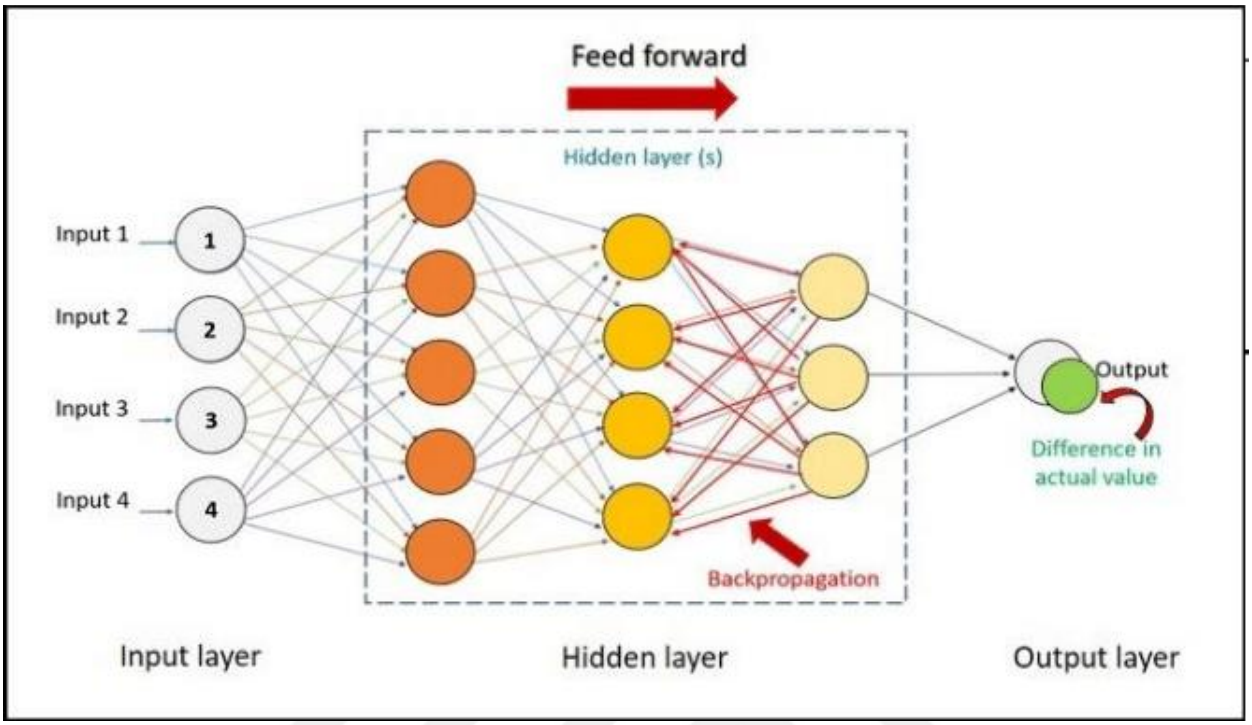


Figure 3. 5: represents only one neuron, with activation function *f*

However, an ANN can have complex schemas with multiple neurons and multiple layers as shown in Figure 3.5

3.5.1 Parameterization of an Artificial Neural Network

Several parameters can be modeled for the evaluation of machine learning algorithms. Each type of algorithm has a class of parameters that can be varied to improve the system.

For the development of the ANN algorithm, four different parameters were varied, namely:

- Network Architecture

Several architectures can be proposed with several neurons and several layers, making the possibilities for this parameter infinite. However, in this work, we will evaluate 3 different architectures, chosen arbitrarily, for the intermediate layer of neurons, namely: (100, 100, 10), (100,50,10), (50,10). These values refer to the hidden layers of the artificial neural network, that is, the layers between the input layer and the output layer. each group of values refers to a different architecture and each value within the value group refers to the number of neurons in the layer.

- *solver*

This parameter is responsible for choosing the algorithm for optimizing the weights of each input. For this work, only one algorithm for this purpose was considered: the stochastic gradient-based optimizer, also known as adam.

- Activation Function

This parameter refers to the function used in the neurons of the hidden layers. This function is responsible for setting the neuron's output value. The function applied in the development of the work was the*rectfied linear unit function*.

- Alpha

This parameter fulfills the role of regulating the established decision boundary. High values for this parameter result in a less flexible decision frontier fighting a high variance value. In contrast, low values for the parameter provide a more flexible decision frontier and more sensitive to the presence of data. Figure 4 shows the results of the variation of this parameter.

It is important to confirm the function of Figure 3.6 , as some versions of it were repeated throughout the chapter. This figure shows the classification task applied to elements of two different classes (blue and red). The highlighted area represents to which class a particular element was classified. That is, if any element of the blue class is in an area demarcated as red, that element was wrongly classified. The difference in tones, most prominent in the alpha 10.0 example, indicates the probability of the element being of a certain class. That is, the more highlighted the color, the greater the possibility that an element in that area is of the class represented by the color.

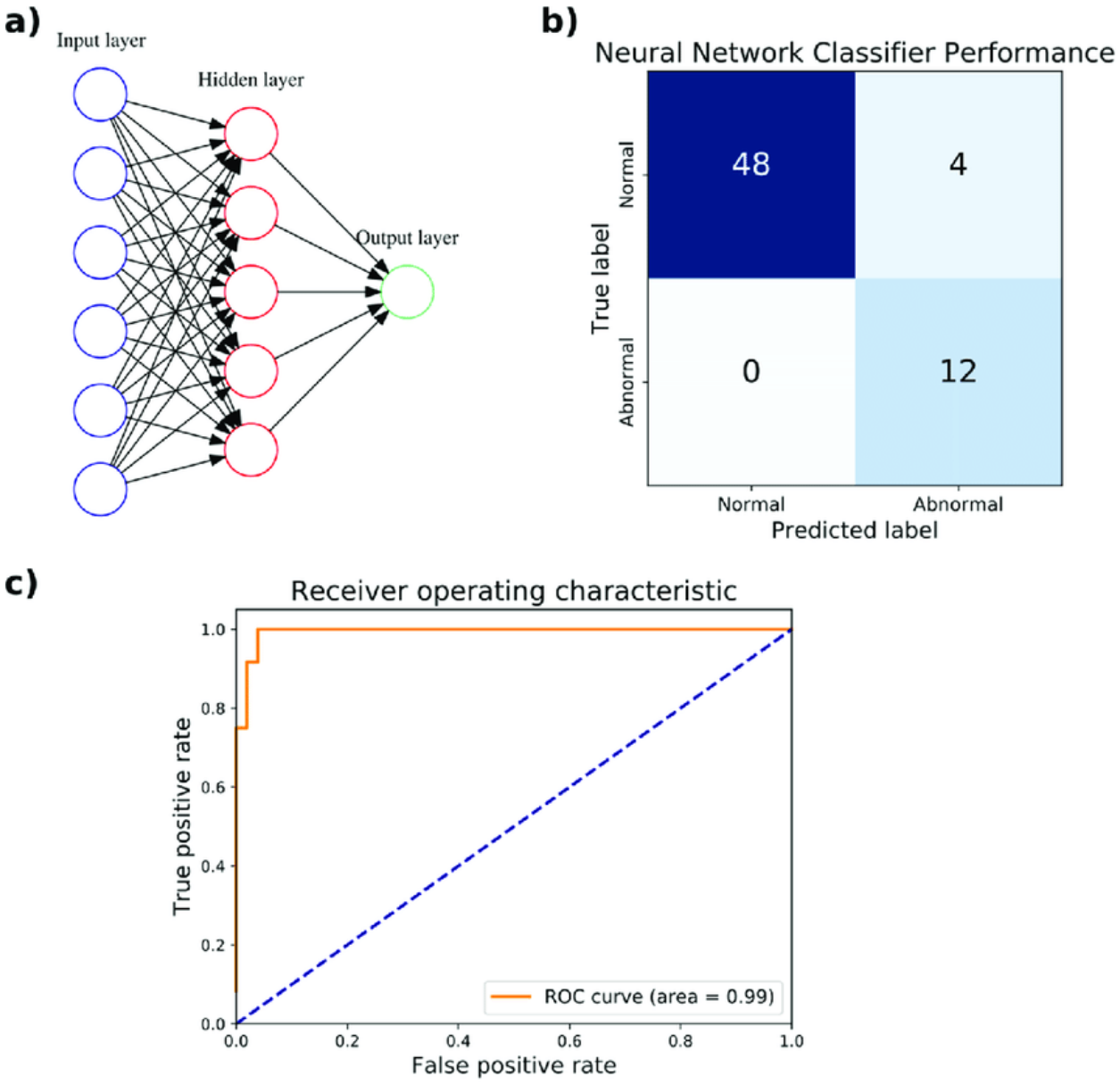


Figure 3. 6: ANN classification

3.6 SUPPORT VECTOR MACHINE

The support vector machine (SVM) a machine learning technique based on the geometric representation of attributes. This approach aims to establish a function, based on the distances between the points of this geometric representation, capable of discriminating different classes.

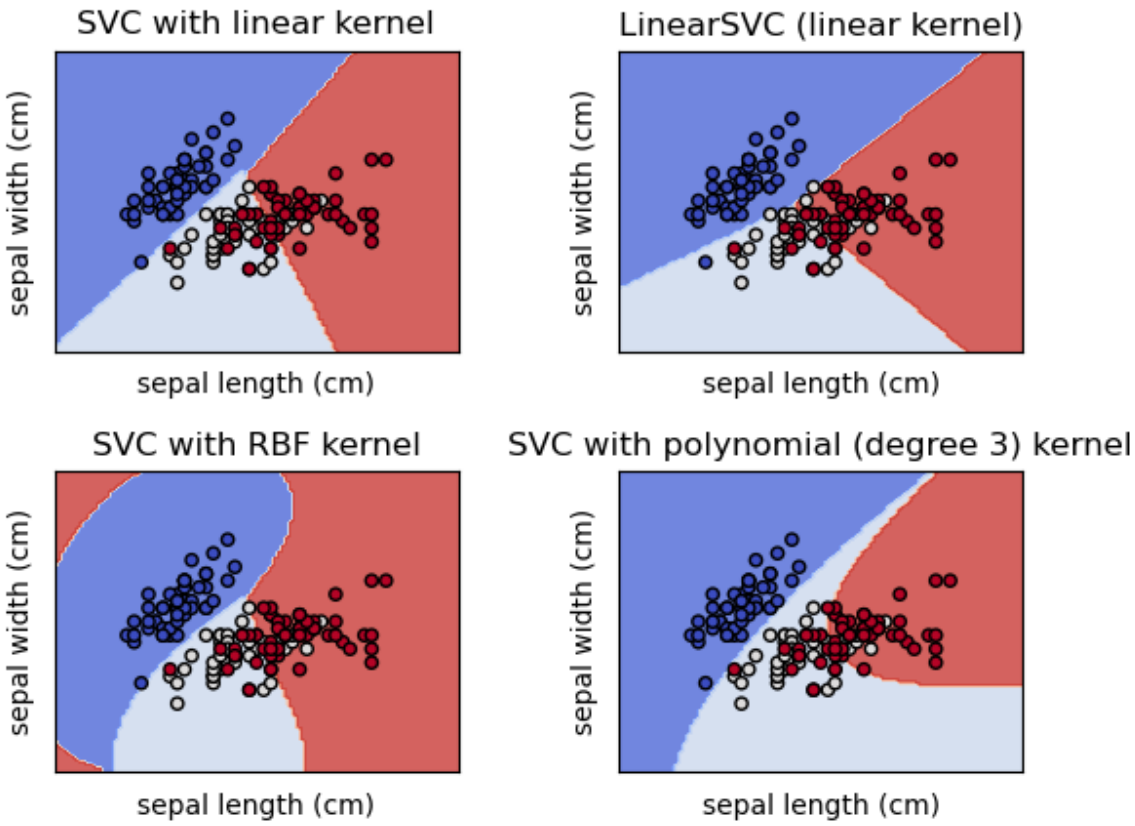


Figure 3.7: shows a simple example of classifying geometric shapes into two distinct classes

In this example, the object of study has two attributes, represented by the horizontal and vertical axes (1 two). From the arrangement of the geometric shapes, the red curve is defined, which is a decision boundary for the system to define to which class the object belongs. This curve is defined based on the Euclidean distance between the shapes in the graphical representation.

3.6.1. Parameterization of a Support Vector Machine

Analogously to ANN, the support vector machine also has important parameters that define its functioning. Each combination of parameters generates a unique result, making the possibility of combinations infinite.

For the development of this work, the parameters considered for the modeling were:

- kernel

This parameter refers to the type of function that will be used to format the decision boundary, in red in Figure 5. Figure 6 illustrates three examples with different kernel functions for solving a problem with three classes.

- Parameter C

The C parameter determines the system's sensitivity to variations. That is, the higher the value of this parameter, the greater the sensitivity of the decision frontier. The advantage of having a less sensitive and more error-prone system is to avoid a possible conditioning of the system to a fixed data model. Figure 7 shows an example of sorting with different values for this parameter.

- Grade

This parameter promotes changes only in the polynomial kernel, not changing the result of the others. This constant refers to the degree of the polynomial used as the decision frontier. Figure 8 shows the difference between different degrees for the polynomial kernel.

- gamma

Among the kernel types used in this work, only the RBF undergoes changes with the change of the gamma parameter. This parameter influences how much a single training sample affects the model. The gamma parameter can be seen as the inverse of the radius of influence of a single sample. That is, the greater this variable, the smaller the radius of influence of a sample and the smaller, the greater the influence of this sample on the construction of the model. Figure 9 shows how this parameter influences the training model.

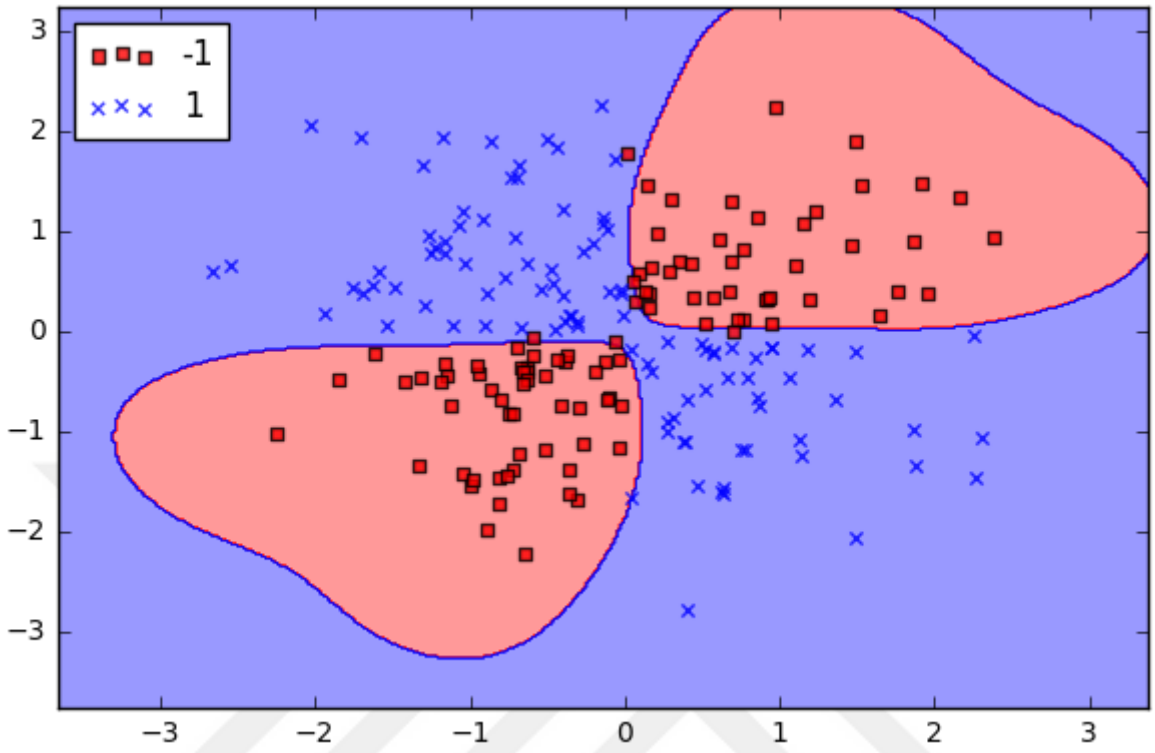


Figure 3.8: RBF kernel with different gamma values

- Class weight

In problems whose data are unbalanced, that is, there are more samples of one class than another, it is common for this parameter to be used to assign a weight to a particular class. This method aims to correct the lack of balance between the classes.

3.7 INTERNET OF THINGS

In terms of data security, the Internet of Things (IoT) presents numerous issues due to the unique characteristics stated in [6]. The fact that several devices communicate with one another poses a challenge since, in certain protocols, the activities of one device are directly characterized by the actions of other devices. It is challenging to identify solutions to security vulnerabilities that affect all of the numerous IoT communication protocols due to the proliferation of IoT protocols. Due to shortcomings in the device's resources, such as memory and battery life, the app's functionality is limited. The large number of Internet of Things (IoT) devices currently in use is described by the word myriad. Due to their location in physically inaccessible or difficult-to-reach regions, it is fairly uncommon for devices to go for extended periods of time without requiring any type of

maintenance. Personal information-storing portable electronic devices, such as smartphones and tablets, contain numerous security vulnerabilities. Causes associated with mobile IoT solutions, in which devices connect unavoidably with a number of other nameless pieces of hardware; mobile IoT-related causes. Internet of Things devices are "ubiquitous," which raises privacy and security concerns. As a result, it is abundantly clear that the Internet of Things contains numerous security flaws that need to be addressed. In this paper, the usage of an Intrusion Detection System, or IDS, is supported as a technique for resolving the Diversity and Myriad issues in regard to information security due to the fact that IDSs are able to detect intrusions before they occur. [6].

3.7.2 Intrusion Detection System

Intrusion detection systems are developed as a security measure to mitigate attacks on certain networks, through the detection and execution of countermeasures. Although these systems are already consolidated in traditional networks, there are still many challenges to be solved in order to fully function in IoT devices [7]. Traditionally, these systems can be classified based on some guidelines, such as: detection method and IDS positioning strategy [7]. The detection method is divided into 4 categories, with respect to which the IDS uses to differentiate the data analyzed by it.

1. Anomaly: information regarding network operation, hardware, such as CPU usage, and software, such as system calls, are previously captured. The captured information is processed to define the thresholds of normal behavior for the system in question. With the threshold defined, it is possible to determine whether new behaviors are anomalous or not. This type of detection is not free of false positives, so techniques for handling false positives should be suggested.

2. Signature: in this type of treatment, it is necessary to create behavioral profiles for each possible threat. Based on the profiles created, the IDS will evaluate the current occurrences, comparing them with the information that was given to it at first. The problem with this type of approach is that it is unable to respond to new threats, given that a signature for it has not yet been created. Note that signatures are details about system calls, or some specificity about the network, for example.

3. Specification: it has similarity with the one based on anomalies, however the creation of the profile of normal behavior is not performed automatically from the analysis of the system

operation. In this case, it is an expert agent that will define the accepted behavior for a given network and devices. Therefore, the quality of the IDS is at the mercy of the capacity of the specialist involved, if he has not defined good rules, the system can emit many false positives for threats, for example.

4. Hybrid: it is when two types of threat detections are combined, seeking the best functioning of each one of them. For example, combining anomaly detection with signature. While it is an attractive strategy, developing this type of solution can be very complicated. The positioning strategy is what defines where the data capture and analysis will take place. There are 3 types of positioning: centralized, which is when the IDS is in a location in the system where it has access to all the information to be analyzed; the distributed one, when the IDS is arranged in some points of the network, and the hybrid, which is a more elaborate approach, but it is the one that can bring better results, considering that it overcomes the adversities encountered by the other two techniques - in this case, the IDS can detect threats that circulate internally in the network, and threats that are directed to the network by an external agent. In the context of IoT it is important that the IDS can work in parallel with the standard activities of the device [8], so as not to affect their performance, which most of the time have resource limitations, as mentioned in the IoT subsection. To this end, there are also implemented IDSs that use machine learning as an important part of the classification of behaviors, as in [5].

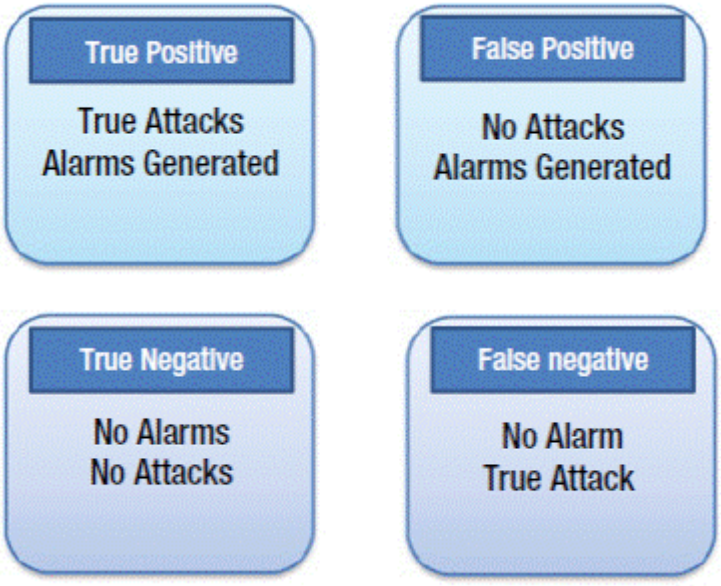


Figure 3.9: Types of evaluation criteria for IDS

3.7.3 Evaluation of IDS

Evaluation of the intrusion detection system's efficacy based on the following three criteria (16):

When an intrusion detection system wrongly identifies a normal behavior as malicious, this is an example of accurate detection. A false positive would be an example of this specific criterion.

• Attaining One's Objectives (performance) The event processing rate of an intrusion detection system can be used to determine its effectiveness. If this rate is insufficient, real-time detection will be difficult.

We explore the incompleteness of an intrusion detection system in the event that it fails to identify an attack. The fact that no one can know everything there is to know about the attacks is the most challenging part of this criterion. Due to the fact that this criterion is the exact opposite of the actual negative, it is appropriate to apply it in this circumstance.

For example, the intrusion detection system must be able to withstand distributed denial-of-service attacks (DDoS). Given the widespread usage of vulnerable hardware and software in numerous types of intrusion detection systems, this is a crucial consideration.

The results of the intrusion detection system must be executed and transmitted as quickly as is practically possible so that the security officer has sufficient time to respond before significant damage has occurred. venue. The time required for event propagation and reaction is in addition to the time required for event processing, hence this exceeds a simple performance estimate.

3.7.4 The different types of IDS

• An IDS depending on the host By scanning numerous domains, a host-based intrusion detection system is able to identify any malicious or abusive network activity (breaches of the network). 'outside). The host-based intrusion detection system (IDS) examines various log files by comparing them to an internal database containing signatures of known malware attacks. This evaluation is conducted on all log files. Unix and Linux-based host intrusion detection systems make considerable use of the syslog command. This command, in conjunction with an event's severity, can be used to classify the event (eg, minor printer messages versus major kernel warnings). The syslog command is accessible after installing the sysklogd package, which is bundled with Red Hat Enterprise Linux. You will be able to monitor your system's logs and kernel messages if you install this package. Administrators are responsible for examining the data collected by host-based IDSs, which filter logs (which, in the case of kernel or network event logs, may be heavily commented out), process it, and then re-mark messages with their own severity rating system. Additionally, host-based intrusion detection systems can evaluate the integrity of critical files and executables. A message file analysis application such as md5sum or sha1sum, both of which employ 128-bit algorithms, is used to determine the checksum of each file in a database containing sensitive files (as well as files added by the administrator). This is performed whenever the database is examined).

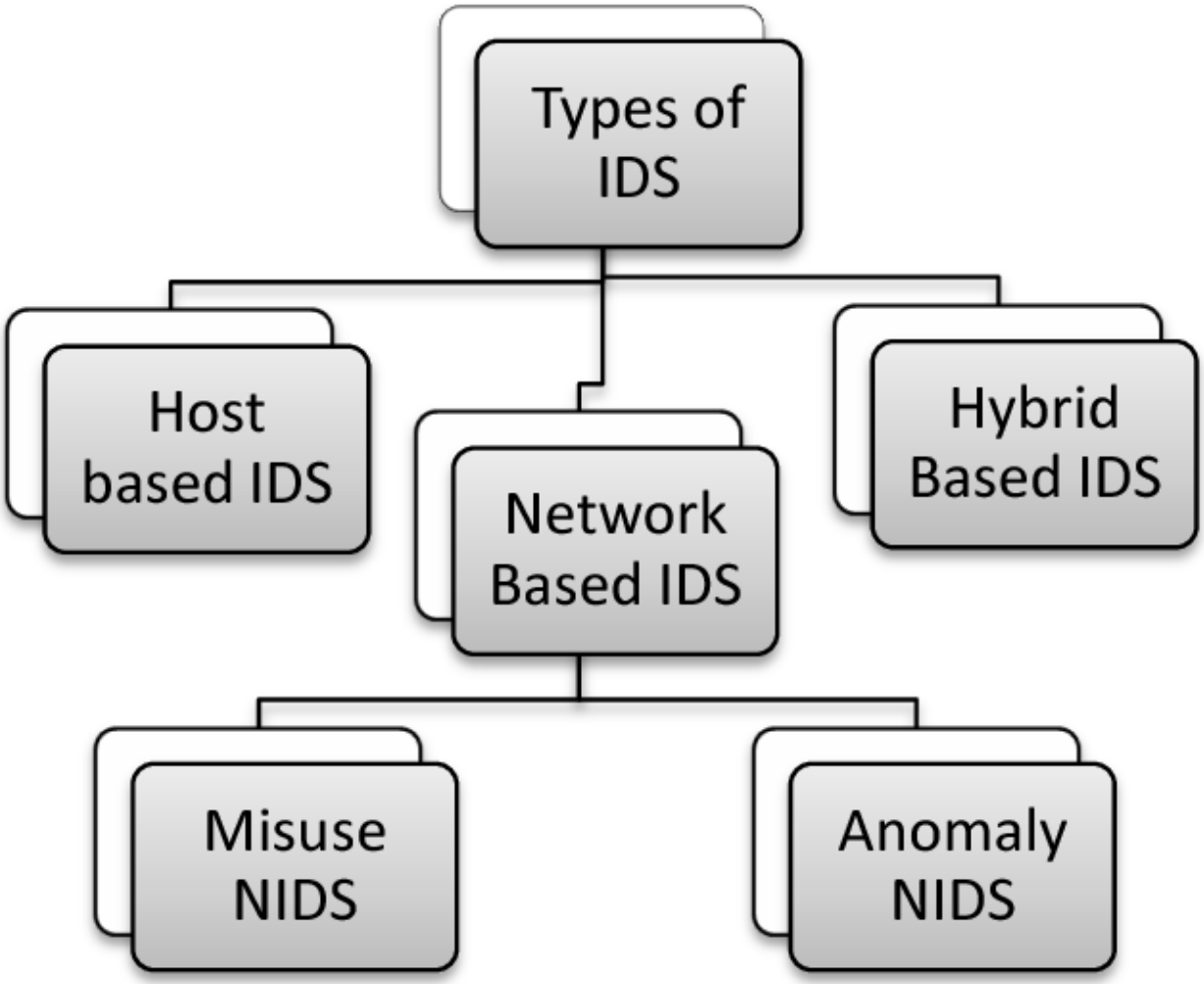


Figure 3.10: types of intrusion detection system

After the totals have been recorded to a plain text file, the host-based intrusion detection system will do periodic checks to confirm that the file checksums match the values written to the file. The IDS will alert the administrator through email or pager if any of the totals are erroneous. The capacity of a host-based intrusion detection system to carefully monitor local user behavior is one of its primary advantages. Determine whether an attack was successful or not. • Capability to

identify certain types of attacks, as a result of the IDS's reliance on operating system audit logs; (e.g. Trojan horse).

A host-based intrusion detection system's sensitivity to denial-of-service attacks is one of its disadvantages. This is because the IDS can exist on the host that is the target of the attacks.

There are deployment and management problems, especially when dealing with a large number of hosts.

These technologies are incapable of recognizing numerous network attacks.

Operated over the Internet Intrusion Detection System (IDS) A network-based system is distinct from a host-based intrusion detection system (also known as an IDS). A network-based intrusion detection system (IDS) scans and processes packets at the host or router level, and suspicious packets are logged in a specified log file with additional information. A network-based intrusion detection system is able to search its own database of known network attack signatures and assign a severity rating to each suspicious packet depending on the packets it discovers. If the severity levels are high enough, members of the security team will receive an email or pager notice so they can explore the anomaly further. Network-based intrusion detection systems are becoming more prevalent as a result of the growth in both the size and volume of internet traffic. In the world of network security, IDSs that are capable of scanning enormous volumes of network data and correctly identifying suspicious signals are welcomed with open arms.

Scanners, sniffers, and other network analysis and detection tools are vital for preventing harmful network activity such as identity theft and denial-of-service assaults. This is due to the fact that TCP/IP protocols are fundamentally susceptible.

Man-in-the-middle attacks • Corruption of the Arp cache • Corruption of the DNS name

There are advantages to employing a network-based IDS.

The IDS network base may control a big number of hosts while requiring only a little amount of resources to install. It has no impact on the performance of the businesses under supervision.

Network-based intrusion detection systems are capable of identifying attacks originating from a wide range of hosts.

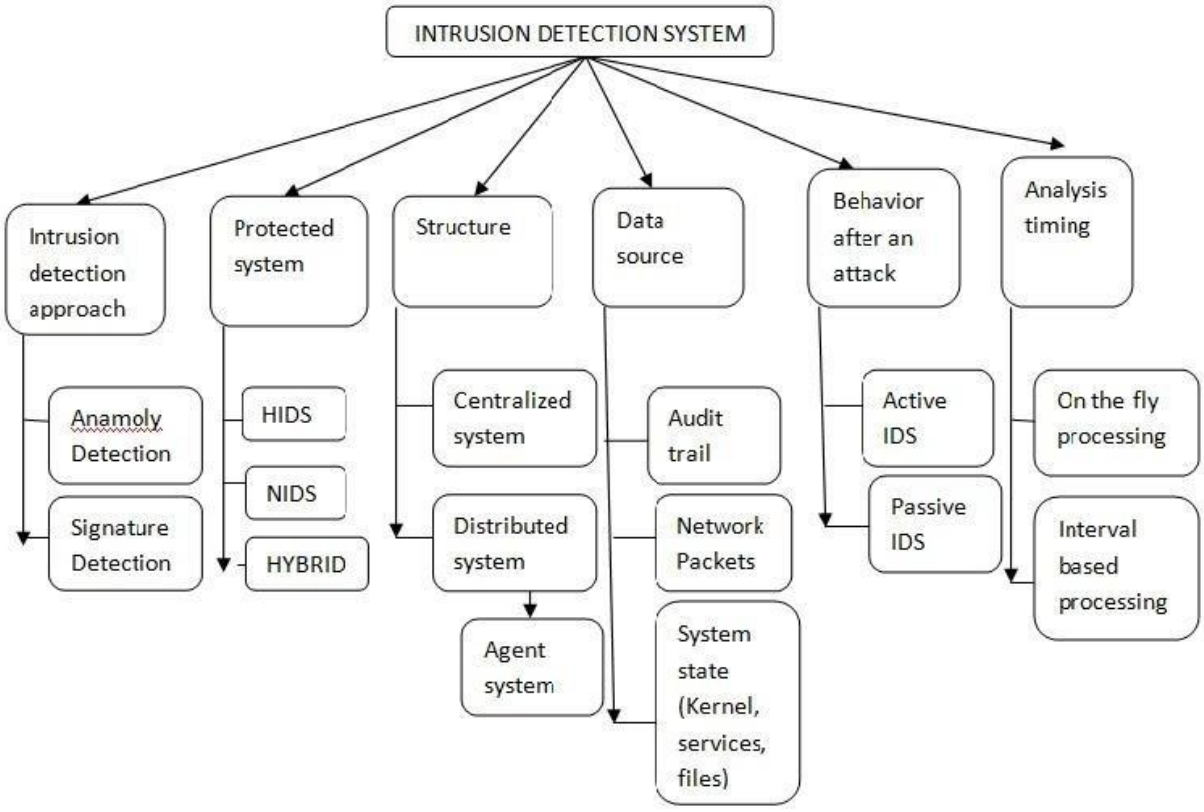


Figure 3.11: types of IDS attack prevented by the system

The network-based intrusion detection system (IDS) provides excellent protection against attacks while evading detection by attackers.

Network-based intrusion detection system disadvantages

In pancake attacks, network-based intrusion detection solutions are ineffective.

Using this type of IDS, it is impossible to verify if an attack was successful.

It is difficult to compare and evaluate different intrusion detection systems since there are so many possible data sources, and the test data is not representative of real-world scenarios.

Historically, the efficacy of an intrusion detection system was measured using two criteria:

An intrusion detection system's (IDS) dependability can be determined by whether or not an intrusion successfully triggers an alarm. The occurrence of false negatives may result from an undetected IDS problem. The optimal level of performance is reached in an intrusion detection system when the false negative rate (the percentage of incursions that are not detected) is as low as possible.

There must be a direct relationship between the alarm and the actual security breach. If an intrusion detection system (IDS) generates a false positive, its utility is lessened. IDS should have the fewest number of false positives feasible. Not only should there be a high percentage of successful detection of intrusions, but there should also be a low number of false alarms. If the security administrator assumes that all alerts are false positives and does not review them, they face the danger of missing out on preventative steps that could have been done had they been pertinent, such as in the case of alerts that are greater than 1 percent false positives. Because they assume that all warnings are false positives, this is the case (.1 percent is a conservative figure). Due to the current state of affairs, the intrusion detection system has become ineffective.

the efficiency with which intrusion detection systems are applied in the environment. The efficiency of intrusion detection systems is evaluated using three criteria proposed by Philip in the sources listed below:

A system's accuracy can be determined by whether or not it wrongly labels a benign behavior as malevolent. This is a measurement of the phenomenon referred to as false positives.

The achievement of the desired result (performance) The processing speed of events determines the effectiveness of a detection system. If this rate is insufficient, real-time detection will be difficult. We refer to the phenomena that occurs when an intrusion detection system fails to recognize an attack as "completeness." This is the most challenging condition to achieve, as it is impossible to have a thorough understanding of the attacks.

This criteria is equivalent to the genuine negative [4]. In addition, Debar included the following two factors in his list:

For example, the intrusion detection system must be able to withstand distributed denial-of-service attacks (DDoS). Given the widespread usage of vulnerable hardware and software in numerous types of intrusion detection systems, this is a crucial consideration.

The analysis must be performed as rapidly as feasible by the intrusion detection system, and the results must be disseminated quickly so that the security officer can respond before major damage is done. venue. As a result, this is more than a simple performance calculation because it takes into account the time required to propagate and respond to this event. [2]

3.8 MACHINE LEARNING

Machine learning is the process by which a computer learns to decide the best effective treatment for a new condition by analyzing the medical records of prior patients. Machine learning enables computers to carry out a broader variety of tasks than was before possible. By modeling the issue using functions defined in a database, the computer will be able to apply these functions to unforeseen conditions. Learning strategies may be supervised, unsupervised, or partially supervised. When applying supervised learning methodologies, it is crucial to remember that a database must already be prepared for computational evaluation. This means that the data in the supplied situation must already be tagged. Its objective is to teach a classifier to classify this collection automatically based on its contents. Using the classifier as the final step in the encoding procedure enables the data to be encoded even if they are not included in the training set [10]. When teaching an algorithm, semi-supervised learning algorithms utilize both labeled and unlabeled database data. Unsupervised learning requires an unlabeled database as input, and the learning approach relies on the data to correctly identify itself as precisely as possible. [12].

4.PROPOSED METHODOLOGY

From the concepts discussed in the previous chapter, there needs to be a methodology behind the execution of the comparative study. This stage of the work determines a process architecture capable of guiding the development of the two approaches that make up the project. In addition, this chapter seeks to guide the best methodology to carry out the comparative study between the approaches, as well as the choice of the best comparison metrics.

4.1 DATA PROCESSING

The first step of the methodology refers to the choice of databases that will be used in the implementation of the study. The second topic of Chapter 4 confirms the importance of using databases that are consistent with the reality of the phenomenon being studied.

However, the choice of data sources is not based only on this, it is important that a study is carried out in the format and arrangement of data within the database.

For this work, it was established as a standard to work with files in the *CSV* for the agility of the analysis through control tools such as the Excel query editor and, also, for the ease of manipulation through the Matlab programming language, used throughout the project.

The first step taken with regard to the architecture of processes, which is common to the two approaches developed in this work, is the reading of the files *CSV* and selection of columns (attributes) that will need to be excluded from the analysis, or transformed in some way

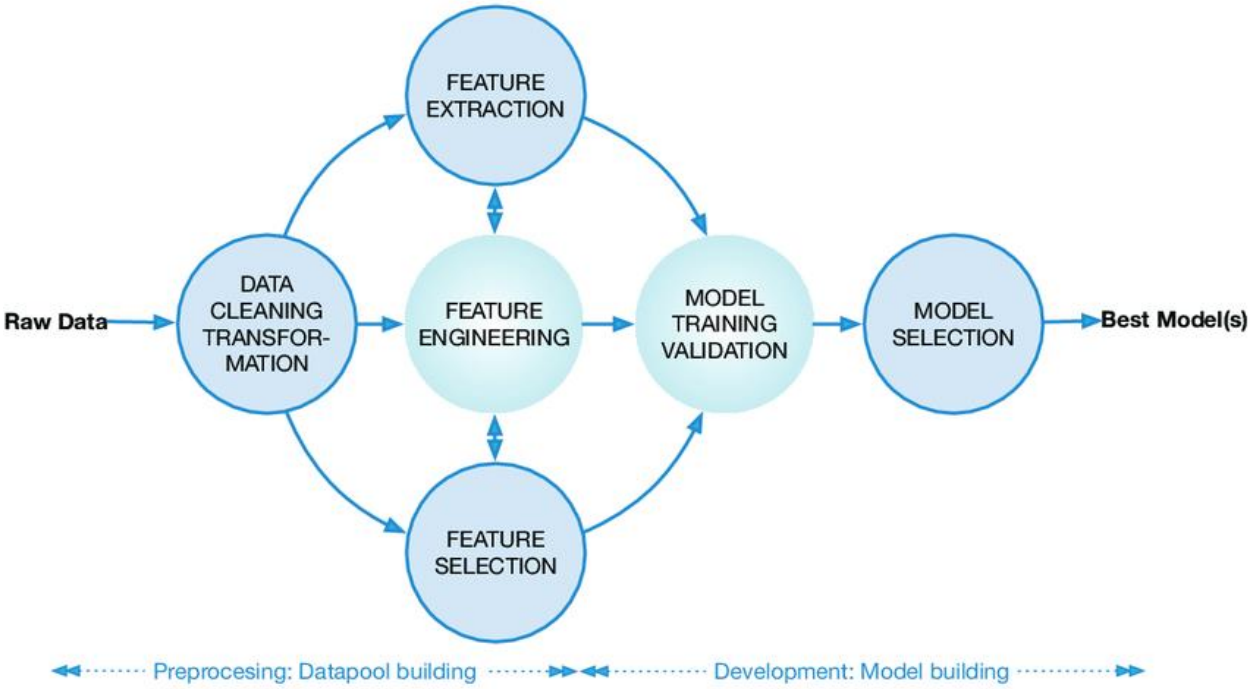


Figure 4.1: Data preprocessing step in the IDS system

Table 4.1 Dataset splitting

Classes	**train**
Benign	8584473
Bot	178520
Brute Force -Web	416771
Brute Force -XSS	416771
DDOS attack-HOIC	**416771**
DDoS attacks-LOIC-UDP	1107
DDoS attacks-LOICHTTP	368509
DoS attacks-GoldenEye	26565
DoS attacks-Hulk	264264
DoS attacks-SlowHTTPTest	3881
DoS attacks-Slowloris	6422
FTP-BruteForce	12589
Infiltration	416771

SQL Injection	416771
SSH-Bruteforce	66014

4.2 APPROACH BASED ON THE ARIMA METHOD

The first step carried out in this approach concerns the implementation of an algorithm for the windowing of data lines according to time, as shown in Figure 4.1. It was established as a standard to use a five-minute window to perform the analyses. Furthermore, the windowing process implemented is not static, that is, it has a slip factor of one minute responsible for standardizing the data for analysis in order to detect the threat as soon as possible. The later steps of the approach based on the ARIMA methodology will focus on calculating the entropy of each data window and, later, the application of the method to perform the entropy prediction of future windows. Then, a thresholding process to define attack situations is performed and applied to the test data.

Figure 4.2: shows the flowchart of the proposed algorithm

In this way, the attack situations will be detected from cases that have entropy values that totally differ from the normal situations.

4.3 SUPERVISED APPROACH

In the supervised approach, machine learning algorithms will be implemented to train a model capable of making predictions about future values. In this approach, the evaluation is performed line by line, that is, without the application of the windowing mentioned in the previous section. The tested algorithms, modeled from the information contained in the data windows, will be the ANN and the SVM. In addition, another algorithm will be developed with the addition of a step prior to training the model, the PCA step, whose objective, previously described, is to reduce the data dimension in order to attribute greater weight to more redundant attributes for the classification of a certain data.

Table 4.2 Confusion matrix preference table

Actual/Predicted	**Test: YES**	**Test: NO**
Train: YES	TP	FN
Train: NO	FP	TN

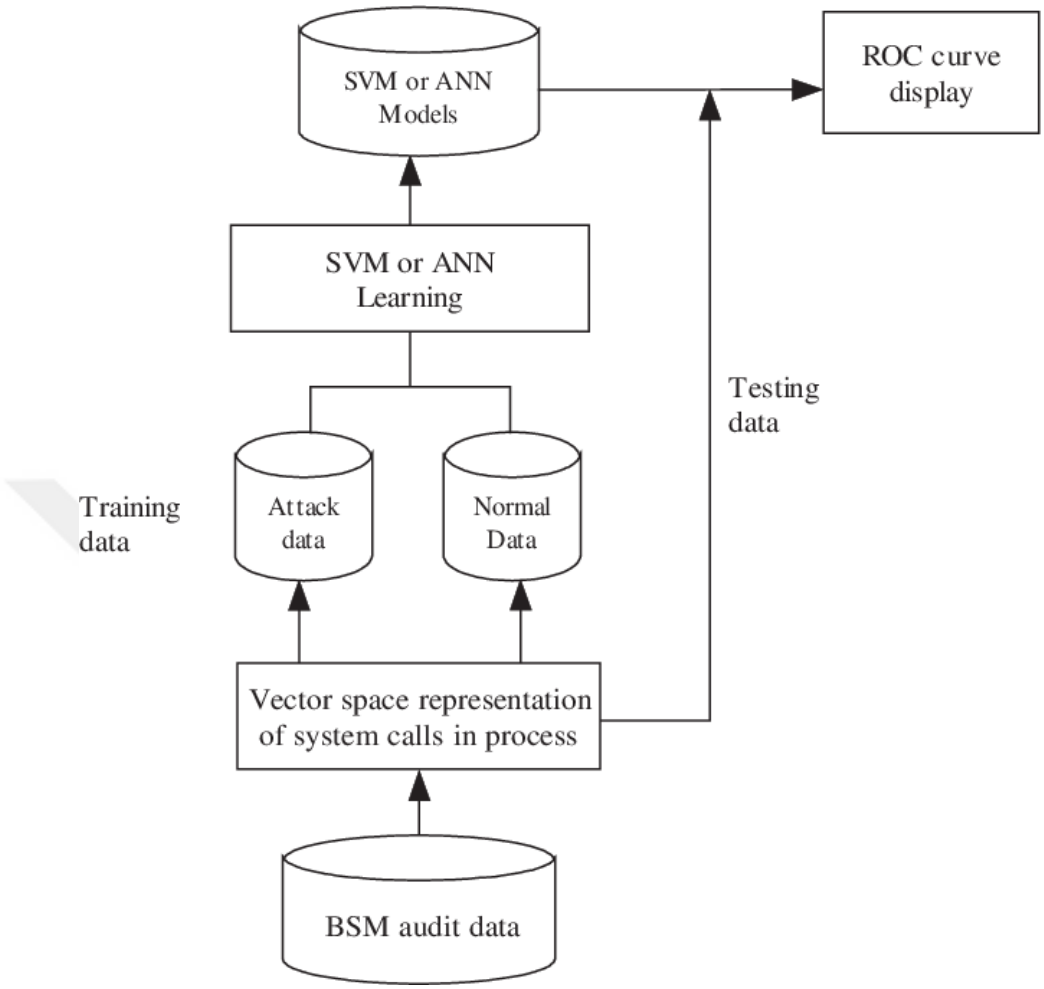


Figure 4.3: shows the data flowchart capable of summarizing the processes implemented in this approach

The data, after the necessary treatment, is divided into two partitions, so that one of them is used in the PCA analysis and in the training of the machine learning algorithms, represented in Figure 4.3 by*DTr*, while the other is used for the evaluation of training models, represented by*DTe*. Another important metric for the study, which is a determining factor for the development of threat detection systems of this nature, is the execution time of the processes. This metric will be used as the third tiebreaker.

4.4 COMPUTATIONAL RESOURCES

The resources that will be used for the development of the proposed algorithm are of great importance. For the visual analysis of the data and assistance in the development of the work, the Excel query editor was used. As a programming language for development, Matlab [32] was

chosen within the IDS function development environment, mainly because of its easy-to-understand syntax and its common use in performing tasks aimed at machine learning.

Along with Matlab, some libraries will be fundamental for the execution of the work in a simpler way, namely:

- csv: library responsible for reading and operating csv files
- ipaddress: library for converting ip address to an integer
- Datetime: library responsible for reading and operating data in date format
- Scipy and Math: for math operations
- Pandas: for data manipulation
- Numpy: for operation with vectors and lists
- Matplotlib: for creating images and graphs
- StatsModels: for ARIMA model implementation
- Scikit Learning: for the implementation of machine learning algorithms
- Excel: Using the Query Editor for Initial Analysis

From these software resources, it will be possible to carry out the implementation of the study proposed by this work.

5.RESULTS AND DISCUSSIONS

This chapter will elucidate the structure of the code developed for each of the proposed methods. In addition, the chapter will bring the results for each of the methods according to their particular variables, as well as compare the results for each methodology. In addition, not all test results were exposed in this work due to the large number of tests performed. As a result, a directory [33] was made available within the GitHub platform for storing the scripts used and the results of the work.

5.1 ARIMA METHOD

The ARIMA method was implemented from subscripts improve code organization and documentation. Figure 5.1 shows a diagram representing the architecture of the code and the flow of data during its execution.

From this, it was possible to assemble the graphs in Figures 5.1, 5.2 and 5.3 to represent the selection task through the threshold obtained by the ARIMA method. The graphs generated from this analysis show the entropy of the data window (y-axis) in relation to time (x-axis).

It is important to emphasize that, due to the similarity between most of the graphs and the number of graphs obtained, only the three cases that have the most different behavior were chosen to compose the work. Furthermore, the time to build each model was omitted from the work due to the low processing time, all models took less than 1 second to be processed. In addition to the visual aspect, the results were also the basis for choosing the graphics used in the work, with one of each model proposed in Table 5.1 being chosen.

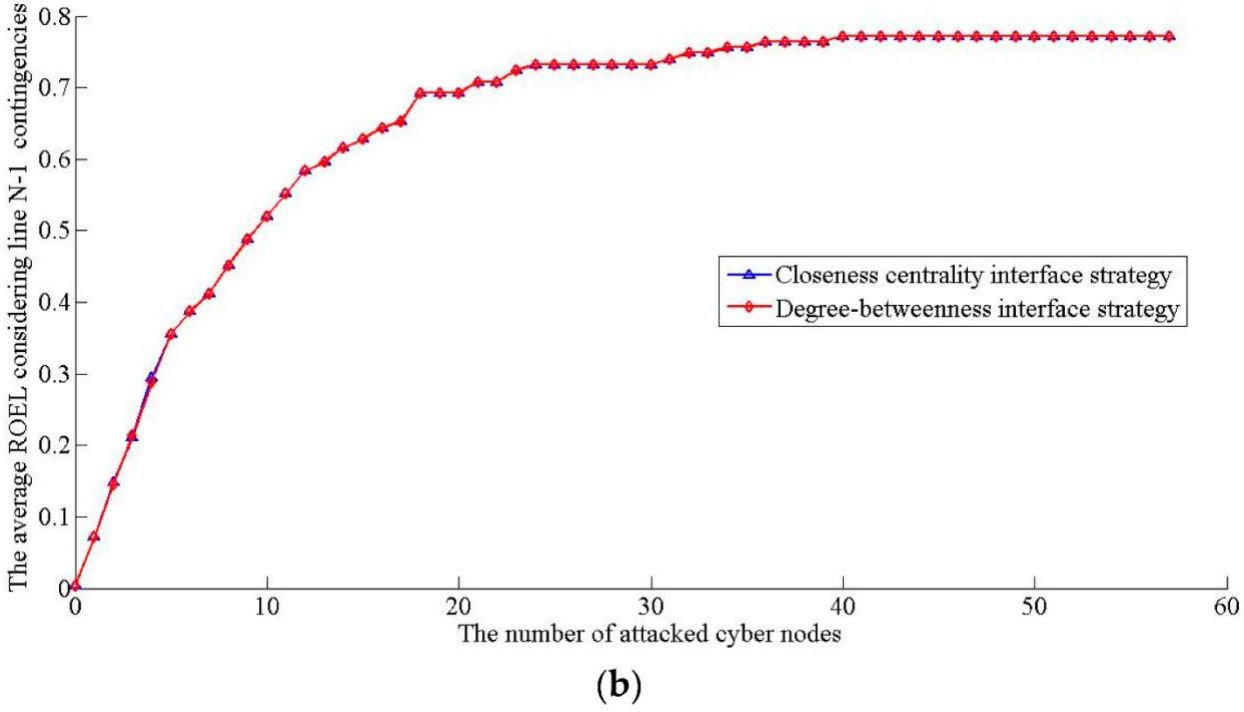
Table 5.1 Evaluation metrics for the ARIMA model

Evaluation	**Percentage**
Recall	0.98
Fscore	0.97
Accuracy	0.97
Precision	0.98

Another important aspect that relates the theory of equation (1) and DDoS attacks in practice is the behavior of the blue curve, represented in the three images. Low entropy values indicate a low amount of information in a data set, showing that the values of this set are very close to each other. This is a characteristic of these types of threats, which send several requests with values close to*bytes*and packages.

The three figures have the same curve in blue, as it represents the average entropy values of the test windows over time. On the other hand, the orange curve, representing the decision threshold, changes according to the parameters (p,q,d) of the Arima method.

The windows (represented by dots) below the orange curve are considered by the system as attacks and the windows above are considered as normal cases.



(b)

Figure 5.1: shows the threshold just after the tenth window. That is, from this window the system considers it as an attack, which ends after the thirty-seventh

Table 5.2 Evaluation metrics for the threshold model

Evaluation	**Percentage**

Recall	0.96
Fscore	0.93
Accuracy	0.97
Precision	0.90

Figure 5.2: Window Entropy Graph and threshold (3,3,1) as a function of time On the other hand

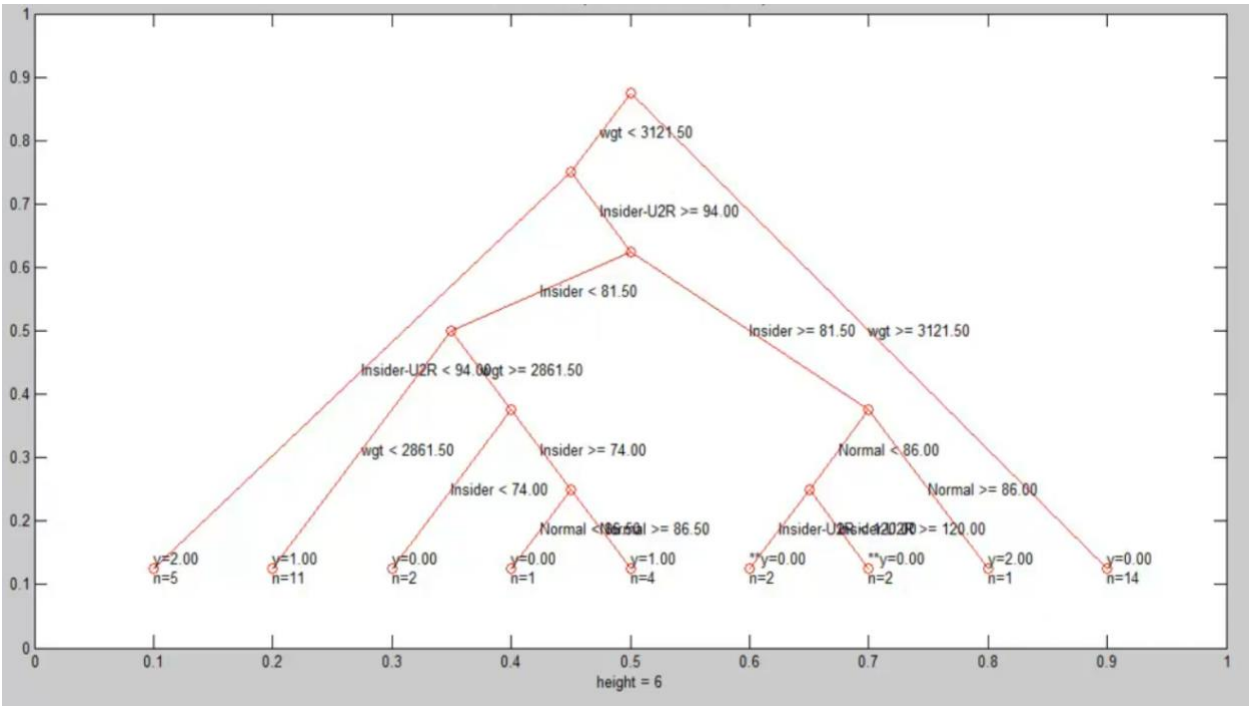


Figure 5.3: shows that the system considered attack situations, starting from the twelfth window, which extended to the thirty-sixth window

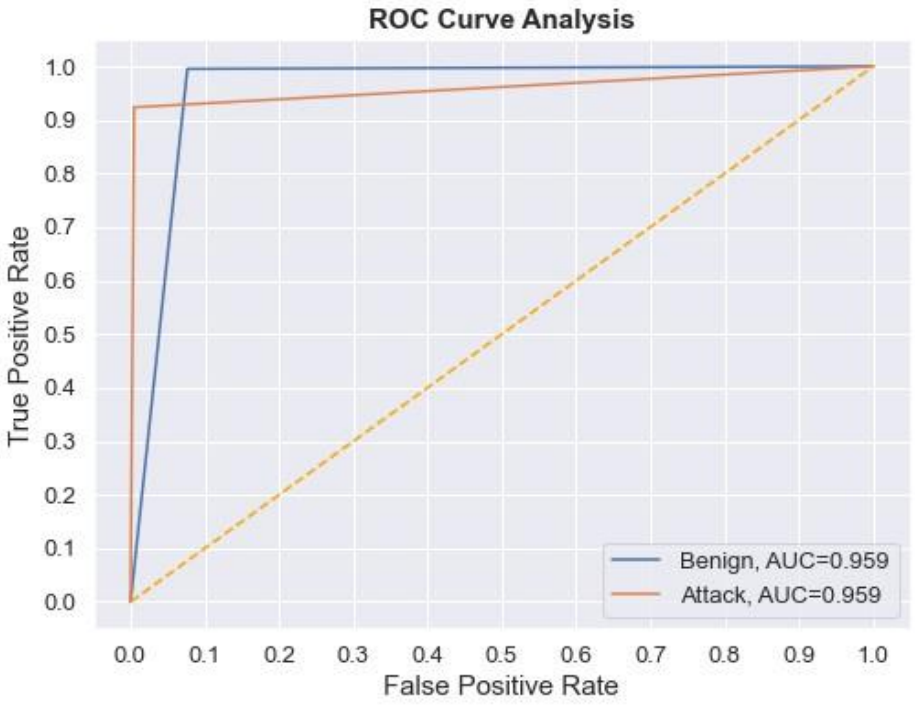


Figure 5.4: Window Entropy Graph and threshold (2,3,3) as a function of time

Finally, the second graph shows the threshold establishing attack situations from the twelfth window that would extend completely to the thirty-sixth, but due to the behavior of the threshold, it considers the thirty-sixth as an attack, but the thirty-fifth as a normal case.

5.2 SUPERVISED LEARNING METHOD

Similarly, to the ARIMA method, the supervised learning method was implemented from modules to improve code organization. Generally speaking, the two classifier models were applied to a subset of 10000 data taken from the main dataset and the remaining data were used for testing. Five mapping vectors, defined randomly, were used to carry out tests using different training sets. The flowchart in Figure 5.5:

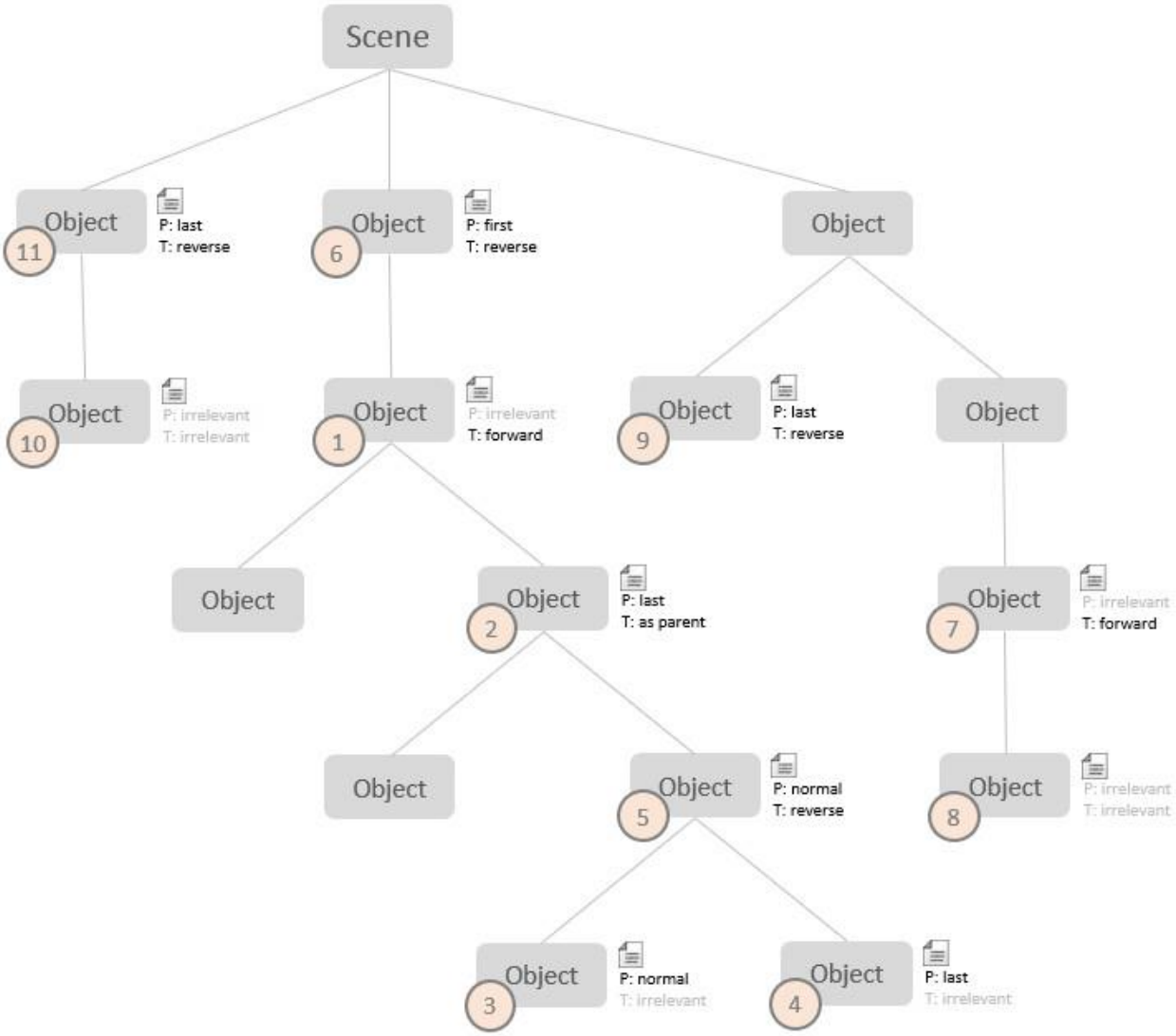


Figure 5.5: shows the execution of the scripts. The ML block refers to the different learning methods applied

Due to the large number of tests performed, the models were divided into the following categories: SVM, ANN, SVM+PCA and ANN+PCA. Based on this division and the selection criteria detailed in item 3.4 of this work, the following models were selected: In order to understand the impact of applying the PCA, the metrics of the two models used in Table 2 in ANN+PCA and SVM+PCA without the application of PCA. Thus, comparing the results of Tables 1 and 2, the ANN+PCA method with dimensionality reduction for 49 components, architecture (100,50,10) and alpha equal to 0.001 and was the one that showed the highest performance in the proposed task, using the criteria stipulated in item 3.4. In

addition, it is possible to perceive the effectiveness of the application of the PCA method for this type of application. If compared to the SVM+PCA method, the metrics are considerably closer, being surpassed by the ANN+PCA method by only one hundredth in the accuracy criterion. However, the processing time of the SVM+PCA method is about 40% shorter compared to ANN+PCA.

Table 5.3 Performance table for the ANN-PCA model

Evaluation	**Percentage**
Recall	0.98
Fscore	0.98
Accuracy	0.99
Precision	0.97

6.COCNLUSIONS AND FUTURE WORK

6.1 CONCLUSION

This work presented different pattern recognition methodologies applied to the task of detecting DDoS attacks. Initially, an exploratory analysis of the preliminary data was carried out in order to choose the database to be used in the work, the selected one being the UNB CIC-IDS 2021, detailed in section 3.2.1 of this work. After the exploratory analysis, carried out with the Excel query editor, the database underwent a selection and normalization process with the objective of transforming it into a base capable of being used in the following processes.

The ARIMA methodology, using entropy in time windows as a basis for establishing thresholds for normal and abnormal cases, demonstrated a maximum f-score of 67.04% with an overall accuracy of 62.02% and an average processing time of less than 1 second. On the other hand, classical methods of*machine learning*proved to be extremely efficient when combined with a good choice of parameters. Among the hundreds of tests performed combining the pre-established parameters, the most successful combination was the ANN+PCA, obtaining an f-score of 99.99% and an overall accuracy of 99.99%. However, it is important to note that the SVM+PCA model obtained significantly important metrics, as it equaled most metrics, losing only by one hundredth in the overall accuracy measure of the ANN+PCA model, but obtained a shorter training time than the model. Another important consideration to be made is the positive impact that the analysis of main components provided, being essential for the improvement of the metrics of most of the models tested.

6.2 FUTURE WORK

For future work, the application of the analyzed techniques in new databases is suggested. Another possibility is to use data selectors to improve training efficiency using fewer samples. Real-time testing would allow a better evaluation of the performance of the methods. Another suggestion is the application of other classification methods. The use of decision trees, for example, could facilitate the interpretation of the decision process, allowing the identification of the most important attributes in this process. More advanced techniques such as convolutional neural networks could also be used.

REFERENCES

[1] D. Oh, D. Kim, W. W. Ro, A malicious pattern detection engine for embedded security systems in the Internet of Things, *Sensors* 14 (12) 24188–24211. 2. 2014.

[2] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, M.-C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN, in: Y.-M. Huang, H.-C. Chao, D.-J. Deng, J. J. J. H. Park (Eds.), Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Vol. 260 of Lecture Notes *in Electrical Engineering, Springer Netherlands*, pp. 1205–1213. 3. 2014.

[3] E. Cho, J. Kim, C. Hong, Attack model and detection scheme for botnet on 6LoWPAN, in: C. Hong, T. Tonouchi, Y. Ma, C.-S. Chao (Eds.), Management Enabling the Future Internet for Changing Business and New Computing Services, Vol. 5787 of Lecture Notes in *Computer Science, Springer Berlin Heidelberg*, pp. 515–518. 4. 2009.

[4] P. Kasinathan, C. Pastrone, M. Spirito, M. Vinkovits, Denial-of-service detection in 6LoWPAN based Internet of Things, in: Wireless and Mobile Computing, Networking and Communications (WiMob), *IEEE 9th International Conference on*, , pp. 600–607. 5.2013.

[5] J. Amaral, L. Oliveira, J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, *in: Communications (ICC), IEEE International Conference on*, pp. 1796–1801. 6. 2014.

[6] C. Liu, J. Yang, Y. Zhang, R. Chen, J. Zeng, Research on immunity-based intrusion detection technology for the Internet of Things, *in: Natural Computation (ICNC), Seventh International Conference on*, Vol. 1, pp. 212–216. 7. N. K.2011.

[7] Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, J. Isoaho, Distributed internal anomaly detection system for Internet-of-Things, *in: 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 319–320. 2016.

[8] J. Amaral, L. Oliveira, J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, *in: Communications (ICC), IEEE International Conference on*, pp. 1796–1801. 2014.

[9] Krimmling, S. Peter, Integration and evaluation of intrusion detection for CoAP in smart city applications, *in: Communications and Network Security (CNS), IEEE Conference on*, pp. 73–78. 2014.

[10] D. H. Summerville, K. M. Zach, Y. Chen, Ultra-lightweight deep packet anomaly detection for Internet of Things devices, *in: IEEE 34th International Performance Computing and Communications Conference (IPCCC*), IEEE, , pp.1–8.2015.

[11] R. Stephen and Dr. L. Arockiam, "Intrusion Detection System to Detect Sinkhole Attack on RPL Protocol in Internet of Things" *International Journal of Electrical Electronics & Computer Science Engineering* Volume 4, Issue 4(August, | E-ISSN : 2348-2273. 2021.

[12] Pavan Pongle and Gurunath Chavan, Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *International Journal of Computer Applications* Volume 121 - Number 9, July 2015.

[13] Shapla Khanam, Ismail Ahmedy and Mohd Yamani Idna Idris, An efficient detection of selective forwarding attacks in heterogeneous IoT Networks, *FCSIT*, pp 1-8. 2021.

[14] Kuan zhang et.al, sybil attacks and their defenses In the internet of things, *Ieee internet of things journal*, vol. 1, no. 5, october 2014.

[15] Tasnuva Mahjabin1, Yang Xiao1, Guang Sun2 andWangdong Jiang, A survey of distributed denial-of-service attack, prevention, and mitigation techniques, *International Journal of Distributed Sensor Networks*, Vol. 13(12). 2021.

[16] D. Chrun, Model-Based Support for Information Technology Security Decision Making, Ph.D. thesis, *University of Maryland* (2011).

[17] Tasnuva Mahjabin1, Yang Xiao1, Guang Sun2 andWangdong Jiang, "A survey of distributed denial-of-service attack,prevention, and mitigation techniques" *International Journal of Distributed Sensor Networks*, Vol. 13(12). 2021.

[18] Raviteja gaddam1 & dr. M. Nandhini, Analytical approach to enhance the intrusion detection in internet of things network,*international journal of latest trends in engineering and technology*, Vol.(9) issue(3), pp.258- 267.

[19] V.Gayathri, Malatesh S H, Smart Intrusion Detection System for HomeSecurity, International Journal of Science, Engineering and Technology, 20. Aastha Puri1, Nidhi Sharma, *A Survey On Intrusion Detection System, IJEDR* | Volume 5, Issue 2 | ISSN: 2321-9939.2021.

[20] Szarvas, M., Yoshizawa, A., Yamamoto, M., and Ogata, J.. Pedestrian detection with convolutional neural networks. *In IEEE Proceedings. Intelligent Vehicles Symposium*, pages 224–229. 2005.

[21] True, N. Vacant parking space detection in static images. University of California, *San Diego*, 17. 2007.

[22] Tsai, L. W., Hsieh, J. W., and Fan, K. C. Vehicle detection using normalized color and edge map. *IEEE Transactions on Image Processing*, 16(3):850–864.2007.

[23] Rashid, M. M., Musa, A., Rahman, M. A., Farahana, N.: Automatic parking management system and parking fee collection based on number plate recognition. *International Journal of Machine Learning and Computing*, vol. 2, no. 2, p. 94 .2012.

[24] Jian, M. S., Yang, K. S., and Lee, C. L. Modular RFID parking management system based on existed gate system integration. *WSEAS Transactions on Systems*, vol. 7, no. 6, pp. 706–716 .2008.

[25] Tsiropoulou, E. E., Baras, J. S., Papavassiliou, S., Sinha, S., RFID-based smart parking management system. *CyberPhysical Systems*, vol. 3, no. 1–4, pp. 22–41 .2021.

[26] Wei, L., Wu, Q., Yang, M., Ding, W., Li, B., and Gao, R., Design and implementation of smart parking management system based on rfid and internet. *In: International Conference on Control Engineering and Communication Technology*, pp. 17–20 .2012.

[27] Bi, Y., Sun, L., Zhu, H., Yan, T., Luo, Z., A parking management system based on wireless sensor network. *Acta Automatica Sinica*, vol. 32, no. 6, p. 968 .2006.

[28] Vera-Gómez, J. A., Quesada-Arencibia, A., García, C. R., Suárez Moreno, R., Guerra Hernández, F., An intelligent parking management system for urban areas. *Sensors*, vol. 16, no. 6, p. 931 .2016.

[29] Gandhi, B. K., Rao, M. K., A prototype for IoT based car parking management system for smart cities. *Indian Journal of Science and Technology*, vol. 9, no. 17, pp. 1–6 .2016.

[30] Sadhukhan, P., An IoT-based E-parking system for smart cities. In: International Conference on Advances in Computing, *Communications and Informatics (ICACCI)*, pp. 1062–1066 .2021.

[31] Abdulkader, O., Bamhdi, A. M., Thayananthan, V., Jambi, K., Alrasheedi, M., A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *In: 15th Learning and Technology Conference (L&T*), pp. 102–106 .2018.

[32] Lou, L., Li, Q., Zhang, Z., Yang, R., He, W., An IoTDriven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System. *IEEE Internet of Things Journal,* vol. 7, no. 11, pp. 11020–11029 .2020.

[33] Rehman, S. U., Gruhn, V., Recommended architecture for car parking management system based on cyber-physical system. *In: International Conference on Engineering and MIS (ICEMIS*), pp. 1–6 .2021.

[34] D. Oh, D. Kim, W. W. Ro, A malicious pattern detection engine for embedded security systems in the Internet of Things, *Sensors* 14 (12) 24188–24211. 2. 2014.

[35] T.-H. Lee, C.-H. Wen, L.-H. Chang, H.-S. Chiang, M.-C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN, in: Y.-M. Huang, H.-C. Chao, D.-J. Deng, J. J. J. H. Park (Eds.), Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Vol. 260 of Lecture Notes in Electrical *Engineering, Springer Netherlands*, pp. 1205–1213. 3. 2014.

[36] E. Cho, J. Kim, C. Hong, Attack model and detection scheme for botnet on 6LoWPAN, in: C. Hong, T. Tonouchi, Y. Ma, C.-S. Chao (Eds.), Management Enabling the Future Internet for Changing Business and New Computing Services, Vol. 5787 of Lecture Notes in Computer Science, *Springer Berlin Heidelberg*, pp. 515–518. 4. 2009.

[37] P. Kasinathan, C. Pastrone, M. Spirito, M. Vinkovits, Denial-of-service detection in 6LoWPAN based Internet of Things, in: Wireless and Mobile Computing, Networking and

Communications (WiMob), *IEEE 9th International Conference on*, pp. 600–607. 5.2013.

[38] J. Amaral, L. Oliveira, J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, *in: Communications (ICC), IEEE International Conference* on, 2014, pp. 1796–1801. 6. 2014.

[39] C. Liu, J. Yang, Y. Zhang, R. Chen, J. Zeng, Research on immunity-based intrusion detection technology for the Internet of Things, in: Natural Computation (ICNC), Seventh *International Conference on*, Vol. 1, pp. 212–216. 7. N. K..2011.

[40] Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, J. Isoaho, Distributed internal anomaly detection system for Internet-of-Things, *in: 13th IEEE Annual Consumer Communications Networking Conference (CCNC),* , pp. 319–320. 2016.

[41] 8. J. Amaral, L. Oliveira, J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for IPv6-enabled wireless sensor networks, *in: Communications (ICC), IEEE International Conference* on, , pp. 1796–1801. 2014.

[42] Krimmling, S. Peter, Integration and evaluation of intrusion detection for CoAP in smart city applications, in: Communications and Network Security (CNS), *IEEE Conference on*, , pp. 73–78. 2014.

[43] D. H. Summerville, K. M. Zach, Y. Chen, Ultra-lightweight deep packet anomaly detection for Internet of Things devices, *in: IEEE 34th International Performance Computing and Communications Conference (IPCCC),* IEEE, pp.1–8.2015.

[44] R. Stephen and Dr. L. Arockiam, "Intrusion Detection System to Detect Sinkhole Attack on RPL Protocol in Internet of Things" *International Journal of Electrical Electronics & Computer Science Engineering* Volume 4, Issue 4(August, | E-ISSN : 2348-2273. 2021.

[45] Pavan Pongle and Gurunath Chavan, Real Time Intrusion and Wormhole Attack Detection *in Internet of Things. International Journal of Computer Applications* Volume 121 - Number 9, July 2015.

[46] Chandra, H., Hadisaputra, K. R., Santoso, H., Anggadjaja, E., Smart parking management system: An integration of RFID, ALPR, and WSN. *In: IEEE 3rd International Conference*

on Engineering Technologies and Social Sciences (ICETSS), pp. 1–6 .2021.

[47] Al-Kharusi, H., Al-Bahadly, I., Intelligent Parking Management System Based on Image Processing. *World Journal of Engineering and Technology*, vol. 2014 .2014.

[48] Lin, S. F., Chen, Y. Y., Liu, S. C., A vision-based parking lot management system. *In: IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2897–2902 .2006

[49] Houben, S., Komar, M., Hohm, A., Lüke, S., Neuhausen, M., Schlipsing, M., On-vehicle video-based parking lot recognition with fisheye optics. *In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 7–12 .2013.

[50] Hamada, K., Hu, Z., Fan, M., Chen, H., Surround view based parking lot detection and tracking. In: IEEE Intelligent Vehicles Symposium (IV), pp. 1106–1111 [17] Suhr, J. K., Jung, H. G., Automatic parking space detection and tracking for underground and indoor environments. *IEEE Transactions on Indu*.2015.

[51] Jayakshei Dadaji Bachhava, Mechkul "Smart Car Parking System", *International Research Journal of Engineering and Technology (IRJET)* Volume: 04 Issue: 06, June -2021.

[52] Sagar Rane, Aman Dubey, Tejisman Parida, "Design of IoT Based Intelligent Parking System Using Image Processing Algorithms", *International Conference on Computing Methodologies and Communication*. 2021.

[53] Abhirup Khanna, Rishi Anand, "IoT based smart parking system", *international conference* of IoT and applications. 2016.

[54] Chungsan Lee, Youngtak Han, Soobin Jeon, Dongmahn Seo*, Inbum Jung "Smart Parking System for Internet of Things", *IEEE International Conference on Consumer Electronics*. 2016.

[55] Basavaraju S R " Automatic Smart Parking System using Internet of Things (IoT)", *International Journal of Scientific and Research Publications*, Volume 5, Issue 12, December 2015.

[56] FastPark System *website, http://www.fastprk.com.*

[57] Zhou, F., & Li, Q. (2014, November). Parking Guidance System Based on Zig Bee and Geomagnetic Sensor Technology. In Distributed Computing and Applications to Business, Engineering and Science (DCABES), *13th International Symposium on* (pp. 268-271). IEEE.2014.

[58] Jian, M. S., Yang, K. S., and Lee, C. L. Modular RFID parking management system based on existed gate system integration. WSEAS Transactions on Systems, vol. 7, no. 6, pp. 706–716 .2008.

[59] Tsiropoulou, E. E., Baras, J. S., Papavassiliou, S., Sinha, S., RFID-based smart parking management system. *CyberPhysical Systems*, vol. 3, no. 1–4, pp. 22–41 .2021.

[60] Wei, L., Wu, Q., Yang, M., Ding, W., Li, B., and Gao, R., Design and implementation of smart parking management system based on rfid and internet. *In: International Conference on Control Engineering and Communication Technology*, pp. 17–20 .2012.

[61] Bi, Y., Sun, L., Zhu, H., Yan, T., Luo, Z., A parking management system based on wireless sensor network. *Acta Automatica Sinica*, vol. 32, no. 6, p. 968 .2006.

[62] Vera-Gómez, J. A., Quesada-Arencibia, A., García, C. R., Suárez Moreno, R., Guerra Hernández, F., An intelligent parking management system for urban areas. *Sensors*, vol. 16, no. 6, p. 931 .2016.

[63] Gandhi, B. K., Rao, M. K., A prototype for IoT based car parking management system for smart cities. *Indian Journal of Science and Technology*, vol. 9, no. 17, pp. 1–6 .2016.

[64] Sadhukhan, P., An IoT-based E-parking system for smart cities. In: International Conference on Advances in Computing, *Communications and Informatics (ICACCI)*, pp. 1062–1066 .2021.

[65] Abdulkader, O., Bamhdi, A. M., Thayananthan, V., Jambi, K., Alrasheedi, M., A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *In: 15th Learning and Technology Conference (L&T)*, pp. 102–106 .2018.

[66] Lou, L., Li, Q., Zhang, Z., Yang, R., He, W., An IoTDriven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System.

IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11020–11029 .2020.

[67] Rehman, S. U., Gruhn, V., Recommended architecture for car parking management system based on cyber-physical system. *In: International Conference on Engineering and MIS (ICEMIS)*, pp. 1–6 .2021.

[68] Chandra, H., Hadisaputra, K. R., Santoso, H., Anggadjaja, E., Smart parking management system: An integration of RFID, ALPR, and WSN. *In: IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*, 2021, pp. 1–6 .2021.

[69] Al-Kharusi, H., Al-Bahadly, I., Intelligent Parking Management System Based on Image Processing. *World Journal of Engineering and Technology*, vol. .2014.

[70] Lin, S. F., Chen, Y. Y., Liu, S. C., A vision-based parking lot management system. *In: IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2897–2902 .2006.

[71] Houben, S., Komar, M., Hohm, A., Lüke, S., Neuhausen, M., Schlipsing, M., On-vehicle video-based parking lot recognition with fisheye optics. *In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 7–12 2013.