People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Chadli Bendjedid University
Faculty of Science and Technology
Department of Computer Science

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

MEMOIRE

Presented by

DJELLAL Basma

To obtain a diploma from

MASTER

Section :computer science

Option :Intelligent Computer System

Theme

A deep learning-based Botnet detection system

Defense date: 30 / 06 / 2022 In front of the jury:

Quality	First and last name	Grade	University
President	Mme. Bougarne.I	MCB	ChadliBendjedid El-Tarf
Reporter	**Mr. Betouil. AA**	**MCB**	**ChadliBendjedid El-Tarf**
Examiner	Mr. Bentrad. S	MCB	ChadliBendjedid El-Tarf

University Year: 2021/2022

Acknowledgements

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my sincere gratitude to my advisor Prof “Dr. BETOUIL Ali Abdelatif”, lecturer class B at the university “Taref”. for directing me in this work. I thank him for his availability, his follow-up, his precious advice and his help.

Besides my adviser, I would like to thank the rest of my thesis committee: Dr.BENTRAD.S and Mme. Bougarne.I., for their insightful encouraging comments, and the difficult questions that await me.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future.

I would like to thank my family: My brothers, my sisters, my uncle and my aunt, God forever.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly

Dedication

I dedicate this modest work

To all who know me, in particular,

To my father and mother for their sacrifices and support.

To my brothers and my sisters.

To all my friends and colleagues.

Without forgetting all the teachers who contributed to me

Primary, middle and secondary education up to higher

Education.

TABLE OF CONTENTS

To be generated automatically.

Table of contents

List of figures

Chapter 01

Chapter 02

Chapter03

Chapter 04

Chapter 05

List of paintings

List of acronyms

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
IDS	Intrusion Detection System

General Introduction

A network is a collection of devices that are connected over the Internet, and the total number of these devices increases every day. Undoubtedly, networks of financial and business institutions are continuously under security risk, which not only costs billions of dollars in damages and recovery, but also has a negative effect on their reputation. The increasing number of users affected by malicious software is becoming a critical problem. Botnets have become the main concern, since they are one of the biggest threats to security systems. Their popularity comes from their ability to control corporate mainframes by infiltrating any Internet-connected device that uses a digital video recorder.

A botnet can be defined as a network of compromised host devices that are used to carry out malicious activities. Desktop computers, smartphones, notebooks, and tablets are examples of such host devices. A botnet consists of three components: an attacker called a botmaster, a command and control (C&C) server, and an infected machine called a bot. The botmaster requires a C&C channel to command the bots and coordinate malicious attacks. Examples of C&C channels are IRC, HTTP, and P2P. According to communication protocols, C&C channels can be centralized or decentralized. Bots are used to send distributed denial of service (DDoS) attacks, phishing attacks, spam emails, and other forms of malicious attacks.

1. Project context and issues

Thus, it is important to detecting Botnet attacks to protect systems and resources. Botnet detection is a challenging research issue that has gained significant attention from researchers in the latest two decades. Ultimately, lots of botnet detection systems have been suggested in the literature [43], [44], [45], [46], [47]. These proposed methods employ different hypotheses and methods about the botnet for modeling and formalizing the traffic of the botnet. One of the most successful sorts of botnet detection systems is based on machine learning (ML) methods.

2. Motivation

Our bot network detection system is implemented as a model 1D-CNN-based on the Botnet dataset CICIDS2017, which includes good traffic in 191033 and 1966BotNet movement. Results show that our proposed model has achieved the best results in terms ofaccuracy (99.73 per cent)

and loss (0.0113). Moreover, our CNN model shows the best performance results compared to other current studies focusing on the discovery of the bot network

3. Objectives

Nowadays, the robot network has become a serious threat. In addition, their increased use complex evasion methods require more efficient detection methods. Hence, in this work, we offer a deep learning method, this method benefits from neuromorphic networks (CNN) to discover the robot network. CNN 78 uses a feature to classify traffic either "benign" or "robot."

4. Contents of the memoire

Chapter 1

In this chapter we presented the following; Computer Attacks, Then we talked about botnets, its history, and its architecture, IT security, its different techniques.

Chapter 2

In this chapter, we will present the Intrusion Detection System and its general model. We will then outline the types of Intrusion Detection Systems. IDS techniques are discussed below, before introducing the botnet detection techniques and concluding the chapter.

Chapter 3

In this chapter, we will introduce machine learning before discussing how it transitions to deep learning. Next, we'll introduce deep learning before ending the chapter on neural networks in the final section.

Chapter 4

In this chapter, several related works in the literature, developed for botnet detection, will be presented before ending the chapter with an analytical study.

Chapter 5

In this last chapter, our proposed model is presented. Kerasis used to implement this model and its optimization solutions are used to achieve the best results. The proposed model is put in a comparison with the state-of-the-art methods.

1. Introduction

The importance of information security has increased due to the increased utilization of computers and the Internet.

Cyber security's core function is to protect the devices we all use (smartphones, laptops, tablets and computers), and the services we access - both online and at work - from theft or damage.

Cyber security is important because smartphones, computers and the internet are now such a fundamental part of modern life, that it is difficult to imagine how we would function without them. From online banking and shopping, to email and social media, it is more important than ever to take steps that can prevent cyber criminals getting hold of our accounts, data, and devices.

In this chapter, we will present the different computer attacks. Then we will explain the botnet, its history, and its architecture. The IT security is explained below, before presenting its different techniques and concluding the chapter.

2. Computer Attacks

2.1DDOS Attacks

A denial of service (DoS) attack aims to disrupt the service provided by a network or server [1]. A denial-of-service (DoS) attack is to make computer resource unavailable to its intended users [2];DoS attacks generally achieve their goal by sending large volumes of packets that occupy a significant proportion of the available bandwidth [1].

Distributed denial of service (DDoS) attack is an attack using multiple distributed resources against targets, which will deprive authorized client from services [3].

DDoS attacks can be launched in two forms, The first one targets to crash a system by sending one or more carefully crafted packets, The second form DDoS is to use a large amount of traffic to exhaust the resources of a victim[4].

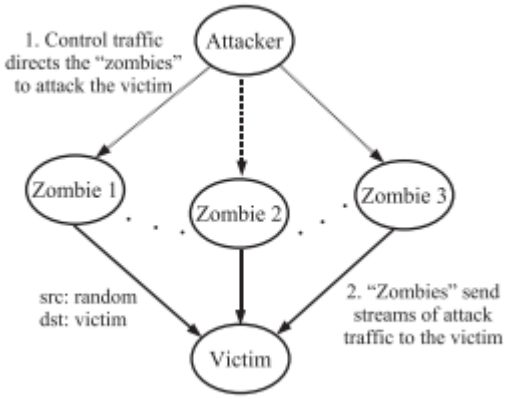

Figure 1.1 Structure of a typical DDoS attack [4].

2.2Man-in-the-Middle attack

The Man-In-The-Middle (MITM) attack is one of the most common attacks employed in the network hacking [5], it is an attack in which the attacker secretly intercepts and relays messages between two parties who believe they are communicating directly with each other [6].

In terms of ways to protect oneself against man-in-the-middle attacks, there are a couple of methods: ARP detection software and Static ARP entries [7].

D. Steinmetzer et al discussed four possible detection metrics to indicate whether an attack is happening or not; the switching between sectors, changes in the received signal strength, beacon interval length, and beacon counters to be valuable for a detection scheme [8].

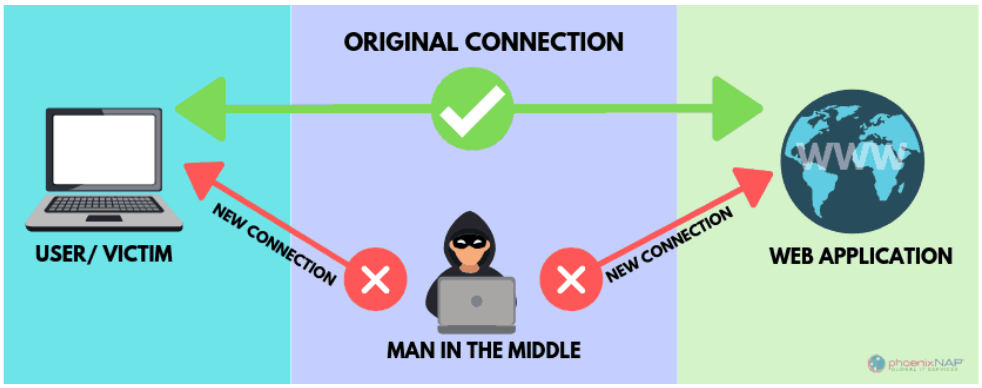

Figure 2.2 Structure of Man-in-the-Middle attack[9].

2.3Password attacks

Passwords are the most common method of authenticating users [10].

The main disadvantage of passwords is the so-called "classic password problem," this problem arises mainly from both of the following facts [11], using an alphabetical password with only

seven characters less status, which can easily be broken by well-known dictionaries, plus increasing the number of characters used in the password can be very difficult and time-consuming to break [12].

Hackers use brute-force, dictionary and rainbow table attacks to retrieve the input password plaintext from its hash value [10].

2.3.1 Brute force attacks

This attack makes use of the possible combinations of the supplied characters [12]

2.3.2 Dictionary Attacks

Dictionary attacks are regional attacks that run through a possible series of dictionary words [13].

2.4 SQL Attacks

SQL injection attacks (SQLIA) are among the most common database attacks, which try to access the sensitive data directly [14], they allow attackers to obtain unrestricted access to the databases [15].

Many technologies are vulnerable to SQL attacks such as PHP, JSP, ASP, ASP.net. In addition, Even the popular database organization MYSQL was hacked using this technique on March 27, 2011[16].

The proposed generic algorithm and Testing of web applications for SQL injection attack are substantial in scrutiny of its simple detection mechanism against SQL injection attacks [17].

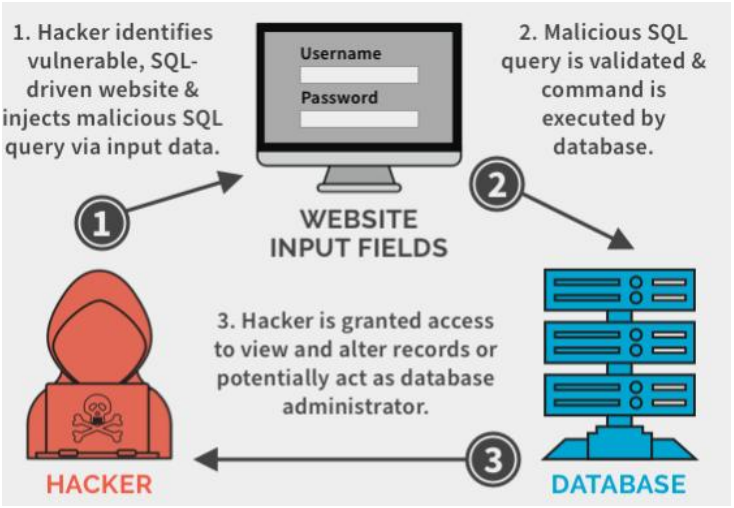

Figure 3.3SQL Injection Attack [18].

2.5 Malware

Malware is any kind of software intentionally developed for malicious purposes without user's awareness [19].

Malware is any code added, changed, or removed from a software system in order to intentionally cause harm or subvert the intended function of the system [20].

Viruses, worms, Trojans horses, Rootkits, spyware are all different forms and types of malicious programs [21].

Different types of malware can harm either slowing down the computer or replacing the web page in the browser and allowing hackers to control the system [21].

2.6 Phishing attacks

Phishing is a form of cybercrime that aims to deceive users into providing personal and/or financial information or to send money directly to the attacker [22].

Phishing attacks start with the phisher sending an email to the victim that appears to be from a legitimate organization, which contains links for which clicking on may either lead the victim to some false webpage where the user is asked to give his or her credentials or to install some spyware on the machine[23].

Phishing is no longer limited to email to but may also be carried out through voice messaging, SMS, instant messaging, social networking sites, and even multiplayer games [22].

3. Botnets

3.1 Definition

A botnet is a network of systems infected by bot malware [24], which are compromised computer systems or devices on the Internet [11].

These bots are commanded remotely by a botmaster, which is also called command and control (C&C) server; a botnet uses different communication protocols such as IRC, HTTP, P2P and IM [11].

Bots are used to carry out a wide variety of malicious and harmful actions against systems and services, including denial-of-service (DoS) attacks, spam distribution, phishing, and click fraud [25].

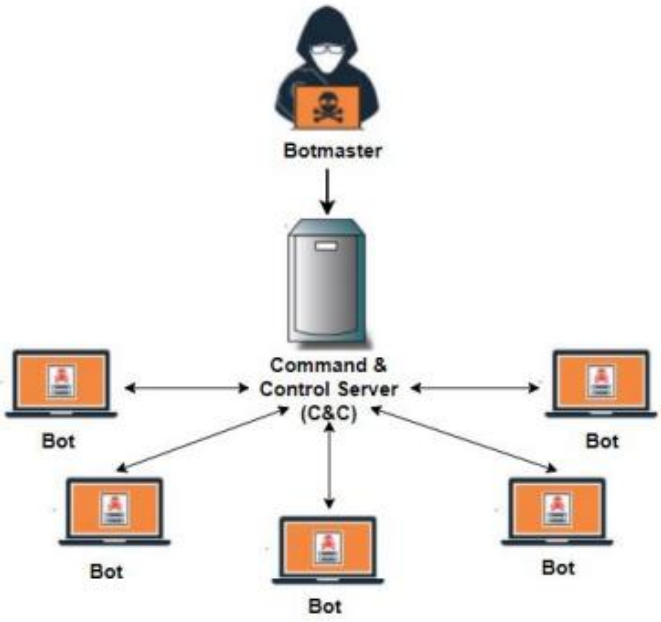

Figure 4.4 Centralized architecture of botnet [26].

3.2 Botnet history

Like many things on the Internet today, bots began as a useful tool without malicious overtones [27]

Computer worms that spread between networked machines appeared on the Advanced Research Projects Agency Network (ARPANET) in 1971[28]

EggDrop (1993) is the first popular bot network with a centralized structure. Developed by Robey Pointer using C programming language [29].

In 1998, GTbot became a widely known malicious IRC botnet used to launch DDoSattacks [28].

Agobot of 2002, an extremely robust collection of worms capable of packet sniffing, keylogging, rootkit installation, and DDoS attacks was one of the first large-scale botnets to require little programming ability [28].

Sinit botnet (2003) used a P2P protocol for communicating. Each infected machine is added to the P2P network where new Trojans are installed on the infected machines. Digital signatures used to encrypt the Trojans in the P2P list [29].

In the beginning of 2017, Twitter discovered 350,000 fake accounts that were part of a botnet [28].

By 2018, Facebook had discovered over 100,000 fake accounts and groups, which were used to influence the outcome of the 2016 United States' elections [28].

Concerns in the present and near future include the instability of the rapidly expanding social media botnets, the capacity for virtual machines and IoT to become botnet nodes [28].

3.3 Botnet architectures

Botnet buildings can be classified into two types by the method of interconnection of robots: client, server and peer-to-peer (P2P) model [30].

3.3.1 Client-serveur model

Client–server model is a typical structure of botnet. It is composed of a botmaster, several C&C servers and a lot of bots (or botclients) which are infected by the botnet viruses. A botmaster is like a commander; it controls the whole botnet remotely by sending commands to a C&C server. A C&C server, on the other hand, passes the commands from a botmaster to the bots. Additionally, a botmaster can learn the number of botclients and their information from the C&C servers in a botnet. The rest of the bots, the compromised devices, execute the given commands stealthily in the meantime. This traditional kind of botnet, however, can be easily discovered since we can track the hostname or the IP address of C&C servers and add them to a blacklist. To avoid from revealing themselves, some botnets render a large number of hostnames dynamically by domain generation algorithm (DGA) [30].

3.3.2 Peer-to-peer model

Instead of communicating with a centralized server, a bot can act as a commander and a receiver at the same time.

Firstly, to find out its neighbors in the botnet, a bot keeps probing random IP addresses until it finds out other infected devices.

Then, when a bot receives a command, it executes the command and distributes the command to other bots.

In this way, it would be hard to find suspicious IP addresses and track down the source of the botnet compared to centralized botnet architecture [30].

4. IT security

Information technology security (or Cyber security) is the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks [31].

5. Security techniques [31]

- Update your software and operating system.
- Use anti-virus software.
- Use strong passwords.
- Do not open email attachments from unknown senders.
- Do not click on links in emails from unknown senders or unfamiliar websites.
- Avoid using unsecure Wi-Fi networks in public places.

6. Conclusion

In this chapter, we have presented the different computer attacks, where we have presented a general overview of these attacks. Before discussing the botnet architectures, we have presented a general overview of the botnet. We have also presented a brief history of the botnet. IT security and its different techniques were discussed in the last part of this chapter.

In the next chapter, we will focus on Intrusion Detection Systems, including the different techniques developed to detect the botnets.

Chapter2:Intrusion Detection System

1. Introduction

Traditional intrusion prevention techniques, such as firewalls, access control or encryption, have failed to fully protect networks and systems from increasingly sophisticated attacks and mal-wares. As a result, intrusion detection systems (IDS) have become an indispensable component of security infrastructure to detect these threats before they inflict widespread damage.

In this chapter, we will present the Intrusion Detection System and its general model. We will then outline the types of Intrusion Detection Systems. IDS techniques are discussed below, before introducing the botnet detection techniques and concluding the chapter.

2. Definition

An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for the harmful activity or policy breaching. Any malicious venture or violation is normally reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system integrates outputs from multiple sources and uses alarm filtering techniques to differentiate malicious activity from false alarms [37].

3. General model

The IETF's IDWG (Intrusion Detection Working Group) has defined a generic intrusion detection model that represents the functionality common to all IDSs (whether they use the behavioral approach or the scenario approach).

Figure 2.1 proposed by the IDWG (Intrusion Detection Working Group) shows the generic process of intrusion detection systems.

The administrator configures the various components (sensor (s), analyzer (s), manager (s)). The sensors access the raw data, filter it and format it to return only the interesting events to an analyzer. Analyzers use these events to decide whether or not an intrusion is present and, if

necessary, send an alert to the manager (who notifies the human operator). A possible reaction can be carried out automatically by the manager or manually by the operator [32]

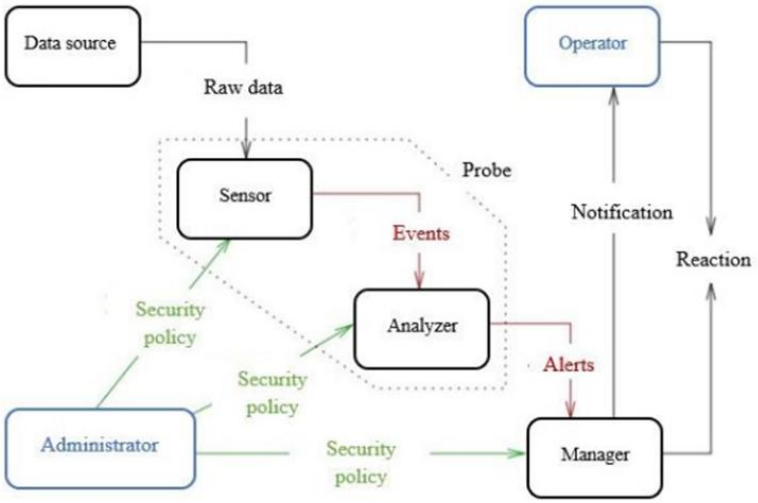

Figure 2.1 Generic intrusion detection model proposed by the IDWG [32].

4. Types of IDS

4.1 Host based IDS (HIDS)

HIDS was the first developed type of intrusion detection [33], HIDS monitors and analyzes the information collected from a specific host machine. Upon detection of deviation from expected behavior, it reports the existence of attack. The efficiency of HIDS depends on chosen system char-acteristics to monitor. Each HIDS detects intrusion for the machines in which it is placed as show in the figure 2.2 [34].

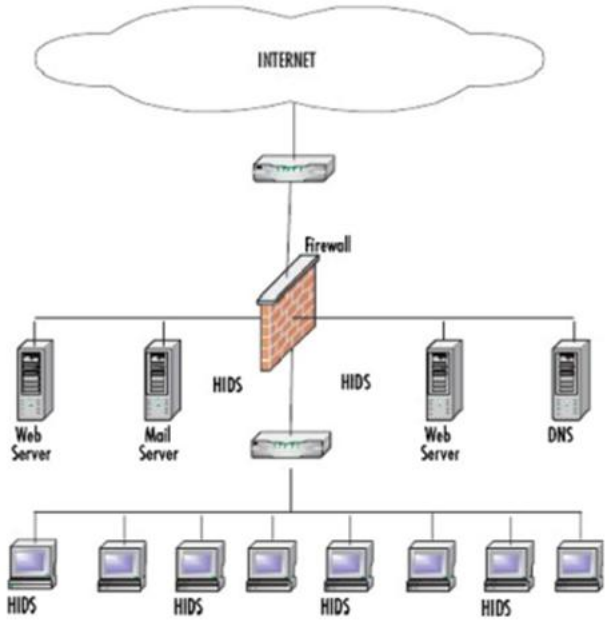

Figure 2.2 Host based intrusion detection system (HIDS)[34].

4.2 Network based IDS (NIDS)

NIDS monitors network traffic to detect malicious activity such as DoS attacks, port scans or even attempts to crack into computers. The information collected from network is compared with known attacks for intrusion detection.

NIDS has stronger detection mechanism to detect network intruders by comparing current behavior with already observed behavior in real time.

NIDS mostly monitors IP and transport layer headers of individual packet and detects intrusion activity.

NIDS uses signature based and anomaly based intrusion detection techniques. NIDS has very limited visibility inside the host machines. If the network traffic is encrypted, there is no effective way for the NIDS to decrypt the traffic for analysis [34].

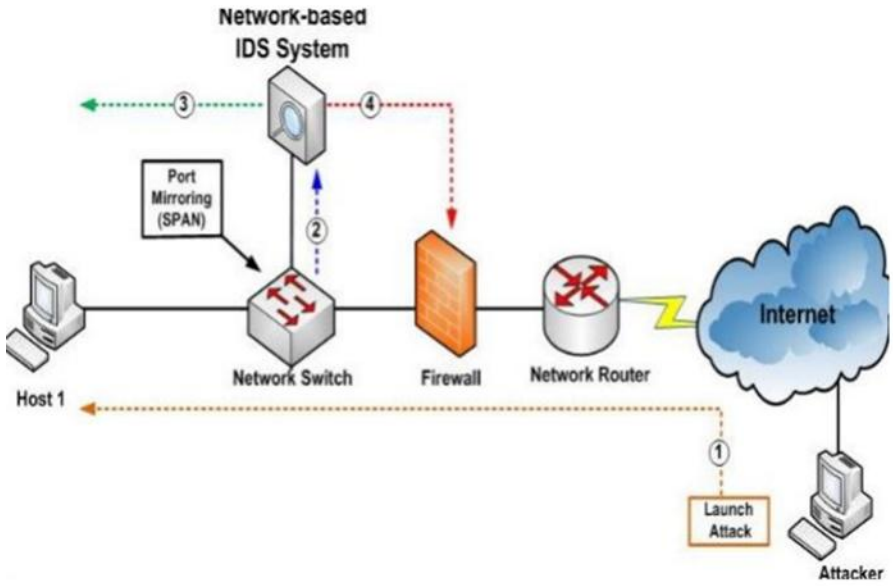

Figure 2.3 Network based IDS [33].

4.3 Hybrid based IDS

Combines two types or more of IDS to achieve the advantages of IDS, complete an accurate detection such as Double Guard that uses host ids, and network IDS. However, Hybrid based IDS takes a long time in analyzing data. Figure 2.4representsHybridbased IDS [33].

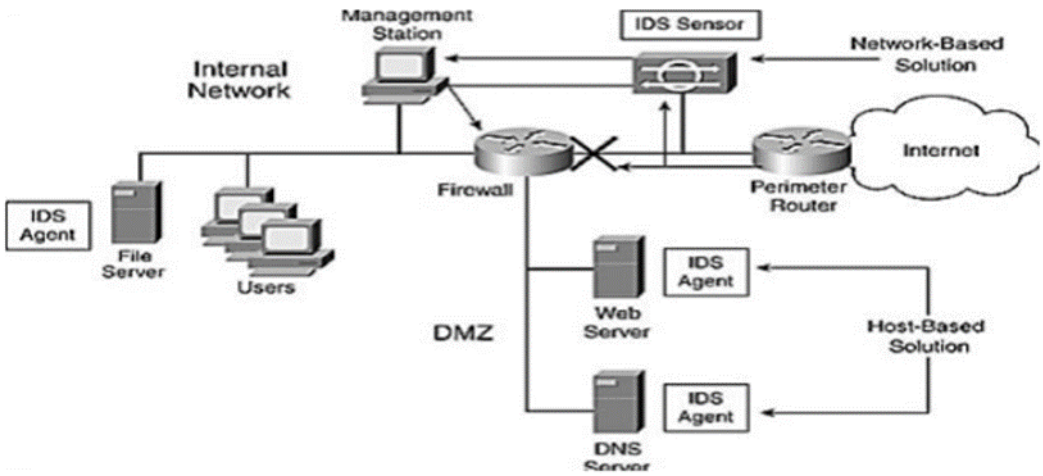

Figure 2.4 Hybrid based IDS [33].

5. IDS Techniques

We can detect intrusions using many techniques, including the following: Signature based detection, Anomaly detection, Artificial neural network (ANN) based IDS, Association rule

based IDS, Support vector machine (SVM) based IDS, Genetic algorithm (GA) based IDS, Hybrid techniques [34].

6. Botnet Detection Techniques [34]

6.1 Honeynet Based Detection

This technique is used to detect and monitor the behavior of botnet; Honeynet based detection works with honeypot and honeywall.

Honeypot term refers to a vulnerable pot which can be easily compromised. It also vulnerable with an intention of become a part of botnet and attracts botmaster to infect it.

Honeywall is a term used for software which is used to collect and control information from honeypot.

6.2 Intrusion Detection System Based Detection

6.2.1 Signature Based

This technique works with the help of using the signature of existing Botnet it creates a database with existing botnets. Then it compare the signature of network traffic with the existing Bots. We can find botnet if the signature has existed in the database.

This technology can only track previously known bots, and it is difficult for them to track new botnets, this technique cannot handle Zero Day Botnets.

6.2.2 Anomaly Based

This technique work with monitoring the network traffic. By differentiating the malicious traffic from normal traffic botnet.

6.2.3 DNS Based

This technique is a combination of both anomaly based and signature based techniques. It is also based on an abnormal traffic, generated by botnet

DNS based techniques works with information collected from DNS queries, It can also trace the location of C&C server and Botmaster behind Botnet.

6.3 Data Mining Based

Data mining technique capture the high volume of network traffic and find the malicious traffic from it.

But to detect C&C server and recognize its pattern used for Botnet is very difficult. As encrypted C&C Server hide itself with normal traffic so to detect this kind of Botnet attacks Data mining techniques are used with data Classification and Clustering.

6.4Detection Techniques using Machine Learning

Many Machine Learning techniques, such as Decision Trees and Classifiers, are used to detect the Botnet. These approaches are also effective for detecting chat Bots, however, they do not detect the C&C Server

7. Conclusion

In this chapter, we first presented Intrusion Detection Systems and their general model. We then discussed the three types of Intrusion Detection Systems: Host-based intrusion detection systems (HIDS), network-based intrusion detection systems (NIDS), and Hybrid based IDS. We had discussed Intrusion Detection Systems techniques before presenting the botnet detection techniques.

In the next chapter, we will introduce machine learning, deep learning, and neural networks.

Chapter3:Deep Learning

1. Introduction

Artificial Intelligence (AI), the study and engineering of intelligent machines capable of performing the same kinds of functions that characterize human thought.

The concept of AI dates from ancient times, but the advent of digital computers in the 20th century brought AI into the realm of possibility. AI was conceived as a field of computer science in the mid-1950s.

The term AI has been applied to computer programs and systems capable of performing tasks more complex than straightforward programming, although still far from the realm of actual thought.

While the nature of intelligence remains elusive, AI capabilities currently have far-reaching applications in such areas as information processing, computer gaming, national security, electronic commerce, and diagnostic systems.

Machine learning is the study of computer algorithms that provides systems the ability to automatically learn and im-prove from experience. It is generally seen as a sub-fieldof artificial intelligence.

Machine learning algorithms al-low the systems to make decisions autonomously withoutany external support. Such decisions are made by findingvaluable underlying patterns within complex data.

Deep Learning (DL) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

In this chapter, we will introduce machine learning before discussing how it transitions to deep learning. Next, we'll introduce deep learning before ending the chapter on neural networks in the final section.

2. Machine learning

2.1 Definition

Machine learning is the technology of developing computer algorithms that are able to emulate human intelligence.

It draws on ideas from different disciplines such as artificial intelligence, probability and statistics, computer science, information the-ory, psychology, control theory, and philosophy.

The most important property of these algorithms is their distinctive ability to learn the sur-rounding environment from input data with or without a teacher [36]

2.2 Types of machine learning [37]

2.2.1 Supervised Learning

Supervised learning is applied when the data is in the formof input variables and output target values. The algorithmlearns the mapping function from the input to the output; it can be divided into two categories:

CLASSIFICATION; the output variable is one of some known number of cat-egorie.

REGRESSION; the output variable is a real or a continuous value.

2.2.2 Unsupervised Learning

Unsupervised learning is applied when the data is availableonly in the form of an input and there is no correspond-ing output variable.

Such algorithms model the underlyingpatterns in the data in order to learn more about its charac-teristics.

One of the main types of unsupervised algorithms is clus-tering.

2.2.3 Semi-supervised Learning

This is an intermediate between su-pervised and unsupervised learning techniques.

These algorithms are trained using a combination of labeled and unlabeled data.

A basic procedure involved is that first similar data is clustered using an unsupervised learning algorithm and then existing labeled data is used to label the rest of the unlabeled data.

2.2.4 Reinforcement Learning

Reinforcement learning is applied when the task at handis to make a sequence of decisions towards a final reward.

During the learning process, an artificial agent gets eitherrewards or penalties for the actions it performs. Its goal isto maximize the total reward.

3. From machine learning to deep learning

For a long period, machine learning was the principal tool. However, with the Appearance of Big Data machine learning approaches have turned to the notion of deep learning, where new architectures have been created, more powerful and efficient, to face the Big Data concept. Since 2006, deep learning has become a very common tool in much research [32].

4. Deep Learning

4.1 Definition

Deep learning is the subfield of machine learning that is devoted to building algorithms that explain and learn a high and low level of abstractions of data that traditional machine learning algorithms often cannot.

It is being used in technologies such as self-driving cars, image recognition on social media platforms, and translation of text from one language to others [38].

4.2 Why and when to apply DL

DL is employed in several situations where machine intelligence would be useful:

- Absence of a human expert (navigation on Mars).
- Humans are unable to explain their expertise (speech recognition, vision, and language understanding).
- The solution to the problem changes over time (tracking, weather prediction, preference, stock, price prediction).
- Solutions need to be adapted to the particular cases (biometrics, personalization).
- The problem size is too vast for our limited reasoning capabilities (calculation webpage ranks, matching ads to Facebook, sentiment analysis) [39].

5. Neural Networks

5.1 Artificial neural networks (ANNs)

The ANN, inspired by the biological neural network, is a set of interconnected neurons, or nodes, where connections are weighted and each neuron transforms its input into a single output by applying a non-linear activation function to the sum of its weighted input[40].

5.2 Convolutional Neural Network (CNN)

The convolutional neural network (CNN) is a specialized feed forward neural network that was designed to process multi-dimensional data, A CNN architecture is typically comprised of convolutional layers, pooling (subsampling) layers, and fully connected layers [40].

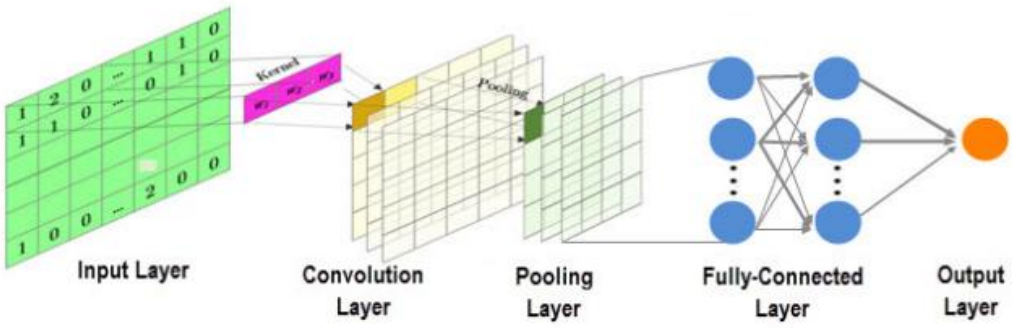

Figure 3.1 The architecture ofConvolutional Neural Network model [41]

5.3 Recurrent Neural Network (RNN)

The recurrent neural network is a typical sequential learning model. It learns features for the series data by a memory of previous inputs that are stored in the internal state of the neural network [42].

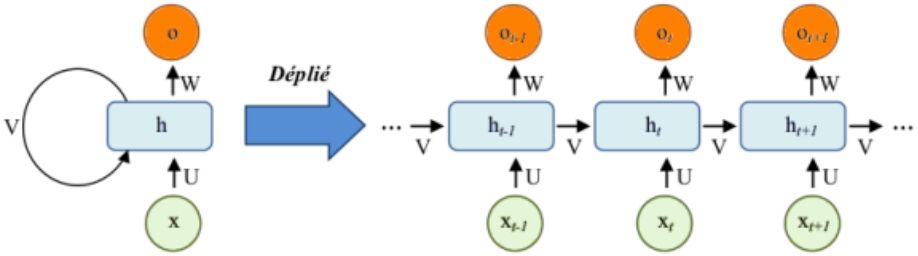

Figure 3.2 The architecture of Recurrent Neural Network model [41].

6. Conclusion

Machine learning and deep learning were introduced in this third chapter. We started with the definition of machine learning as a subfield of artificial intelligence before discussing its different types, including supervised, unsupervised, semi-supervised Learning, and reinforced learning. Then we presented deep learning before ending the chapter on the different architectures of neural networks in the last section.

In the next chapter, we will present some related work found in the literature, which uses different techniques such as machine learning to detect botnets.

Chapter4:Evaluation and Discussion

1. Introduction

Currently, the increasing number of botnets has become a serious security issue. Many approaches for detecting botnets have been developed in recent years.

Botnet detection happens in network computers and hosts. Some approaches concentrate on incoming system data to identify unusual behaviors of botnet while others concentrate on the analysis of package contents. These methods employ a lot of resources and time-consuming data analysis. Although these approaches have benefits compared to other existing approaches, these approaches need optimization for requiring less memory for the incoming packets.

In this chapter, several related works in the literature, developed for botnet detection, will be presented before ending the chapter with an analytical study.

2. Related Works

2.1 Botnet detection via mining of traffic flow characteristics

In [43],the authors proposed a botnet detection via mining of traffic characteristics,the proposed system extracts important features and these features are used to model the behavior of network flows to identify the botnets, the functional block diagram of the proposed system is given in Figure 4.1. The proposed system was applied to different datasets.

They tested three classifiers; Boosted decision tree (AdaBoostM1+J48), Naive Bayesian (NB), and Support vector machine (SVM), and evaluated the results using five metrics; Precision, Recall, F-measure, Accuracy, and False Positive Rate (FPR).

The naive Bayesian has reached better results, where the precision 97.8%, Recall 96.1%, F-measure 96.9%, Accuracy 99.14%, and FPR 4.81%.

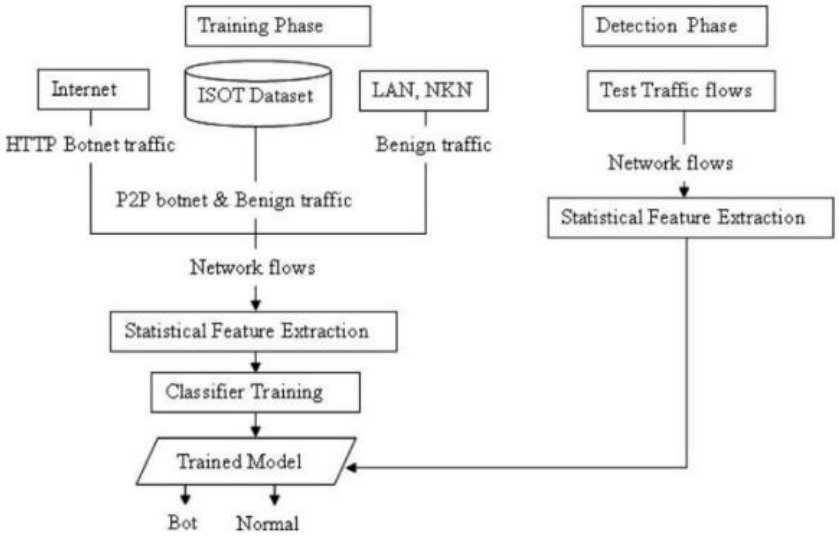

Figure 4.1 Functional block diagram of the proposed scheme [43].

2.2Detection of Botnet traffic by using Neuro-fuzzy based Intrusion Detection

In [44],the authors proposed a system of botnet detection of traffic by utilizing Neuro-fuzzy based the intrusion detection,The proposed model is applied on a dataset, which was produced by deploying an application on the Eucalyptus cloud, then assaulting it with several open-source botnet simulation tools.

The proposed system's architecture is illustrated in Figure 4. 2. They compared their model with four classifiers; Naïve Bayes, Multilayer Perceptron, Decision Tree, and Support vector machine (SVM), and evaluated the results using one metric; Accuracy.

The model achieved the best results compared to the other classifiers, where the Accuracy of 96.8%.

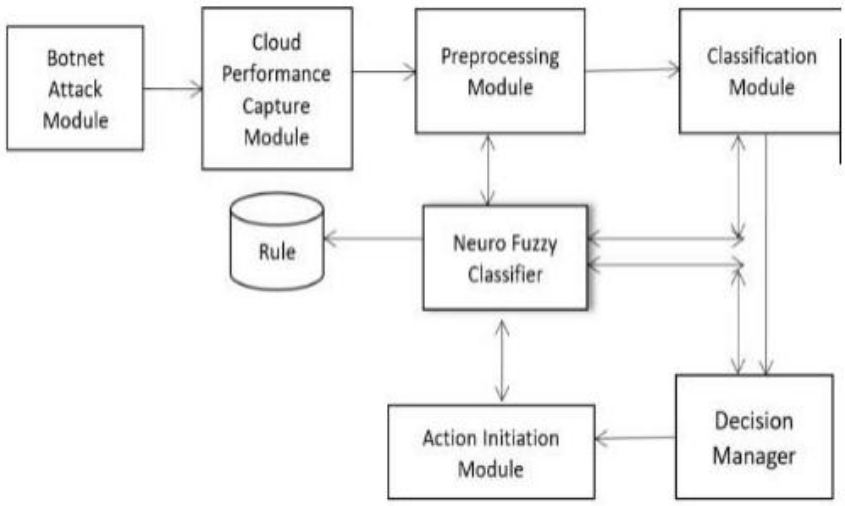

Figure 4.2 System Architecture [44].

2.3Botnet Detection Based on Machine Learning Techniques Using DNS Query Data

In [45], the authors proposed a system to detect botnets based on ML (machine learning) and utilizing the Domain Name Service Query Data. Figure 4. 3 is shown the proposed model. The proposed system was applied to a dataset, which includes benign domain names and malicious domain names using by botnets. They tested four classifiers; k-Nearest Neighbor (KNN), Naïve Bayes, C4.5, and Random Forest (RF), and evaluated the results using five metrics; FPR (False positive rate), PPV (Positive predictive value), ACC (Accuracy), TPR (True positive rate), and F1 (F1 measure).

The model achieved the best results using the Random Forest, where the PPV 90.70%, FPR 9.30%, TPR 91.00%, ACC 90.80%, and F1 90.80%.

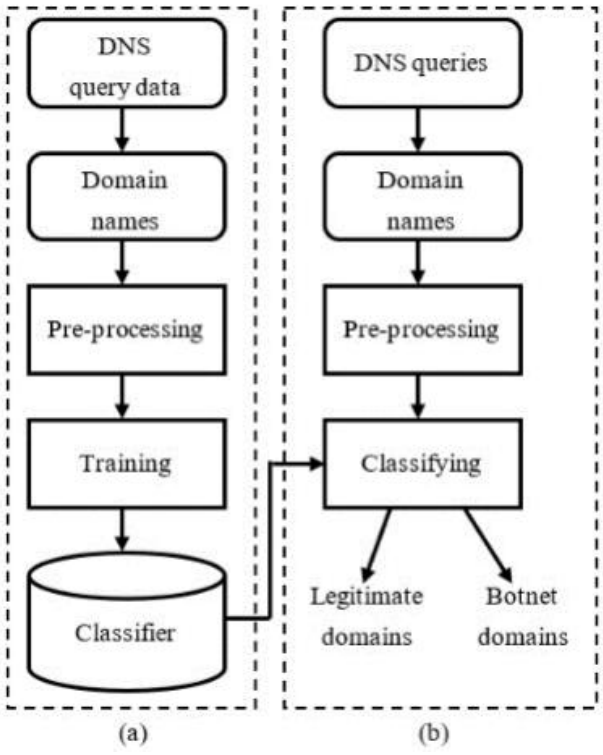

Figure 4.3 Proposed botnet detection model based on machine learning using Domain Name Service (DNS) query data: (a) training phase and (b) detection phase [45].

2.4Botnet traffic detection using RPCA and Mahalanobis Distance

In [46], the authors proposed a system for Botnet traffic detection utilizing RPCA (Robust Principal Component Analysis) and MD (Mahalanobis Distance). Figure 4. 4 is shown the

proposed model structure. The proposed system was applied to the CTU-13 dataset, which was divided into 13 scenarios.

They used each of the 13 scenarios separately and evaluated the results using three metrics; Precision, Recall, and F1-Score.

The model achieved the best results in scenario 7, where the Precision 95.9%, Recall 95%, and F1-Score 95.5%.

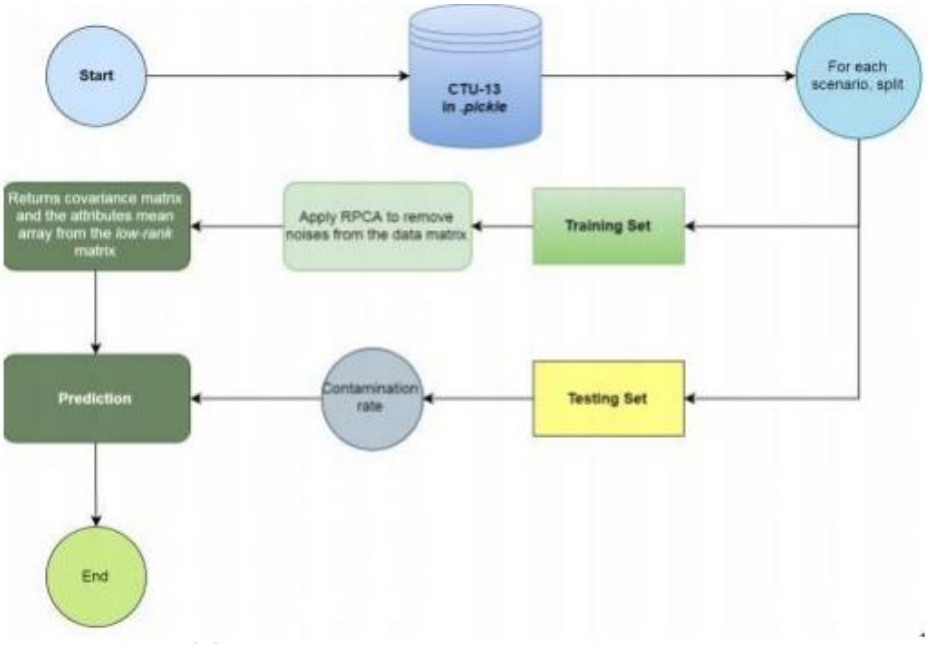

Figure 4.4 Model structure [46].

2.5Botnet detection using negative selection algorithm, convolution neural network and classification methods

In [47], the authors proposed a system for Botnet detection utilizing a negative selection algorithm, CNN, and classification methods. the proposed method framework is illustrated in Figure 4. 5. The proposed system was applied to ISCX 2012 dataset.

They compared their model with six machine learning methods; K-Neighbors, Random Forest, SVM, GaussianNB, CNN, and LSTM, and evaluated the results using three metrics; Accuracy, Precision, and Recall.

The model achieved the best results compared to the other machine learning methods, where the Accuracy 99%, Precision 98.0%, and Recall 100%.

Figure 4.5 Framework of proposed method [47].

3. Analytical study

Datasets	**Reference**	**Precision %**	**Recall %**	**F-measure %**	**Accuracy %**	**FPR %**
Differentdatasets	[43]	97.8	96.1	96.9	99.14	4.81
Generateddataset (Eucalyptus cloud)	[44]	X	X	X	96.8	X
Differentdatasets	[45]	X	X	90.8	90.8	9.30
CTU-13 dataset	[46]	95.9	95	95.5	X	X
ISCX 2012 dataset	[47]	98	100	X	99	X

Table 4.1 Comparison between related works.

As a result of the absence of many performance metrics, we choose Accuracy as a comparison metric. Table 4.6 shows that the best botnet detection system is [43] where its accuracy achieved 99.14%.

4. Conclusion

Currently, Botnet networks are one of the most cybersecurity threats. It is therefore important to detect Botnet attacks to protect systems and resources. Tracking and discovering Botnet have been common research topics in recent years.

Although there are a lot of robot detection techniques, many of them are ineffective in detecting existing robot networks. In the next chapter, we will make our contribution, the deep learning approach to robot detection based on the Synthetic Neural Networks (CNN).

Chapter5: CONTRIBUTION

1. Introduction

The attackers use botnets to perform illegal activities such as, DDoS, spamming, click fraud, phishing, etc. Over the past years, even though many researchers have developed various botnet detection methods, there is a necessity for new techniques to detect new botnets irrespective of C & C protocols, structures in the early stage itself.

In this chapter, we will present a deep learning approach for botnet detection based on Convolutional Neural Networks (CNN). Our proposed botnet detection system will be implemented as a 1D-CNN-based model using Keras in Google Colab.

CICIDS2017 is a public dataset that has 78 features. Using the information stored in this database to generate our model requires a whole process to make them suitable as learning databases. In this context, we apply preprocessing process and normalization process on this dataset.

We then illustrate the architecture of the proposed approach. The evaluation results will then be presented and discussed. This chapter will end with a comparative analysis and conclusion.

2. Classification in deep learning

A Classification problem is when the output variables are a class or a category. Depending on the number of classes, we have a binary classification problem (two classes), multi-class classification problem (more than two classes), and multi-label classification problem (multiple classes) [48].

2.1 Classification in CNN

CNN is used in many areas includingImage data, Classification prediction problems, and Regression prediction problems. More generally, CNNs work well with data that has a spatial relationship [48].

2.2 Classification in RNN

RNNs in general and LSTMs in particular have received the most success when working with sequences of words and paragraphs, generally called natural language processing [48].

2.3 CNN advantages

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs, it can learn the key features for each class by itself [49].

3. Hyperparameters problem

Deep learning (DL) models are widely employed in a variety of application domains, including image and text classification, natural language processing (NLP). This is due to their great performance in the problems of data analytics.

In general, developing an effective deep learning model is a time-consuming and complex process that including the determination of the best algorithm and the best model structure by optimizing the hyperparameters of the algorithm.

Hyperparameter tuning refers to the process of building the best model architecture with an optimal hyperparameter configuration.

Some key reasons for employing Hyperparameter methods to DL models such as it minimizes the amount of human effort necessary and increases the performance of deep learning models [50].

4. Implementation

4.1 Dataset description

CICIDS2017 is a well-known Intrusion Detection System (IDS) dataset that was used in our work. It is a public dataset that can be accessed for free at (http://www.unb.ca/cic/datasets/IDS2017.html). It consists of actual data that was collected based on the behavior of a 25-users network based on the FTP, HTTPS, HTTP, email, and SSH protocols. CICIDS2017 includes benign and different malicious attack traffic such as Infiltration, Web Attack, Botnet, and Heart bleed..., etc. We selected the file that contains a botnet attack

(Friday-WorkingHours-Morning.pcap_ISCX) and tested it in our proposed model. This file includes 191033 benign traffic and 1966 Botnet traffic [51].

4.2 Data Preprocessing

The existence of repetitive and undesired values in the dataset is considered a common problem that impacts the performance and the accuracy of the detection system.

To find a solution to this problem, follow Figure 5.1, after we have done our library calls and dataset we collect them in one table.

```
import tensorflow as tf
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
dataset = pd.read_csv('/content/drive/MyDrive/dataset/botdetection/Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv')
ds = pd.read_csv('/content/drive/MyDrive/dataset/botdetection/Friday-WorkingHours-Morning.pcap_ISCX.csv')
dataset.append(ds)
#dataset.append(dsthursday)
display(dataset)
```

Figure 5.1 Preprocessing of dataset.

4.3 Data normalization

The attributes values in CICIDS2017 dataset show a diverse range of values. Thus, data normalization required to support both the accuracy and performance of the developed Botnet detection framework.

In current work, Minmax normalization proposed as one of the normalization strategies that transform the dataset values into a certain suitable processing range per attribute, as formulated in the following equation (1).

$$norm\_value = Xj - Xmin / Xmax - Xmin \qquad (1).$$

Where i is the counter of values Xi of an attribute (X), Xmin and Xmax are minimum and maximum values of an attribute (X). The new range of data would be within a 0–1 range [51].

```
maxes = []
mins = []
dataset = dataset.dropna()
inputs = dataset.iloc[:,0:len(dataset.columns)-1]
scalMapper = {"BENIGN":0.0,"Bot":1.0,"PortScan":1.0}
outputs = dataset.iloc[:,len(dataset.columns)-1:].replace(scalMapper)
outputs = outputs.to_numpy(dtype=float)
outputs.reshape(-1,1)
for col in inputs.columns :
  max = np.max(inputs[col])
  min = np.min(inputs[col])
  maxes.append(max)
  mins.append(min)
  if max-min>0 :
    inputs[col] = (inputs[col]-min)/(max-min)
display(inputs)
outputs
```

Figure 5.2 Normalization of dataset.

We retrieved the output vector "outputs", which represents the vector of class labels, from the CSV file of dataset, and replaced each BENIGN with the value "0" and Bot with "1". Next, we applied normalization to the dataset to give ourselves limited values between -1 and 1. Show figure 5.2.

4.4Google Colab

Colaboratory is a Google research project, and it was created to help disseminate machine learning education and research.

By using Colab, programmers could write, edit, and execute code in python. Additionally, popular python libraries such as NumPy and Matplotlibcould be used to analyze and visualize data. It also allows us to integrate open-source libraries named PyTorch, TensorFlow, Keras, and OpenCV [52].

4.5Keras

It is an open-source neural network library written in python. It has the capability to run on top of TensorFlow or Theano. It is designed for enabling fast experimentation with deep neural networks. The main model type of Keras is a sequence of layers called Sequential, and it is a linear stack of layers [52].

4.6Proposed model

Our proposed model is a 1D-CNN model. As shown in Figure 5. 4, it contains five layers; an input layer (Convolution1D with 64 neurons and 78 inputs), three hidden layers (global_max_pooling1d, dense, dropout), and an output layer.

1D-CNN accepts the input shape of data in the 3D form therefore we transformed data to a 3D shape using reshape function as shown in Figure 5.3.

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(inputs, outputs, test_size=0.4, random_state=42)
     X_train.replace([np.inf,-np.inf],np.nan,inplace=True)
     X_train.fillna(0,inplace=True)
     X_test.replace([np.inf,-np.inf],np.nan,inplace=True)
     X_test.fillna(0,inplace=True)
     X_train = X_train.to_numpy(dtype=float)
     X_test = X_test.to_numpy(dtype=float)
     X_train = np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1))
     X_test = np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
     y_train = y_train.reshape(y_train.shape[0],y_train.shape[1])
     y_test = y_test.reshape(y_test.shape[0],y_test.shape[1])
     X_train.shape
```

Figure 5.3 Reshaping data using the reshape function.

We used “He Uniform” as kernel initializer, where the kernel size is 3, also we used Rectified Linear Unit (Relu) as the activation function for the input and the hidden layers, the Sigmoid function for the output layer, Adam as an optimizer and the binary cross-entropy loss function as a loss function. The detailed configuration is listed in Figure 5. 5.

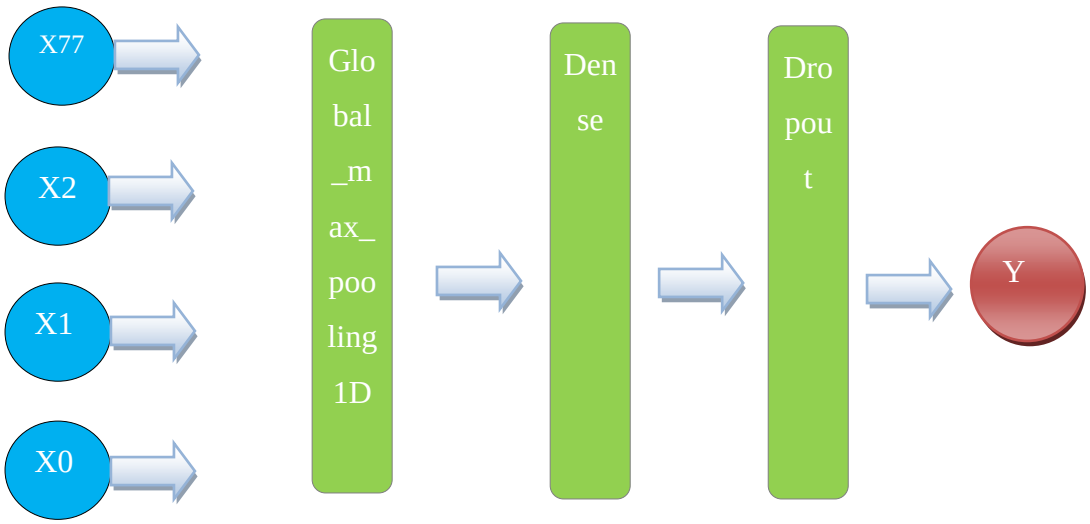

Figure 5.4 Proposed model layers.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 73, 64)            448

 global_max_pooling1d (Globa  (None, 64)               0
 lMaxPooling1D)

 dense (Dense)               (None, 64)                4160

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 4,673
Trainable params: 4,673
Non-trainable params: 0
```

Figure 5.5 The detailed configuration of our proposed model.

The following figure (Figure 5.6) is shown the instructions that we used to develop our model using the Keras library.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, Flatten,Dropout,MaxPooling1D,GlobalMaxPooling1D
from tensorflow.keras.callbacks import EarlyStopping
model = Sequential()

verbose, epochs, batch_size = 1, 30, 32
n_timesteps, n_features, n_outputs = X_train.shape[1], X_train.shape[2], y_train.shape[1]

def evaluate_model(trainX, trainy, testX, testy):
    model.add(Conv1D(filters=64, kernel_size=6,padding='valid', activation='relu', input_shape=(n_timesteps,n_features)))
    model.add(GlobalMaxPooling1D())
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    model.summary()
    callback = EarlyStopping(monitor="val_loss", patience=2, restore_best_weights=True)
    # fit network
    hist = model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose,
                     callbacks=[callback],validation_data=(X_test,y_test))
```

Figure 5.6 The instructions for developing our model using Keras.

To train and evaluate our model, we randomly divided the dataset using the function “train_test_split” into two parts: training (60% of the dataset) and testing (40% remaining) as shown in Figure 5. 7.

```
X_train, X_test, y_train, y_test = train_test_split(inputs, outputs, test_size=0.4, random_state=42)
```

Figure 5.7 Split dataset into training and testing set.

4.7Results and discussion

In this section, our model will be evaluated based on different performance measures confusion matrix (figure 5.8),To better demonstrate the effectiveness of our model, compare with some works in literature

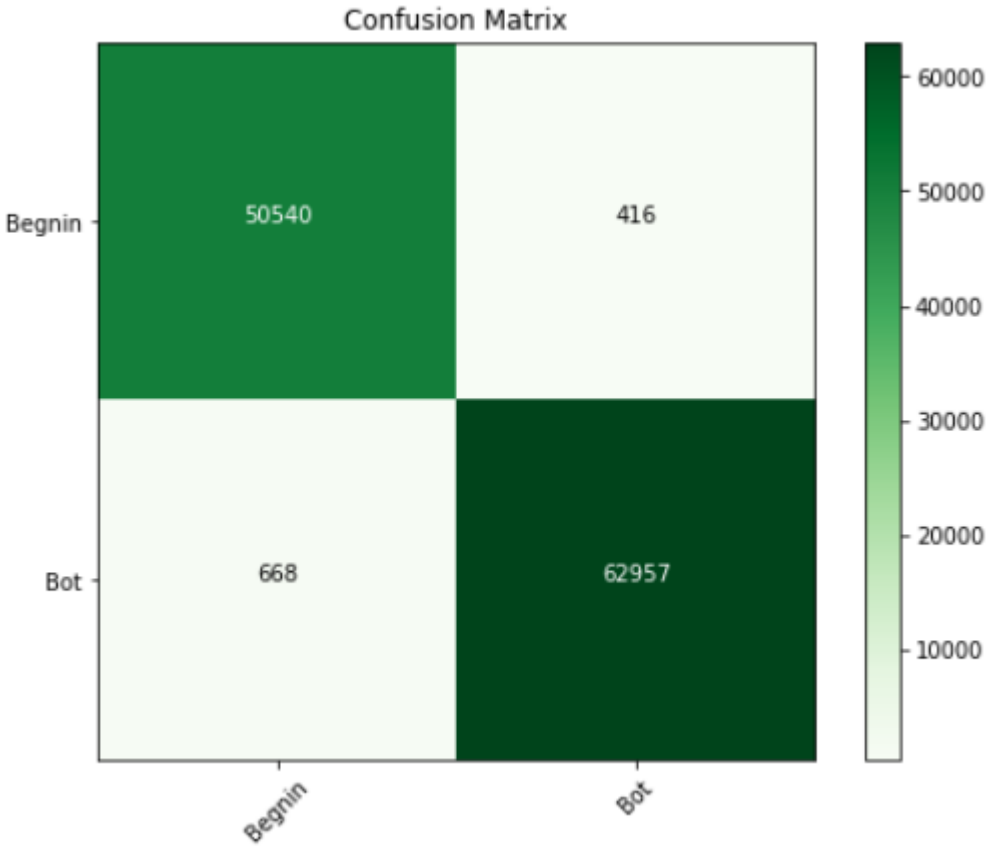

Figure 5.8 Confusion matrix of our model.

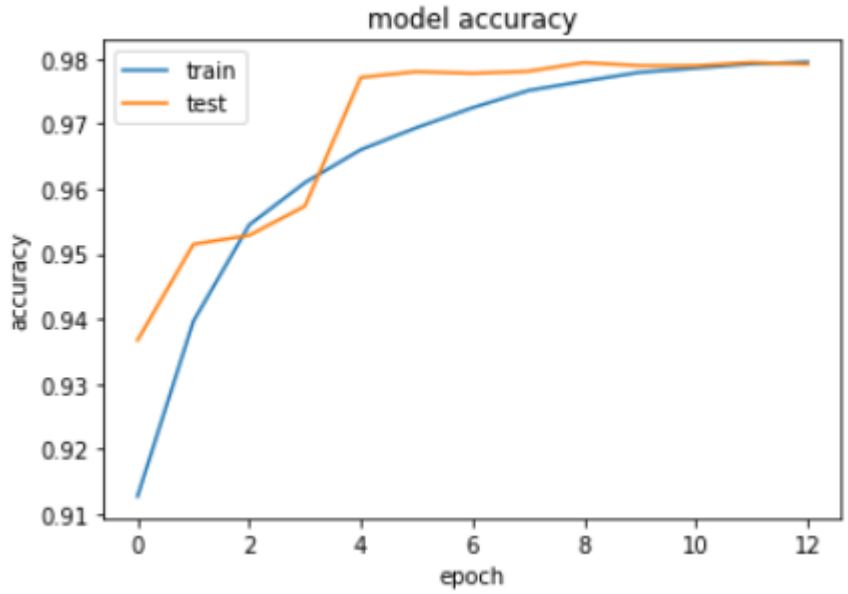

Figure 5.9 The values of Accuracy vs the number of epochs.

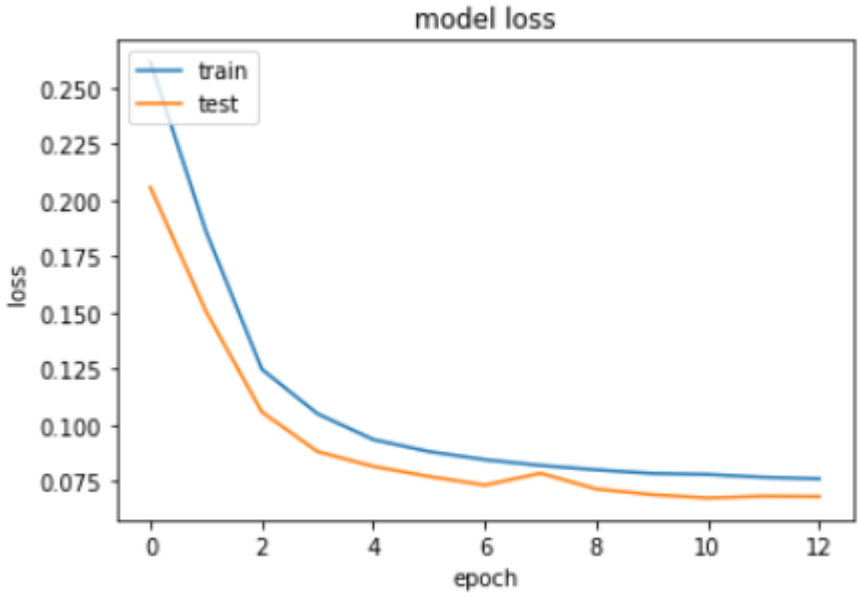

Figure 5.10 The values of Loss vs the number of epochs.

We trained our model for 10 epochs where the batch size is 64. The figures (Figure 5. 9, Figure 5. 10) show that our model achieved the best results in terms of accuracy (99.73%) and loss (0.0113).

4.8 Comparative analysis

To prove the efficiency of our model in the detection of botnets, we compared our results with the results of related works methods mentioned in chapter 3.

As shown Table 5. 1, Our proposed model outperforms by reaching 99.74% accuracy. In addition, the comparison results showed that the proposed model achieved better botnet detection accuracy

Datasets	**Reference**	**Precision %**	**Recall %**	**F-measure %**	**Accuracy %**	**FPR %**
Different datasets	[43]	97.8	96.1	96.9	99.14	4.81
Generated dataset (Eucalyptus cloud)	[44]	X	X	X	96.8	X
Different datasets	[45]	X	X	90.80	90.80	90.30
CTU-13 dataset	[46]	95.9	95	X	95.5	X
ISCX 2012 dataset	[47]	98	100	X	99	X
CICIDS2017	Our model	X	X	X	**99.73**	**0.0113**

Table 5.1 Comparison of our model with related works.

5. Conclusion

In this chapter, we have presented a deep learning approach for botnet detection based on Convolutional Neural Networks (CNN). Our proposed botnet detection system is implemented as a 1D-CNN-based model using some libraries in Google Colab such as Keras, NumPy, etc. It is tested on the CICIDS2017 botnet dataset, which includes 191033 benign traffic and 1966 Botnet traffic.

In the last section, before concluding the chapter, we put our approach in a comparative study with the different related works in the literature presented in the previous chapter. This comparison proved the superiority of our model (1D-CNN).

Conclusion and Outlook

Conclusion

Many years ago in order to detect and address malicious botnet, many methods and computer applications were developed.

Our objective of this research is to find a solution to improve the results of detecting botnet and we have proposed a system based on deep learning using GoogleColab.

Our proposed approach is a deep learning approach for botnet detection based on Convolutional Neural Networks (CNN). We implemented our proposed botnet detection system as a 1D-CNN-based model using the sequential model of the library Keras. We tested it on the CICIDS2017 botnet dataset.

Outlook

The research conducted in this dissertation represents a link in a chain of complementary work that can be done to improve the process of botnet detection in terms of accuracy and loss. Several perspectives and extensions emerge during this work, the most important are the following:

- Test the proposed approach on other datasets such as the CTU-13 dataset and ISCX 2012 dataset.
- Our proposed approach is limited by using the CNN model. The adoption of other techniques such as RNN, Transfer Learning may improve the prediction results;

A. Bibliographic References

[1] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Comput. Surv. vol. 39, no. 1, pp. 1–42, 2007, doi: 10.1145/1216370.1216373.

[2] G. N. Nayak and S. G. Samaddar, "Different flavours of Man-In-The-Middle attack, consequences and feasible solutions," Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010, vol. 5, pp. 491–495, 2010, doi: 10.1109/ICCSIT.2010.5563900.

[3] B. Zhang, T. Zhang, and Z. Yu, "DDoSdetection and preventionbased on artificial intelligence techniques," 2017 3rd IEEE Int. Conf. Comput. Commun. ICCC 2017, vol. 2018-Janua, pp. 1276–1280, 2018, doi: 10.1109/CompComm.2017.8322748.

[4] X. Chen, Distributed denial of service attack and defense, vol. 3. 2010.

[5] X. Li, S. Li, J. Hao, Z. Feng, and B. An, "Optimal personalized defense strategy against man-in-the-middle attack," 31st AAAI Conf. Artif. Intell. AAAI 2017, pp. 593–599, 2017.

[6] O. Eigner, P. Kreimel, and P. Tavolato, "Detection of man-in-the-middle attacks on industrial control networks," Proc. - 2016 Int. Conf. Softw. Secur. Assur. ICSSA 2016, pp. 64–69, 2017, doi: 10.1109/ICSSA.2016.19.

[7] M. Denis, C. Zena, and T. Hayajneh, "Penetration testing: Concepts, attack methods, and defense strategies," 2016 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2016, 2016, doi: 10.1109/LISAT.2016.7494156

[8] D. Steinmetzer, Y. Yuan, and M. Hollick, "Beam-stealing: Intercepting the sector sweep to launch man-in-the-middle aacks on wireless ieee 802.11ad networks," WiSec 2018 - Proc. 11th ACM Conf. Secur. Priv. Wirel. Mob. Networks, pp. 12–22, 2018, doi: 10.1145/3212480.3212499.

[10] J. Jose, T. T. Tomy, V. Karunakaran, V. Anjali Krishna, A. Varkey, and C. A. Nisha, "Securing passwords from dictionary attack with character-tree," Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016, pp. 2301–2307, 2016, doi: 10.1109/WiSPNET.2016.7566553.

[11] A.Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," Neural Comput. Appl., vol. 31, no. 11, pp. 8021–8033, 2019, doi: 10.1007/s00521-018-3595-x.

[12] A. K. Kyaw, F. Sioquim, and J. Joseph, "Dictionary attack on Wordpress:

Security and forensic analysis," 2015 2nd Int. Conf. Inf. Secur. Cyber Forensics, InfoSec 2015, pp. 158–164, 2016, doi: 10.1109/InfoSec.2015.7435522.

[13] A. Sharma, V. Ojha, R. C. Belwal, and G. Agarwal, "Password based authentication: Philosophical survey," Proc. - 2010 IEEE Int. Conf. Intell. Comput. Intell. Syst. ICIS 2010, vol. 3, pp. 619–622, 2010, doi: 10.1109/ICICISYS.2010.5658405.

[14] A. K. Kyaw, F. Sioquim, and J. Joseph, "Dictionary attack on Wordpress: Security and forensic analysis," 2015 2nd Int. Conf. Inf. Secur. Cyber Forensics, InfoSec 2015, pp. 158–164, 2016, doi: 10.1109/InfoSec.2015.7435522.

[15] W. G. J. Halfond, J. Viegas, and A. Orso, "A Classification of SQL Injection Attacks and Countermeasures," Prev. Sql Code Inject. By Comb. Static Runtime Anal., p. 53, 2008

[16] V. K. Gudipati, T. Venna, S. Subburaj, and O. Abuzaghleh, "Advanced automated SQL injection attacks and defensive mechanisms," 2016 Annu. Connect. Conf. Ind. Electron. Technol. Autom. CT-IETA 2016, 2017, doi: 10.1109/CT-IETA.2016.7868248.

[17] K. Natarajan and S. Subramani, "Generation of Sql-injection Free Secure Algorithm to Detect and Prevent Sql-Injection Attacks," Procedia Technol., vol. 4, pp. 790–796, 2012, doi: 10.1016/j.protcy.2012.05.129

[19] M. Eskandari and S. Hashemi, "A graph mining approach for detecting unknown malwares," J. Vis. Lang. Comput., vol. 23, no. 3, pp. 154–162, 2012, doi: 10.1016/j.jvlc.2012.02.002.

[20] I. A.Saeed, A. Selamat, and A. M. A. Abuagoub, "A Survey on Malware and Malware Detection Systems," Int. J. Comput. Appl., vol. 67, no. 16, pp. 25–31, 2013, doi: 10.5120/11480-7108.

[21] N. Idika and A. P. Mathur, "A Survey of Malware Detection Techniques," SERC Tech. Reports, 2007, [Online]. Available: http://www.serc.net/report/tr286.pdf.

[22] JunaidAhsenali Chaudhry, Shafique Ahmad Chaudhry, Robert G. Rittenhouse. "Phishing Attacks and Defenses, » V ol. 10, No. 1 (2016), pp.247-256

[23] A. Tewari, A. K. Jain, and B. B. Gupta, "Recent survey of various defense mechanisms against phishing attacks," J. Inf. Priv. Secur., vol. 12, no. 1, pp. 3–13, 2016, doi: 10.1080/15536548.2016.1139423.

[24] J. Van Roosmalen, H. Vranken, and M. Van Eekelen, "Applying deep learning on packet flows for botnet detection," Proc. ACM Symp. Appl. Comput., pp. 1629–1636, 2018, doi: 10.1145/3167132.3167306.

[25] R. A. Rodriguez-Gomez, G. Macia-Fernandez, and P. Garcia-Teodoro, "Survey and taxonomy of botnet research through life-cycle," ACM Comput. Surv., vol. 45, no. 4, 2013, doi: 10.1145/2501654.2501659.

[26] KhloodShinan, Khalid Alsubhi, Ahmed Alzahran, and Muhammad Usman Ashraf. "Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review",Symmetry 2021, 13, 866, doi: 10.3390/sym13050866.

[28] P. Wainwright and H. Kettani, "An analysis of botnet models," ACM Int. Conf.

Proceeding Ser., pp. 116–121, 2019, doi: 10.1145/3314545.3314562.

[29] M. Singh, M. Singh, and S. Kaur, "Issues and challenges in DNS based botnet detection: A survey," Comput. Secur., vol. 86, pp. 28–52, 2019, doi: 10.1016/j.cose.2019.05.019.

[30] W. C. Shi and H. M. Sun, "DeepBot: a time-based botnet detection with deep learning," Soft Comput., vol. 24, no. 21, pp. 16605–16616, 2020, doi: 10.1007/s00500-020-04963-z.

[32] Memoire

[34] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, 2013, doi: 10.1016/j.jnca.2012.05.003.

[35] N. Kaur and M. Singh, "Botnet and botnet detection techniques in cyber realm," Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016, 2016, doi : 10.1109/inventive.2016.7830080.

[36] Issam El Naqa and Martin J. Murphy "Machine Learning in Radiation Oncology" Theory and Applications, pp. 3-11, DOI: 10.1007/978-3-319-18305-3.

[38] TawehBeysolow II, "Introduction to Deep Learning Using R", pp.1-10. DOI: I 10.1007/978-1-4842-2734-3.

[39] M. Z. Alom et al., "A state-of-the-art survey on deep learning theory and architectures, Electron" vol. 8, no. 3, pp. 1–66, 2019, doi: 10.3390/electronics8030292.

[40] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019- 0192-5.

[41] Hamouda Djallel « Un système de détection d'intrusion pour la cybersécurité », Mémoire de Fin d'études Master, universityGuelma

[42] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," Inf. Fusion, vol. 42, pp. 146–157, 2018, doi: 10.1016/j.inffus.2017.10.006.

[43] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," Comput. Electr. Eng., vol. 50, pp. 91–101, 2016, doi: 10.1016/j.compeleceng.2016.01.012.

[44] K. V. Pradeepthi and A. Kannan, "Detection of Botnet traffic by using Neuro-fuzzy based Intrusion Detection," 2018 10th Int. Conf. Adv. Comput. ICoAC 2018, pp. 118–123, 2018, doi: 10.1109/ICoAC44903.2018.8939109.

[45] X. D. Hoang and Q. C. Nguyen, "Botnet detection based on machine learning techniques using DNS query data," Futur. Internet, vol. 10, no. 5, pp. 1–11, 2018, doi: 10.3390/FI10050043.

[46] E. S. C. Vilaca, T. P. B. Vieira, R. T. De Sousa, and J. P. C. L. Da Costa, "Botnet traffic detection using RPCA and mahalanobis distance," WCNPS 2019 - Work. Commun. Networks Power Syst., no. Wcnps, 2019, doi: 10.1109/WCNPS.2019.8896228.

[47] S. Hosseini, A. E. Nezhad, and H. Seilani, "Botnet detection using negative selection algorithm, convolution neural network and classification methods," Evol. Syst., pp. 1–15, 2021, doi: 10.1007/s12530-020-09362-1.

[51] A. F. Jabbar and I. J. Mohammed, "Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CICIDS2017 Dataset," IOP Conf. Ser. Mater. Sci. Eng., vol. 928, no. 3, 2020, doi: 10.1088/1757-899X/928/3/032027.

[52] S. Ray, K. Alshouiliy, and D. P. Agrawal, "Dimensionality reduction for human activity recognition using googlecolab," Information (Switzerland), 2021.

B. Web references (Technical)

[9] A Complete Guide to Man in The Middle Attack (MitM), (accessed June 03, 2022), https://wallstreetinv.com/cyber-security/man-in-the-middle-attack-mitm/

[18] Rahul Kadapalla "SQL Injection", (accessed June 03, 2022), https://medium.com/@rahulkadapalla/sql-injection-b6dc78df2b3a

[27] Intrusion Detection System (IDS) (accessed may 19,2022), https://www.geeksforgeeks.org/intrusion-detection-system-ids/

[31] What is Cyber Security ? (accessed May 19, 2022), https://www.kaspersky.com/resourcecenter/definitions/what-is-cyber-security

[33] S. M. Othman, F. Mutaher Ba-Alwi, N. T. Alsohybe, and A. T. Zahary, "Survey on Intrusion Detection System Types," Int. J. Cyber-Security Digit. Forensics, vol. 7, no. 4, pp. 444–462, 2018, [Online]. Available: https://www.researchgate.net/publication/329363322.

[37] Machine Learning: A Review of Learning Types, (accessed May 22, 2022), https://www.researchgate.net/publication/342890321

[48] J. Brownlee, "When to Use MLP, CNN, and RNN Neural Networks," 2019, (accessed Jun. 16, 2022)https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural networks/

[49] "Why Convolutional Neural Networks Are The Go-To Models In Deep Learning," 2018,(accessed Jun. 16, 2022)https://analyticsindiamag.com/why-convolutional-neural-networks-are-the-go-to-models-in-deep-learning/

[50] L. Yang and A. Shami, "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice,"

2020.https://www.researchgate.net/publication/343390531(accessed Jun. 19, 2022)

Résumé

Les botnets constituent une menace primaire pour la sécurité d'Internet, un botnet est un groupe d'ordinateurs ou de dispositifs sous le contrôle d'un attaquant, utilisés pour mener des activités malveillantes contre une victime ciblée.

Dans ce travail, nous présentons une approche d'apprentissage en profondeur pour la détection de botnets basée sur les réseaux de neurones convolutifs (CNN). Notre système de détection de botnet proposé est implémenté en tant que modèle basé sur 1D-CNN qui est testé sur l'ensemble de données de botnet CICIDS2017, qui comprend 191033 trafic bénin et 1966 botnet trafic.

Selon nos résultats, nous pouvons détecter les robots avec une grande précision et une faible perte et c'est par le modèle que nous avons proposé.

Mots clés : Apprentissage profond, Bonet, CNN, IDS.

Abstract

Botnets constitute a primary threat to Internet security, a botnet is a group of computers or devices under the control of an attacker, used to carry out malicious activities against a targeted victim.

In this work, we present a deep learning approach for botnet detection based on Convolutional Neural Networks (CNN). Our proposed botnet detection system is implemented as a 1D-CNN-based model that is tested on the CICIDS2017 botnet dataset, which includes 191033 benign traffic and 1966 Botnet traffic.

According to our findings, we can detect robots with high accuracy and low loss and this is by the model we have proposed.

Keywords: Deep Learning, BotNet, IDS, CNN.

ملخّص

تشكل شبكات الروبوت تهديدًا رئيسيًا لأمن الإنترنت، والشبكة الروبوتية هي مجموعة من أجهزة الكمبيوتر أو الأجهزة تحت سيطرة مهاجم، تستخدم للقيام بأنشطة ضارة ضد ضحية مستهدفة.

في هذا العمل، نقدم نهجًا متعمقًا لاكتشاف شبكة البوت بناءً على الشبكات العصبية التلافيفية (NNCs). يتم تنفيذ نظام الكشف عن شبكة البوت الخاص بنا كنموذج D-CNN-based1 يتم اختباره على مجموعة بيانات بوت نت CICIDS2017، والتي تشمل حركة المرور الحميدة عام 191033 وحركة مرور بوت نت عام 1966.

وفقا للنتائج التي توصلنا اليها فانه يمكننا اكتشاف الروبوتات مع تحقيق دقة عالية وخسارة منخفضة وهذا بواسطة النموذج الذي اقترحناه.

الكلماتالمفتاحية:التعلمالعميق،Bonet،CNN.، IDS.