



REPUBLIC OF TURKEY
ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering Department

**INTRUSION DETECTION IN IOT NETWORKS
USING FEATURE SELECTION AND SVM
CLASSIFICATION**

Maryam Ali Hussein AL-BALHAWI

Master's Thesis

Supervisor

Prof. Dr. Galip CANSEVER

Istanbul, 2022

INTRUSION DETECTION IN IOT NETWORKS USING FEATURE SELECTION AND SVM CLASSIFICATION

Maryam Ali Hussein AL-BALHAWI

Electrical and Computer Engineering Department

Master Thesis

ALTINBAŞ UNIVERSITY

Istanbul, 2022

The thesis titled INTRUSION DETECTION IN IOT NETWORKS USING FEATURE SELECTION AND SVM CLASSIFICATION prepared and presented by MARYAM ALI HUSSEIN AL-BALHAWI and submitted on 08/08/2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

Prof. Dr. Galip CANSEVER

Supervisor

Thesis Defense Committee Members:

Prof. Dr .Galip CANSEVER

Faculty of Engineering and
Architecture,

Altinbas University

Asst. Prof. Dr. Sefer KURNAZ

Faculty of Engineering and
Architecture,

Altinbas University

Asst. Prof. Dr. Yuksel BAL

Faculty of Engineering
Computer Engineering
Department,

Top Kapi 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/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.



Maryam Ali Hussein AL-BALHAWI

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through the whole research work as well as my family for always assisting me in my hard time.



PREFACE

First and foremost, I would like to thank my supervisor Prof. Dr. Galip CANSEVER for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Prof. Dr. Galip CANSEVER a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work.



ABSTRACT

INTRUSION DETECTION IN IOT NETWORKS USING FEATURE SELECTION AND SVM CLASSIFICATION

AL-BALHAWI, Maryam Ali Hussein,

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

Supervisor: Prof. Dr. Galip CANSEVER

Date: 08/2022

Pages: (47)

The steady growth in the number of devices connected to the Internet has attracted cyber criminals looking for vulnerabilities in computer networks and systems. The objective of this paper is to develop a model to identify DDoS, Infiltration, Web, and Brute force attacks on computer networks, using Machine Learning (ML) techniques, increasing the accuracy, sensitivity, precision and measurement values. -F in relation to existing work.

Keywords: Malware, IOT, ML, DDoS

TABLE OF CONTENTS

	<u>Pages</u>
ABSTRACT.....	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xi
1. INTRODUCTION	1
1.1 BACKGROUND	1
1.2 THESIS OBJECTIVES	3
1.3 PROBLEM STATEMENT.....	3
1.4 THESIS ORGANIZATION	5
2. LITERATURE REVIEW.....	6
3. MATERIALS AND METHODS.....	11
3.1 COVID-19	11
3.2 DEEP LEARNING	15
3.2.1 CNN.....	17
3.2.2 Feature Extraction.....	20
3.3 SVM CLASSIFICATION	22
4. PROPOSED METHOD	25
4.1 SDN SECURITY ISSUES	26
4.2 INTRUSION DETECTION IN SDN	28
4.3 IMPLEMENTATION OF IDS- FOR SDN.....	28
4.4 MACHINE LEARNING DATA CLASSIFICATION.....	35
5. CONCLUSION	38
REFERENCES.....	39

LIST OF TABLES

	<u>Pages</u>
Table 4. 1: the KDD dataset structure.....	26



LIST OF FIGURES

	<u>Pages</u>
Figure 1. 1: Network IDS architecture.....	3
Figure 1. 2: DDoS attack implementation	4
Figure 3. 1: Covid-19 structure	12
Figure 3. 2: Covid-19 transmission.....	15
Figure 3. 3: Deep learning and Machine learning.....	17
Figure 3. 4: CNN chest x-ray architecture	19
Figure 3. 5: SVM X-ray classification.....	23
Figure 4. 1: DDOS attack on a victim.....	28
Figure 4. 2: Proposed SDN Intrusion System	29
Figure 4. 3: Attack model for the IDS	30
Figure 4. 4: Network Dynamics for the used data	32
Figure 4. 5: Data rates in different devices	33
Figure 4. 6: Detected attackers using the CNN-SVM workflow	34
Figure 4. 7: CNN-SVM data classification.....	36

ABBREVIATIONS

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

1. INTRODUCTION

1.1 BACKGROUND

As the number of devices linked to the Internet continues to grow, cyber thieves are increasingly focusing their efforts on computer networks and systems in search of potential weaknesses. Cisco anticipates that by 2023, the number of Internet-connected devices will be three times that of the world's population, which the United Nations projects would reach more than 8 billion people (2019). According to Cisco's data, this equates to an average of 3.6 devices per person (2019). With the proliferation of Internet-connected devices, security becomes an increasingly critical issue to monitor. To combat this, protective mechanisms must be implemented. Intrusion Detection Systems (IDS), sometimes known as intrusion detection systems (IDS), are computer network security systems that detect and prevent malicious software and other types of intrusions. As a general rule, this kind of technology is limited to dealing with previously recognized detections. To avoid false positives, intrusion detection systems (IDS) signatures ignore zero-day assaults, which use an undiscovered vulnerability to launch a Distributed Denial-of-Service (DDoS) attack on a business. This topic involves investigating if Machine Learning (AM) algorithms can be used to recognize and identify anomalous traffic patterns. The study in [3] focuses on the application of machine learning techniques by using AM techniques like as Decision Trees (DT), Naive Bayes, Random Forests (RF), and Support Vector Machines (SVM). According to the Balakrishnan, K., and A. (2014) paper, when utilizing the CSE-CIC-IDS2018 dataset, the accuracy of these algorithms in identifying Denial-of-Service attacks is often less than 80%. The use of Deep Learning (DL) algorithms to intrusion detection systems (IDSs) has gained in popularity over time. [3] demonstrates how well-suited DL is for developing a high-accuracy classification model based on the Recurrent Neural Network (RNN) paradigm. DL outperformed established AM classification techniques such as DT, RF, and NB in both binary and multiclass classification, suggesting that DL is better in both binary and multiclass classification. Manimurugan and colleagues (2020) developed a Deep Learning (DL) model based on the Deep Belief Network (DBN) and applied it to the issue of classifying attacks (DBN). They used the publicly accessible dataset CIC-IDS2017 to do the tests, which yielded optimistic findings such as 97.71 percent brute force, 96.67 percent DoS/DDoS, 96.37 percent infiltration, and 98.37 percent web. Another way to phrase it is that researchers are always working to improve the accuracy of AM and DL models,

particularly when it comes to information security, such as intrusion detection systems. The purpose of this research is to offer an attack classification model for DDoS, infiltration, web, and brute force assaults based on a DL algorithm with the objective of enhancing detection accuracy for these attacks.

Current datasets must be updated on a regular basis [5] to ensure that AM models are not negatively impacted by developing malware and attack techniques during training. As the number of invasive actions increases, various studies, such as [6] and [7], have been conducted to build a complete and trustworthy dataset for the aim of attack detection. Finding an adequate dataset for detections is very challenging, since the bulk of available databases characterizing prospective attacks are out of date, posing a considerable obstacle. The present bases suffer from a lack of traffic and attack variety. Utilizing a fresh dataset to optimize an IDS's performance, as advised by [8], may be advantageous. According to these studies, a comprehensive dataset on computer network assaults must adhere to certain criteria in order to be termed complete. According to [9], it is necessary to understand the various sorts of assaults, as well as the attackers' protocols and the labels attributed to them. A new dataset will be created as part of this project in order to train and evaluate the proposed deep learning model for attack detection, as well as to compare it to existing attack detection models. This dataset, which will be constructed entirely of genuine network records, will be used to analyze DDoS, infiltration, web, and brute force assaults, among others.

In order to obtain a current and objective dataset, pre-processing techniques are applied. Such as the Principal Component Analysis (PCA), which allows the selection of more representative attributes in order to characterize the types of attacks already mentioned and form new models in short periods of time. Existing databases, such as CIC-IDS2017, work with 86 attributes, and this causes a higher latency in the formation of a machine learning model to characterize possible attacks. [2]:

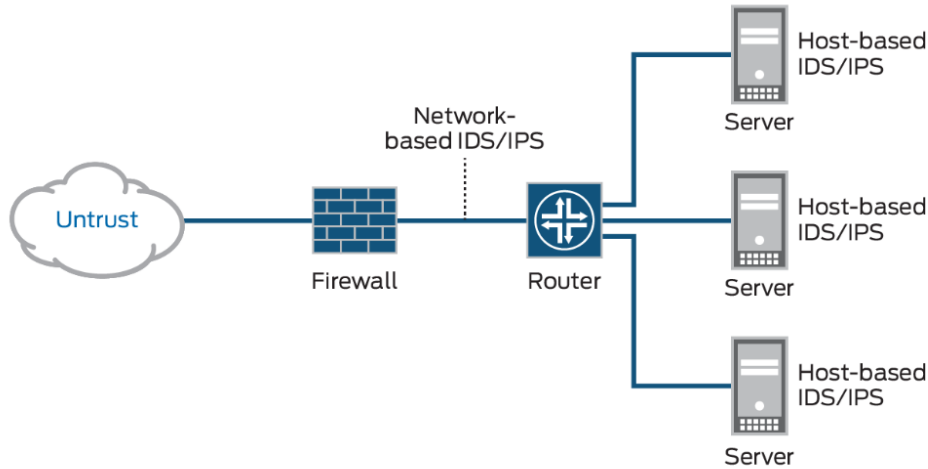


Figure 1. 1: Network IDS architecture

1.2 THESIS OBJECTIVES

The objective of this work is to develop a model to identify DDoS, Infiltration, Web and Brute force attacks on computer networks, using Machine Learning (ML) techniques, increasing the accuracy, sensitivity, precision and measurement values. -F in relation to existing work.

- Build a more comprehensive and current dataset, with DDoS, Infiltration, Web and Brute force attacks in a real environment;
- Apply feature selection techniques in order to reduce the dimensionality of the set, increasing the speed of attack identification;

1.3 PROBLEM STATEMENT

Existing attack detection techniques are not prepared for today's high volume of traffic. Therefore, more and more solutions are needed that are capable of analyzing traffic automatically, avoiding bottlenecks in the network that cause slowness and false positives in the analyses. Seeking to accompany this growth, ways of using AM in these analyses are being studied, so that the attack characterization models have greater accuracy. One of the machine learning techniques that have obtained results superior to existing techniques are those of DL

Therefore, it is intended to study the performance of DL in detecting the types of DDoS, Infiltration, Web and Brute force attacks, and to create a model that obtains accuracy, sensitivity, precision and F-Measure superior to currently proposed algorithms, as in accordance with [9] DDoS, Infiltration, Web and Brute force attacks are the most common types of attacks. Classification of computer network traffic through DL is a recent technique. Some studies, such as in [5] and [6] developed models in DL, but there is still a need for more studies, especially with more recent datasets and more sophisticated algorithms, creating more efficient models for this task. In the work of [12] and Yang et al. (2019a), the authors show that the application of DL algorithms in network traffic analysis, specifically in intrusion detection, has been done using static bases, such as KDDCup99 [11]. In this work, the DL algorithms will be applied to a dataset containing traffic from recent attacks, obtained during this research. In this context, some DL algorithms will be studied, and with the adjustment of hyperparameters of the proposed model, it is intended to increase the accuracy in the detection of DDoS, Infiltration, Web and Brute force attacks. For validation, the algorithms will be applied in own and public datasets. The use of its own dataset is necessary to reproduce a small and medium-sized environment, such as in Brazilian universities. Such dataset was built because these types of attacks studied are more common in this type of environment. [2]:

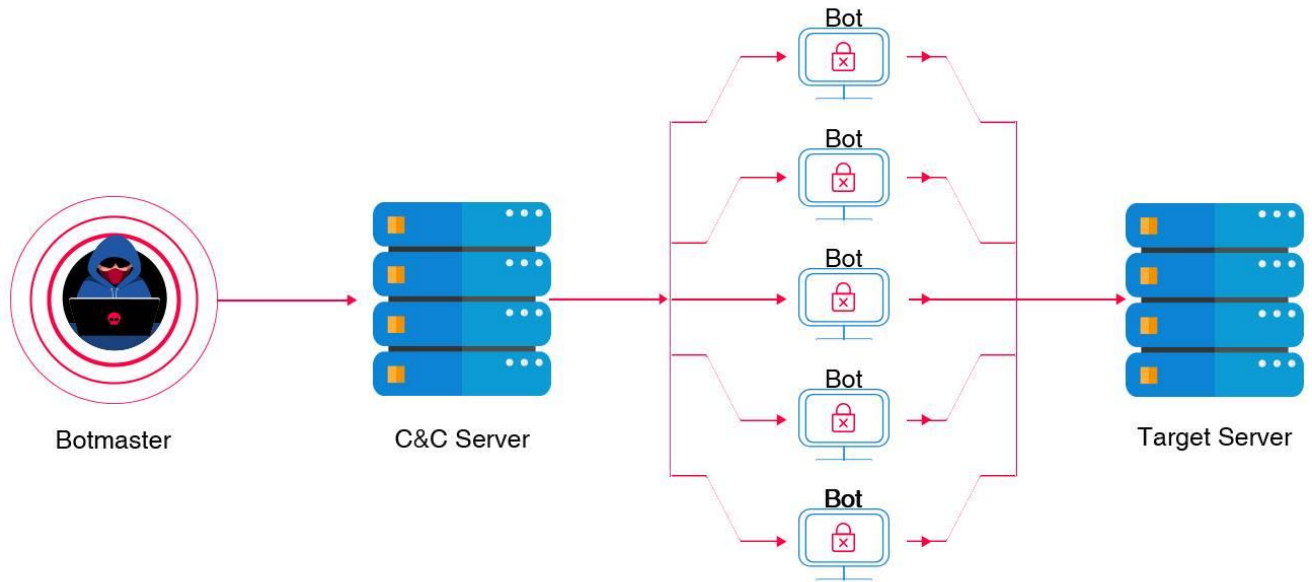


Figure 1. 2: DDoS attack implementation

1.4 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. Section 4 is where we explain our model in details and implementation of that model in details, Section 5 is where we simulate our model and record the results obtained. Section 6 is where we conclude our work and put a future scope into perspective.



2. LITERATURE REVIEW

Despite recent advances in IDS research, current IDS systems are unsuitable for general usage in Internet of Things applications. They might be improved. According to the researchers, intruder detection technologies have been developed by researchers who have shown the capacity to adapt to and overcome the limitations of the Internet of Things. Due to the growing popularity of Internet of Things intrusion detection, we performed a research to get a better understanding of how IDS training and detection methodologies are being utilized. Numerous studies on intrusion detection systems (IDSs) in the context of the Internet of Things (IoT) have been conducted [16, 28, 91]. While an overview of an IDS's effectiveness in an IoT system has been provided, no research has been undertaken to establish if it is really advantageous. Prior to the Internet of Things (IoT), there was another kind of restricted network environment known as wireless sensor networks (WSN). Alternatively, WSNs are a subset of Internet of Things systems that share many of the same security concepts and constraints as IoT systems in general. Butun et al. (2013) [16] provide an early review of wireless sensor network intrusion detection systems (IDSs). MANETS, or mobile ad hoc networks, are also discussed in the next section, which discusses wireless sensor networks. The study's results address various aspects of some of the most widely used methodologies for detecting attacks on these systems, including Due to the fact that WSNs are often homogeneous networks, statistical methodologies may be used to discover attacks that would go unnoticed in many IoT systems due to the networks' homogeneity. The authors analyze the effectiveness of different approaches by examining the forwarding percentage of packets sent by a node. The authors discuss the advantages and disadvantages of each methodology for detecting WSN attacks, as well as the method's limits. In contrast to the Internet of Things (IoT), these solutions do not suffer from the same interoperability challenges that have plagued the IoT owing to the use of several protocols and networks. There is no systematic comparison of techniques when it comes to the use of methodologies in a specific context. Despite the fact that it is not as exhaustive as it might be, this is an excellent introduction of MANETS and WSN methods and their operation. Zarpelo et al. (2017) [91] discovered that WSNs have been neglected. The study of classified intrusion detection systems (IDS) for the Internet of Things (IoT) should begin immediately (IoT). To do this, they seek for emerging trends, unsolved issues, and new study topics to analyze and explore. It is determined, among other things, by the detection mechanism, the IDS deployment

strategy, the security threat, and the validation process. The authors conduct a methodological and attack-specific analysis of the intrusion detection system (IDS). A literature review highlights these three points in more detail for each article in the survey. Additionally, the authors examine the validity of these strategies, which they feel are successful. According to specialists, no one strategy is applied in this case. Comparing and contrasting the different methods was difficult, and some of them provided no benefit at all. Elrawy et al. (2018) [28] published the most recent work on intrusion detection systems for the Internet of Things. This research takes a comprehensive look at the Internet of Things' components, including its architecture and design, as well as potential hazards and application scenarios. Each article describes in depth the approaches used, as well as their advantages and disadvantages. After reading the overview, which provides context for the concepts, preparing for the IDSs is uncomplicated. A table summarizes the approach and deployment strategy for each study. As a result, the authors note that direct comparisons between this data and those from other studies are problematic. Numerous questions are duplicated throughout all polls. Due to the constant evolution of network architecture, it is vital to create new research and survey methodologies. There has been no research comparing the effectiveness of different IDS strategies since none of these techniques have been assessed. While we collected over 50 IDS articles for our more complete examination, each of these studies chose over 20 different ones, making inter-study comparisons challenging. As a result, identifying remaining obstacles, selecting the best suited strategy for a certain circumstance, and determining whether the content offered is genuine or not becomes tough. As a result, standards-based ways to compare multiple IDSs are required. Protecting the security of our homes, companies, and even healthcare is becoming more critical as the Internet of Things (IoT) takes over more and more aspects of our everyday life. Due to the significant differences between these networks and traditional internet systems, additional security measures are required. To build safe solutions, it's critical to understand the Internet of Things' (IoT) influence on their design. Due to resource constraints on the Internet of Things, such as battery life, bandwidth, processor, and memory capacity, it is difficult to implement computing-intensive security methods. Despite this, it is unclear how these variables affect an IDS's ability to be deployed effectively. The following are some of the most critical points to keep in mind: 1. A number of attacks are inapplicable to the majority of systems, including the following: As a consequence, IDS detection skills must be strengthened. In light of these additional dangers, it is critical to reconsider conventional therapies. Intrusion detection

systems (IDS) agents are often placed on nodes with increased processing capacity in traditional networks. It is vital to include cloud, fog, and edge computing while establishing an IDS strategy for the Internet of Things. In contrast to typical IDS classifications, it is difficult to make general assertions about the first categorization of IoT devices. When it comes to intrusion detection systems, determining whether they are centrally located or distributed is difficult owing to the importance of factors such as fog, clouds, and devices linked to a local area network (LAN). Security and performance may be impacted differently depending on the choice selected. According to Raza et al. (2013) [70], a solution must integrate a strategy for each of these scenarios in order to properly cover the Internet of Things. Demonstrating how the approach circumvents one of the constraints is insufficient (e.g. new protocols). The location of an IDS is highly dependent on the kind of IDS. Additionally, there is a categorization for the Internet of Things (IoT): collaborative intrusion detection systems (CIDS), which are composed of network-based and host-based intrusion detection systems. While these attacks are often focused on protocol weaknesses, they may also have a substantial effect on the target device. Because DoS (distributed denial-of-service) attacks target device and network limitations, Internet of Things (IoT) devices may be particularly vulnerable. Due to their limits, security systems, for example, may be unable to guard against future attacks on these devices. The majority of Internet of Things (IoT) systems use a variety of different protocols, which distinguishes them from conventional infrastructure. As a consequence, new technologies have supplanted more established protocols in the Internet of Things, allowing for the constraints of particular devices. The RPL protocol, which is only used on a few select networks, is an extreme example of this current trend toward protocol specialization. Due to a major redesign of these systems' architecture, network-based intrusion detection methods may no longer function correctly on them. Numerous approaches have been developed to address the unique difficulties raised by these new methods. Because they are built on machine-to-machine (M2M) interactions, unsupervised systems for the Internet of Things are gaining popularity. As a result, these systems are likely to operate with little or no human supervision for the majority of the time. In certain cases, a compromised node may remain undetected, allowing an attacker to conduct attacks from inside the system without being noticed. External firewalls are incapable of preventing these sorts of attacks, necessitating the use of an internal intrusion detection system (IDS) [46, 54, 89]. As a result of the architecture of the systems, new dangers and problems about device authentication are becoming more frequent over time. The

capacity of an unsupervised system to route traffic in real time is a critical attribute. Alternatively, routes are generated as a result of changes in system factors such as signal strength and resource allocation. When packets are routed via a hacked node, an attacker may gain an advantage by interfering with the process and routing them in a less-than-optimal or more-than-optimal fashion. WSN and IoT routing exploits such as Wormhole attack 1 and Sybil attack 2 have gained popularity in recent years owing to their reliance on unsupervised and self-adaptive systems. As noted in Section 7.2, some research publications have been written specifically to identify the types of attacks addressed in this section. The Internet of Things' intrusion detection system is not the only component that suffers as a consequence of infrastructure changes. Despite their flaws, some systems may be more successful in detecting possible threats. Due to the fact that the components in these systems exhibit confined behavior, novel ways are being developed and studied to overcome this limitation. As a result of automaton-based behavior models' constrained behaviors There has been research on automaton-based behavior modeling [32, 63]. As a consequence, automated processes may generate a greater number of actionable IDSs than human methods do. By using a set of simple behaviors [74], these strategies were able to determine the best behavior of a basic system. Additionally, clustering algorithms are becoming increasingly prevalent. As a result, computational intelligence refers not only to a computer's capacity to adapt to changes in its environment and to accept mistakes, but also to a computer system's ability to perform on a par with a real-world human person. [23,93]. [23] While the specific definition of computational intelligence (CI) remains debatable, ideas such as swarm intelligence and artificial immune systems, as well as evolutionary computations, artificial neural networks, and soft computing, are all considered components of CI. These strategies may be beneficial in both supervised and unsupervised learning environments because they enable the autonomous collecting and integration of information. Unlike artificial intelligence, which is often used in many of the ways outlined above, computational intelligence is a unique field. Artificial immune systems, a relatively recent idea in computational intelligence, may be able to aid with IoT intrusion detection due to its self-adaptation and durability. Leandro Nunes and his colleagues built adaptable artificial immune systems. They deal with challenges by using concepts from theoretical immunology and observations of immune activity as inputs and basing their conclusions on these concepts. The most difficult part of the method is matching network data to immunity algorithm inputs; the ability of any system to adapt to changing conditions and repeated assaults is a vital component of its

efficacy. On the other side, it will be unable to recognize the danger for an extended period of time. This strategy may be advantageous for detecting intruders or for identifying changes in a gadget's behavior. According to Arrington et al. (2016), human intruders in a smart home may be identified and recognized using these tactics. Due to the restricted structure of the Internet of Things, it is possible to match attack signatures to 'antigens' and normal behaviors to 'self-components,' according to Liu and colleagues (2011) [53-54].



3. MATERIALS AND METHODS

3.1 COVID-19

There are several coronaviruses that may infect both animals and humans. According to the World Health Organization, the four most prevalent human coronaviruses are 229E, NL63, OC43, and HKU1. Numerous airborne illnesses, including SARS, MERS, and other severe acute respiratory syndromes (SARS), as well as other coronavirus-related infections, have been reported recently, illustrating the broad spectrum of airborne illnesses that may occur (COVID-19). [1] [1] A zoonotic disease induced by a severe case of acute coronavirus syndrome 2. (SARS-CoV-2). The World Health Organization designated a new coronavirus discovered in 2019 as New Coronavirus-Infected Pneumonia (NCIP) in honor of the coronavirus that caused it. (2019-nCoV). On February 11, 2020, the World Health Organization (WHO) officially classified the clinical ailment COVID-19 (Corona Virus Disease-19), after its first report in a tweet that day. The COVID-19 epidemic in Wuhan, Hubei Province, China, which began in December 2019, was caused by a newly discovered coronavirus (SARS-CoV-2). The World Health Organization declared the current epidemic a pandemic in September. Due to the rapid evolution of viral information, readers are advised to stay current on the newest advances (Fig. 1). [3] [3] Coughing and sneezing produce respiratory droplets, which contribute to the virus's rapid transmission from one person to another and to other individuals. Despite the fact that sick persons are regarded to be the most contagious, transmission may occur prior to the onset of symptoms. Symptoms often manifest two to fourteen days after exposure, with an average of five days between exposure and beginning of symptoms. Shortness of breath, sneezing, and a fever are all signs of the common cold. Complications may include pneumonia, sore throat, and acute respiratory distress syndrome. Due to the lack of a viable antiviral medication or vaccine, efforts to alleviate symptoms are now limited to supportive care. Coughing, hand washing with soap, and maintaining a one-meter distance from others are just a few of the recommended preventative measures.

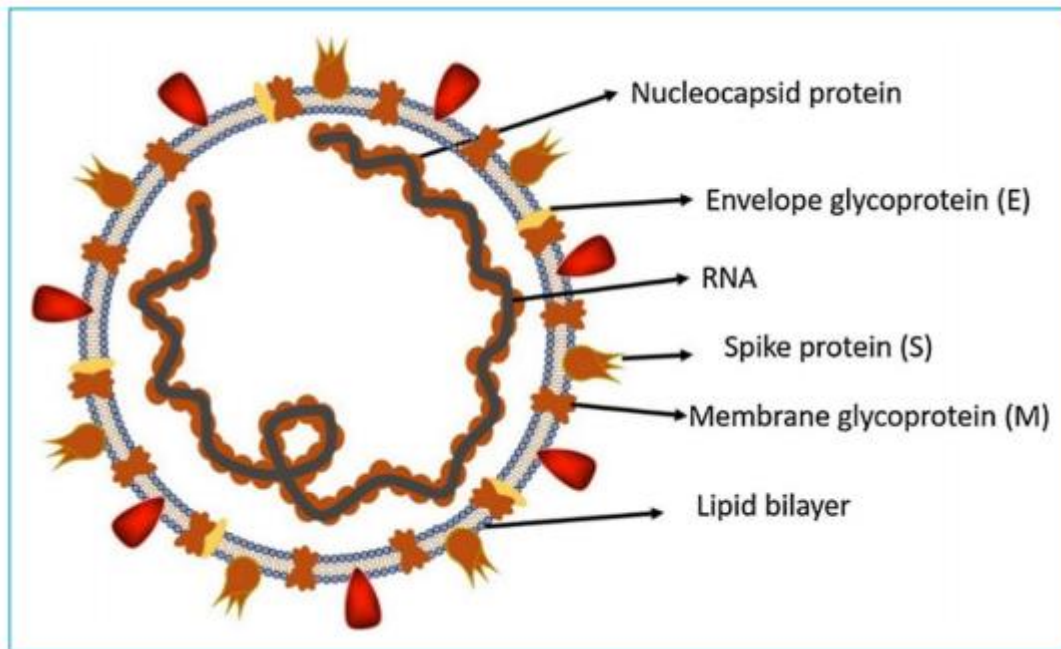


Figure 3. 1: Covid-19 structure

The most often used diagnostic tool is a reverse transcription polymerase chain reaction (rRT-PCR) from a throat or nasopharyngeal swab. Additionally, the infection may be diagnosed based on a combination of symptoms, risks, and a chest CT scan with pneumonia features. [5] Epidemiologics As of April 15, 2020, 210 nations and territories worldwide have reported more than 1,998,111 confirmed cases and 126,604 deaths from COVID-19. According to the medical journal published by the Johns Hopkins University. Despite the fact that the number of verified cases outside of China continues to grow. [2–6] The following graph illustrates the nations that have been most impacted. COVID-19 statistics Fig. The death rate varies by age, health status, and sex. Tedros Adhanom Ghebreyesus, Director General of the World Health Organization, said that around 3.4 percent of Covid-19 instances reported globally had resulted in death. The UK governments' highest assessment of Matt Hancock's health secretary was that the death rate was "2% or possibly less." However, it is dependent on a number of factors, including your health, your age, your gender, and the health system in which you reside. In the first large examination of almost 44,000 cases in China, the elderly had a death rate ten times that of the middle-aged. In 4,500 cases, death rates were lower, with eight fatalities occurring in people under the age of 30. And death was at least five times greater in individuals with diabetes, hypertension, heart disease, or a respiratory ailment. Males suffered much more deaths than females. [7]. COVID-19 Death

rate by age group: Death rate = Number of fatalities divided by the total number of cases = Cause of death if infected with the virus (percent) (%) Numerous studies indicate that death rates increase with age. Children under the age of nine seemed to be mostly unaffected, either with no or mild symptoms, and none died from COVID-19 infection. Individuals above the age of 80 and those who suffer from chronic diseases are especially at risk. Around 14.80 percent of people infected die as a result of those crossings 80. Data are shown in Figure 3. Over the age of 50, the death rate starts to rise. The death rate for people under the age of 50 who are unwell is 0.40 percent, and for those aged 50 to 59 years, it is 1.3 percent. 3.60 percent for those aged 60-69, 8.00 percent for those aged 70-79, and 14.8 percent for those above the age of 80.

Females infected with the coronavirus. Nonetheless, its distribution varies greatly across nations. It may or may not reflect biological differences. Scientists are not certain, but males may be more active in health-damaging behaviors such as drinking and smoking on average than females (See Figure 4.) Demonstrate the difference in the lethality of sexes. [8] COVID-19 Death rate by health status: The Centers for Illness Control and Prevention (CDC) and several other studies are demonstrating an increasing risk of severe disease and death as people age. Adults who are both elderly and have unresolved medical conditions are more susceptible to infection. Over half of those aged 60 or older also have a significant health condition, which increases to over two-thirds of people aged 80 or older. [2, ten] [2, ten] When older people and younger people with substantial diseases such as diabetes, heart disease, or lung disease are infected with coronavirus, the risk of becoming very ill increases. Without pre-existing conditions, the death rate is around 1%. The Centers for Disease Control and Prevention have issued specific advice to members of these groups. [10] Cardiovascular (heart) disease has a death rate of 10.5 percent; diabetes has a death rate of 7.3 percent; chronic respiratory diseases (for example, asthma and chronic obstructive pulmonary disease) has a death rate of 6.3 percent; and cancer has a death rate of 5.6 percent in the summarized statistics. (8,9) COVID-19's genesis and transmission are shown in Figure 5. Tyrrell and Bynoe found the first human coronaviruses in 1965. They noted the possibility of a virus known as B814 slipping through. It has been seen in human embryonic tracheal organs that were recovered by an adult suffering from a common cold-like respiratory complaint. The first cases were reported in December 2019 in the city of Wuhan in the Chinese province of Hubei, and were associated with the Huanan Seafood Market (South China) (South China). [2] The novel coronavirus was discovered in Wuhan, Hunan, South China, where raccoon dogs, bats, snakes,

palm calves, and other animals are trafficked and swiftly spread over 109 countries. The zoonotic origins of SARSCoV-2 are unknown, however sequence-based research indicates that bats are the major reservoir. Recombination of DNAs with RBDs from another Beta CoV was revealed to be a mechanism of cross-species transmission and rapid infection in spike glycoproteins supporting SARS-CoV (CoVZXC21 or CoVZC45). [7] Coronavirus-related virus 19 (COVID-19) is a highly contagious and dangerous virus that is mostly transmitted by air rather than respiratory direct contact. [7, 11, 12] Coronavirus disease 19 (COVID-19) is a virus that is mostly disseminated via contact with affected individuals. A single cough might result in the circulation of up to 3,000 droplets. While some of these droplets may fall on other persons, the majority of these microscopic particles will remain in the atmosphere. Additionally, the virus is spread by anybody who does not thoroughly wash their hands after using the restroom, and it has the ability to infect everything, similar to many respiratory viruses such as influenza. Covid-19 may be spread by intimate contact with minute droplets released by infected individuals' upper respiratory tract secretions [12], such as sneezing, a chronic cold, or coughing. As a result, you should keep a distance of at least one meter (3 feet) from a sick individual. [2] The virus may also be transferred by surface pollution if the virus settles on things and surfaces in the individual's near area and then other individuals come into contact with these objects or surfaces and get COVID-19, which then affects their eyes, nose, and mouth much more. [11-13] The COVID-19 transmission signal is seen in Figure 3.2.

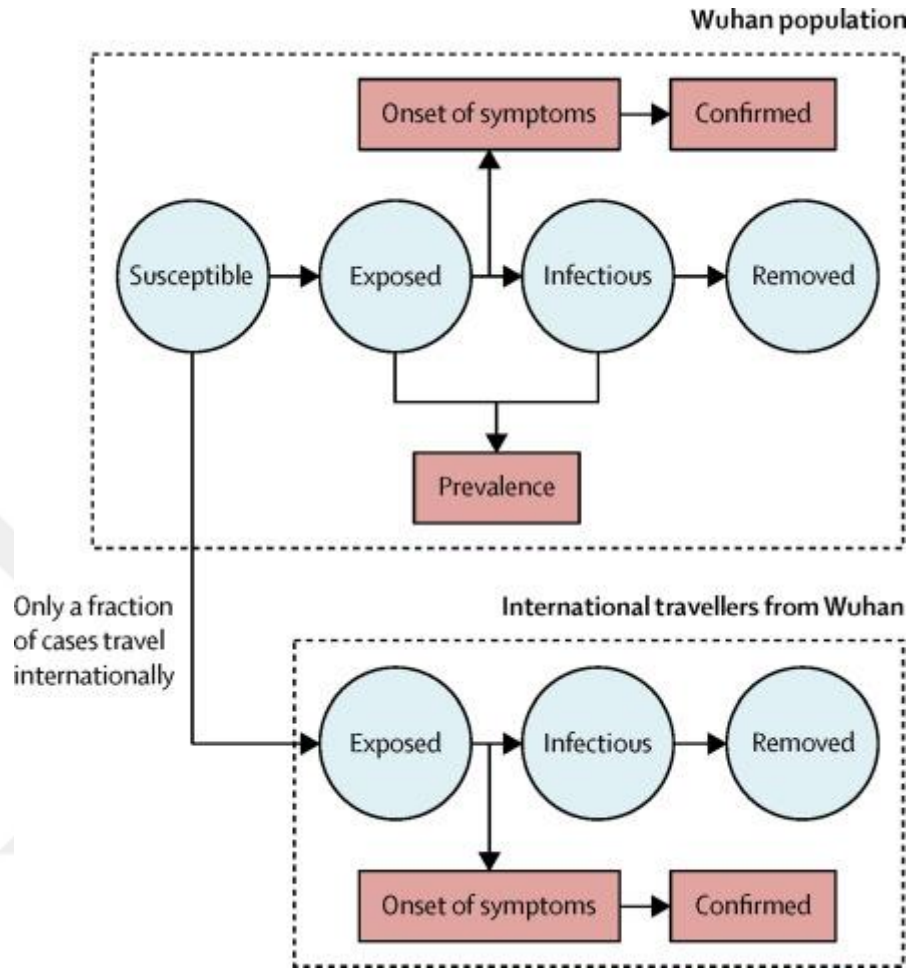


Figure 3. 2: Covid-19 transmission

3.2 DEEP LEARNING

Deep learning is a subset of machine learning that is inspired by the human brain's structure. It is a 2010 subfield in machine learning. The use of deep learning algorithms in the area of medical image processing, as well as in a variety of other fields, has yielded considerable advantages in recent years. Deep learning algorithms have been employed to attempt to extract meaningful information from medical data, with encouraging results. Among other uses, deep learning models have been effectively utilized to classify, segment, and identify lesions in medical data. Image and signal processing is the process of evaluating image and signal data acquired by medical imaging procedures such as MRI, CT, and X-rays using deep learning models. Along with diabetes and brain tumors, skin cancer, and breast cancer, these investigations enable the speedy and precise detection and diagnosis of a number of different ailments [36-41]. Convolutional neural networks

(CNNs) are a kind of deep neural network that are often employed in image recognition applications. Before they can be used, CNN's input images must be identified and transformed to a computer-readable format. Following that, the photos are converted to a matrix format. The system uses the disparities between the pictures, and hence the matrices, to identify which image corresponds to each label. After the training phase is complete, it will be able to identify how these deviations affect the label and forecast future photos based on this knowledge. CNN utilizes three layers to do these tasks successfully and efficiently: a convolutional layer, a pooling layer, and a fully connected layer. The feature extraction method must use both convolutional and pooling layers. On the other hand, classifying is a one-layer procedure. The sections that follow detail each of these steps in detail. CNN is built on top of the convolutional layer. It is responsible for specifying the specifics of the pattern to be employed. This layer applies a filter on the supplied image. It is the feature map that results from the filtering. This layer employs kernels to traverse the pattern, extracting both low-level and high-level information [42-43]. The kernel is a three-dimensional (3x3) or five-dimensional (5x5) matrix that is used to convert the input pattern matrix. By adjusting the stride parameter, you may freely navigate around an input matrix. A convolutional layer's output may be stated in terms of the number of kernels in layer $l-1$ (w_j), the number of kernels in layer $l-1$ (y_a), the number of kernels in the a -th feature map (b_j), and the number of kernels in the j -th feature map (y_a) (x_j). This layer continues where the convolutional layer left off, and then some. After feature maps are developed, a pooling layer is often added to them to reduce the amount of feature maps and network parameters that must be constructed. Maximum pooling and global average pooling were used in this experiment. Maximum pooling is used to choose the largest value from the given matrix size of each feature map, resulting in a smaller number of output neurons than would be the case otherwise. Data is reduced to a single dimension using a global average pooling layer. This layer is utilized just before to the completely linked layer and not thereafter. After being connected to the global average pooling layer, it is connected to the fully connected layer. Between the two other layers, the dropout layer serves as an additional intermediate layer. This layer [44] may be able to avoid the occurrence of network overfitting and divergence. The last and most critical layer in CNN's design is the "fully connected" layer. This layer may include multi-layer perceptrons. On completely connected layers, both the Rectified Linear Unit (ReLU) and the Softmax activation functions are often utilized, with the ReLU activation function being the most frequently used. The two activation functions that were

calculated are as follows: When x_i and m are the number of classes and corresponding input data, the outcome is a ReLU in which If the preceding layer is fully linked, the succeeding layer's activation functions are completely coupled to each and every neuron. models that have been previously taught in a certain manner

3.2.4 One of the most challenging difficulties that researchers encounter while analyzing medical data is the scarcity of accessible datasets. Large amounts of training data are often required for good deep learning models. Expert data labeling is a time-consuming and expensive procedure that demands specialized knowledge. The primary benefit of employing transfer learning is that it enables data training to be performed with fewer datasets and needs less computational time. Transfer learning is a frequently used approach in the area of deep learning. It entails transferring the knowledge gathered by a pre-trained model on a large dataset to the model that is being trained.

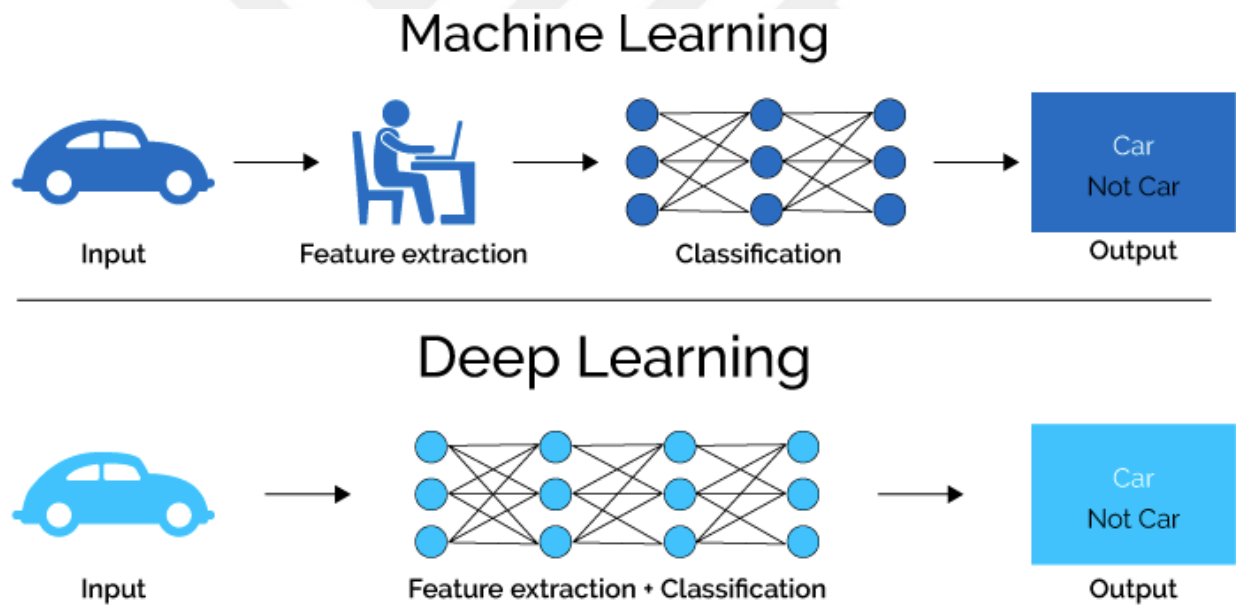


Figure 3. 3:Deep learning and Machine learning

3.2.1 CNN

CNNs use a variety of techniques for predicting the class label in an input image, including gradient class activation maps and average global pool class activation mappings. [35] [14] [25]. [25]. To appreciate the output selection of the CNNs examined in this study, it is important to create heat maps of the most discriminating areas of the input images, and class activation mappings are used to do this (CAM). It may be possible to attract attention to the class-specific

regions that CNNs use in their prediction process by using CAMs. When a test image is put into the network, the network uses the feature maps from the previous convolutional layer to construct a feature map of the test image. This section will provide a high-level summary of the techniques used to create the average pooling operation and input from SoftMax, including the following: It is advised to use the value '1' to indicate the weight of the unit class containing the class label. The following are the "one critical activation highlights" for each class. SoftMax's probability score for a given class may be defined as follows: By inserting Equation, the following results are obtained (1). By using SoftMax, we can get the activation output for each of the class labels in equation (5) in the following manner: $1 = 1, 2, 4, 1()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$, $1 = 1, 2, 4()$. [6/] (6) The CNN classifies an image input into one of two categories using activation maps in the space grid ($\cdot$), which contribute to this classification decision because to their strong discriminating power. Following that, the activation map is bilinearly upsampled to match the size of the input images that each CNN model takes. Figure 4 illustrates the image regions that CNN models use to anticipate class numbers accurately in a classroom scenario. Despite its relevance, CNN algorithms pay less attention to the frontal area of the chest, which contains indications and symptoms of COVID-19 and other disorders. Row 1 of row 1 demonstrates how people derive their options from the photographs rather than the text and medical equipment in rows (d) and (e) of photos with text and medical equipment, as well as places beyond the front view of the chest area (b) Similarly, these artifacts may result in incorrect data categorization. This argument is shown in Figure 5, which depicts examples of CNN images and their accompanying CAMs that have been incorrectly labeled. Figure 5 illustrates an x-ray image with the letter R, which is nearly often employed by architects to make a poor decision in the fourth column, as seen in the example below. According to our findings, relying only on radiation pictures without preprocessing the region of interest and removing some hidden noise is inadvisable and results ill an erroneous and misleading comparison. The purpose, for want of a better description, is to construct a CNN model that is exclusively based on these traits and is mindful of COVID-19 symptoms.

Pneumonia Detection using Convolutional Neural Network (CNN)

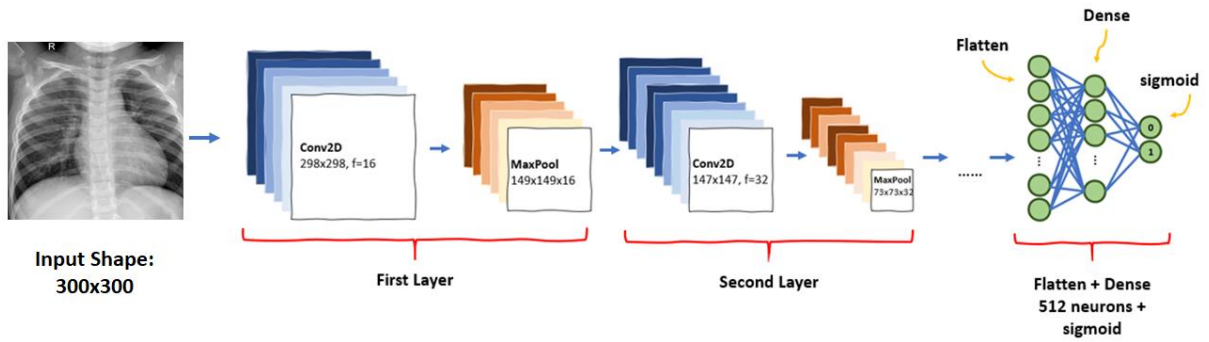


Figure 3. 4: CNN chest x-ray architecture

The new data sets were created by starting with an existing pneumonia (X-ray image). Despite the large sample size, the sample size for COVID-19 was randomly chosen from the Kaggle database to avoid asymmetric sample size changes that could affect the sensitivity and specialty of the results. According to the researchers' study questions, the sample sizes for bacterial pneumonia, non-COVID-19 viral pneumonia, and normal pCXR were 492, 552, and 532 patients, respectively. Additionally, each photograph was examined by a thoracic radiologist to ensure that it was of the greatest possible quality. The analysis of images is a great application for a CNN, a kind of neural network. A filter (or "window") scans each segment of an image, looking for distinguishing components such as edges and colors, as well as patches and other features, in order to classify the image using the classic CNN paradigm. To maximize computational efficiency and ease of implementation, we developed our CNN on the basis of the VGG16 artificial neural network architecture (Simonyan & Zisserman, 2014). Our VGG16 architecture consists of thirteen convolution layers, each of which contains a set of filters for extracting critical attributes. Transfer learning is used to accelerate the training process for the VGG16 model, which was trained on the ImageNet database. As a result, rather than having to learn all of the connections from scratch, the model is already aware of the use of transfer learning. Because the VGG16 framework requires 224 Euros for 224 pixels as a minimum, all files were converted to RGB images and then scaled down to be compatible. After that, images were compressed in a single pass and distributed 75/25 between the training and testing groups. The computer will have a much easier time interpreting all labels if they are converted to numerical numbers containing nozzles. VGG16 contains 13 convolutional layers: 5 max pool layers and 3 dense layers, for a total of 21 and 16 weighted layers,

respectively (Ren et al., 2020). Conv 1 has 64 filters, Conv 2 contains 128 filters, Conv 3 contains 256 filters, and Convs 4 and 5 include 512 filters apiece. Conv 1 is the smallest configuration among the five. The first two levels include sublayers, but the fourth and fifth levels have three sublayers on each level. Numerous sublayers exist inside the convolutional layer, each of which filters and sends information to the layer below it in turn. The data from all sublayers is gathered and sent to the next layer, where it may be utilized to create a more exact prediction. At the conclusion of each model stage, a maximum pooling layer was used to examine the input and identify its important attributes, as described in Ren et al., and the findings were published in the journal Science (2020). When used with a max pooling layer, you may assume features in the image's subregions while concurrently lowering the image's total size (Ren et al., 2020). All convolutional layers employ rectified linear units (ReLUs) as their activation function, which results in the model having a small number of learning parameters (Ren et al., 2020). Each of the three entirely connected tiers included a total of 4,096 nodes, for a total of 4,096 nodes. Ultimately, the models will be constructed of entirely linked layers, with each layer's inputs connected to the activation unit of the one below. As a result, the use of dropout layers and other approaches has grown.

3.2.2 Feature Extraction

For these techniques, the guidelines are based on professional knowledge: how is a normal chest produced and what are its factors? Simple guidelines for characterizing the properties of a chest and its spatial connection are straightforward to develop. For instance, a chest is commonly shown in a picture with two symmetrical eyes, a nose and a mouth. The connections between characteristics may be represented by their respective distances and placements. However, a fair rule including just one type of chests is exceedingly difficult to establish. The technique described by Yang and Huang in [3] is a typical way of this technique. They employed a hierarchical approach for detecting chests (a simple average and sub-sampling). • All the probable chest candidates may be discovered at the highest level by swipping a window on the input picture and applying a set of criteria at each site. The rules here describe how the chest looks. The coded criteria used for finding chest candidates with the lowest resolution are: 1st rule: four cells with a substantially consistent intensity in the center area of the chest. Towards 2 rules me, the upper circular chest piece has essentially a homogeneous intensity. — 3 Rule: is there a considerable

difference between the center part's average gray value and the top circular part? The lowest resolution picture for candidate chests is evaluated and it should be inspected in further detail.

a. Local histogram equalization is performed to the bottom level of chest candidates, followed by edge identification.

b. At level 3, the regions of candidates are next assessed using a different set of algorithms that react to chest organs like the eyes and lips. Lower rules are based on covid component details.

The distinctive element here is that a screening procedure is used to identify potential chests. Let $I(x, y)$ be the pixel (x, y) intensity value of the picture. The horizontal and vertical profiles are specified:

$$HI(x) = \sum_{y=1}^n I(x, y)$$

$$VI(y) = \sum_{x=1}^m I(x, y)$$

Where m and n denote the picture width and height. The horizontal profile of an input picture is first obtained, and then both local minima, which are found by detecting rapid changes, may match the left and right sides of the head. The vertical profile is similarly derived and the local minimum for the placement of the lips of the mouth, tip of the nose and eyes is calculated. These discovered features are a contender for a chest. Figure 3.4. An example shown where the chest borders are consistent with the local minimum and where there are dramatic fluctuations in intensity. Next, these candidates will be validated by the detection rules. The strategies of this method are often appropriate to solve the issue of chest localisation in photos with a homogeneous backdrop. Shows a counterexample of the poor result: by using this approach you cannot find the chest due to the non-uniform backdrop of the picture. The difficulty in converting human knowledge into well-defined rules is a downside to this technique. If the rules consider the details too much, a few chests which do not check all rules may be missed. If the rules are overly broad, a lot of positive errors may be created. On the other side, this strategy is difficult to identify chests in various situations as it may propose rules that are fair in every circumstance. Moreover, the chest method works only when frontal chests are detected in simple backdrop situations.

3.3 SVM Classification

SVM is a frequently used approach in chest detection systems. Vapnik proposed the theory of statistical learning [17]. A learning algorithm is an algorithm which automatically selects the optimum function to characterize a machine's input/output connection based on a limited number of samples. This methodology enables a surchest to be found which divides the data classes best by maximizing the margin between these classes. In contrast to empirical risk minimisation-based learning techniques, the risk calculated on learning data such as medium squared errors is based on the notion of structured risk minimisation. This reduces the top limit of the real inaccuracy to a minimum. This allows you to generalize the found machine[18]. A SVM classification function is a straight line in which a separation hyperplane has been set to minimize the predicted classification error of the models being evaluated. This ideal hyperplane is determined by a limited group of formation vectors known as vectors of support. A basis of examples where samples S of the size m are then m independent and equally distributed:

$$S = (x_1, y_1); (x_2, y_2); \dots; (x_m, y_m)$$

To appropriately portray sample I in the context of the supplied x_i , it is the expert's responsibility to provide the customer with sample I and its value y_i ($y-1, 1$). Our objective, among an infinite number of classification functions, is to determine the ideal linear classification function that minimizes generalization error. Once it has been established that both classes are linearly separable, the space separation hyperplane must be located. Our objective, however, would be to lower the total number of errors if the classifications were not easily distinguishable. The hyperplane with the largest range between two classes (a margin is defined as the sum of the distances between the hyperplane and its narrowest points), which minimizes misclassification errors, is an intuitive choice. Support vectors are points on the class boundary. All other data points in the dataset can be removed without affecting the outcome, and these vectors are the only ones that can be used to resolve the issue. Support for SVMs has grown significantly in recent years as a result of their large-scale data processing capabilities, minimal hyper-parameter requirements, theoretical stability, and impressive practical results. Vector support machines perform on a par with or better than neural networks and Gaussian models [19].

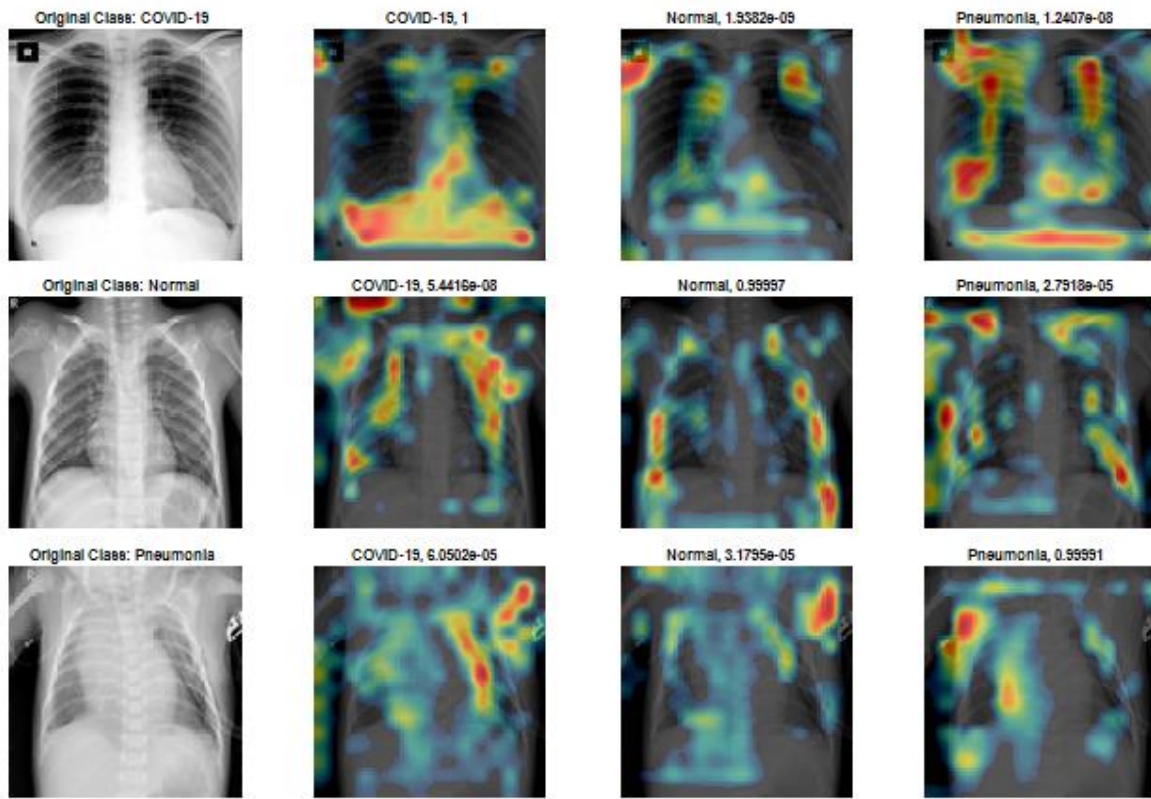


Figure 3. 5: SVM X-ray classification

Thanks to the classification capability, an SVM-based methodology is commonly used to address form identification issues, and in particular covid detection, which may be reduced to a classification issue. Haizhou, Lihang Ying and Guangyou Xu [20] are proposing a covid detection method consisting of 2 SVM stages: O Stage 1: A linear SVM filters the covid candidates of the Entrance picture from all observed windows. O 2nd stage: A nonlinear SVM decides if a chest candidate is a chest or not. The observing window is standardized to 20x20, the chest candidates may be discovered in all locations of the picture by scanning this window. such that chests of varying sizes may be found Osuna, R. Freund and F. Girosiont[14] are developing an effective approach to train an SVM to resolve big issues and deal with the difficulty of chest recognition. In order to train the system, they create a database with chests and non-chests. Each design is 19x19 in size. Preprocessing activities are conducted before creating the SVM: histogram equalization, mask, etc. The experimental findings indicate that this strategy is superior than Sung and Poggio The system operates 30 times quicker when results are equal to Sung's. In the previous decade, the

area of covid detection has progressed significantly. The release in 2001 of the Viola and Jones approach [21] in particular was the first way to recognize chests in real time. Viola and Jones's approach enables real-world chest sensing applications such as digital cameras, mobile phones, security cameras to be used.



4. PROPOSED METHOD

Algorithms are quite important when it comes to networking. Routing is decentralized in older networks. To enable SDN to function, the controller must configure flow tables in the hardware and then transmit them to the switches, as detailed previously. The quality of service (QoS) optimizes the utilization of network capital. According to established routing principles, route decisions should be made with the controller's global perspective in mind. This analysis makes use of the KDD network traffic dataset. Table 4.1 comprises samples of distributed denial-of-service attacks from the KDD and CSE-CIC-IDS 2018 security systems. It is the smurf attack that has the most samples in KDD and is used to drain network resources by broadcasting a huge number of Internet Control Message Protocol (ICMP) packets to a target system. The attack is disseminated to the target system through the use of a bogus IP address. Neptune, the second-largest sample attack, uses SYN flooding to create a broken TCP session, depleting the target server's resources. With the exception of the Smurf and Neptune attacks, all subsequent KDD attacks are considered as Denial of Service attacks. However, there are insufficient data points available to develop viable training models. Table 2 for CSE-CIC-IDS 2018 contains a list of reported denial-of-service attacks. These distributed denial-of-service attacks are more advanced than those launched by KDD. Our team of researchers classifies DoS attacks connected to CSE-CIC-IDS 2018 and KDD in binary and multi-class approaches. We will execute this experiment if our DL model KDD has 41 traffic characteristics and one feature that identifies the data collection to which each piece of traffic belongs, as mentioned previously. Each of the 41 distinct traffic characteristics is quantified, with 38 being mathematically represented and three being symbolically represented. We quantitatively represent symbolic data in order to centralize all data types. It is composed of three components: the TCP/IP layer protocol type, the destination system's service type, and the flag type, which reflects the session's connection state. The following table summarizes the TCP/IP layer protocol types: Protocol types include, but are not limited to, the ICMP, TCP, and UDP protocols. These protocols must be encoded using one-hot encoding, which produces an asymmetric three-dimensional vector containing the values 1, 0, and 0. The nine properties of the flag type are represented by a nine-dimensional vector, as are the 67 service types that comprise HTTP and FTP. The 79-dimensional vector is eventually created with the assistance of these adjustments.... When the 79-dimensional vector is added to the 38 qualities that make up the

original numerical characteristics, a 117-dimensional vector with 117 dimensions results. Additionally, the numerical characteristics are scaled to be between 0 and 255, resulting in 13 9-pixel images from a 117-dimensional information vector. Each color channel in each photograph should be assigned a value between 0 and 255. Following that, we submitted these images into the CNN model, which generated the results shown above. Given that CNN is a data-rate training model, the numerical input must be converted to images prior to training with it.

Table 4. 1: the KDD dataset structure

<i>Classification</i>	<i>Name of attack</i>	<i>Number of samples</i>
<i>Remote to Local (R2L)</i>	buffer-overflow, load-module, perl, rootkit, xterm, ps, sqlattack	229,853
<i>DDoS</i>	Neptune, smurf, pod, teardrop, land, back, apache2, udpstorm, processtable, mail-bomb	16,347
<i>Probing</i>	port-sweep, ip-sweep, nmap, satan, saint, mscan	70
<i>User to Root (U2R)</i>	guess-password, ftp-write, imap, phf, multihop, spy, warezclient	4166

4.1 SDN SECURITY ISSUES

A denial of service (DoS) assault can be defeated by flooding the target with traffic until the target is unable to reply to valid requests from legitimate users; this is what happens in a DoS attack. Distributed denial-of-service (DDoS) attacks are also possible, in which the attacker uses a Trojan horse virus or something similar to infect a large number of zombie machines, which are then utilized to attack the target concurrently (server, website, etc.)

Due to the difficulty in distinguishing between genuine and malicious requests, this is an exceedingly difficult attack to decode.

The effect of distributed denial-of-service attacks on SDN networks Due to the SDN design's inherent concentration of control, denial-of-service (DoS) attacks can have a significant influence on network performance, culminating in instances where the entire network is unable to respond to user requests. Each of these assaults is directed at a different component of the SDN controller, such as the connection between the controller and switches or the transmission plane (the switches and the network links). The control aircraft was subjected to a denial-of-service attack. Denial of service (DoS) or distributed denial of service (DDoS) attacks involve simultaneously flooding the SDN network with a high number of streams (all at once in the case of DDoS). When new flows are received at the infrastructure layer, switches communicate with the controller to obtain switching rules that enable them to route those flows to their intended destinations. As a result of the overload, SDN controllers will be overwhelmed with requests from the network's data plane, rendering them completely immobilized and incapable of making routing decisions. A denial of service attack was employed to launch the attack on the data plane (DDoS). Switches, on the whole, must keep and use switching rules until timers, such as an idle timeout or a "hard" timeout, have expired. When an attacker floods a switch with traffic, the controller converts all of the data into switching rules that enable it to be transferred to its intended destination as rapidly as feasible. Despite this, the controller's rules will continue to fill the switch's TCAM until it reaches its storage capacity. Congestion and transmission delay occur as a result of switches frequently adding and removing flow rules and sending additional controller requests. The DDoS attack had an effect on the controller-to-switch connection. Due to the high volume of communication between the SDN controller and the switches that require routing decisions, numerous "packet-in" messages are lost or respond slowly. This is occasionally referred to as "control plane bandwidth," as it is the bandwidth needed to connect the controller and switches.

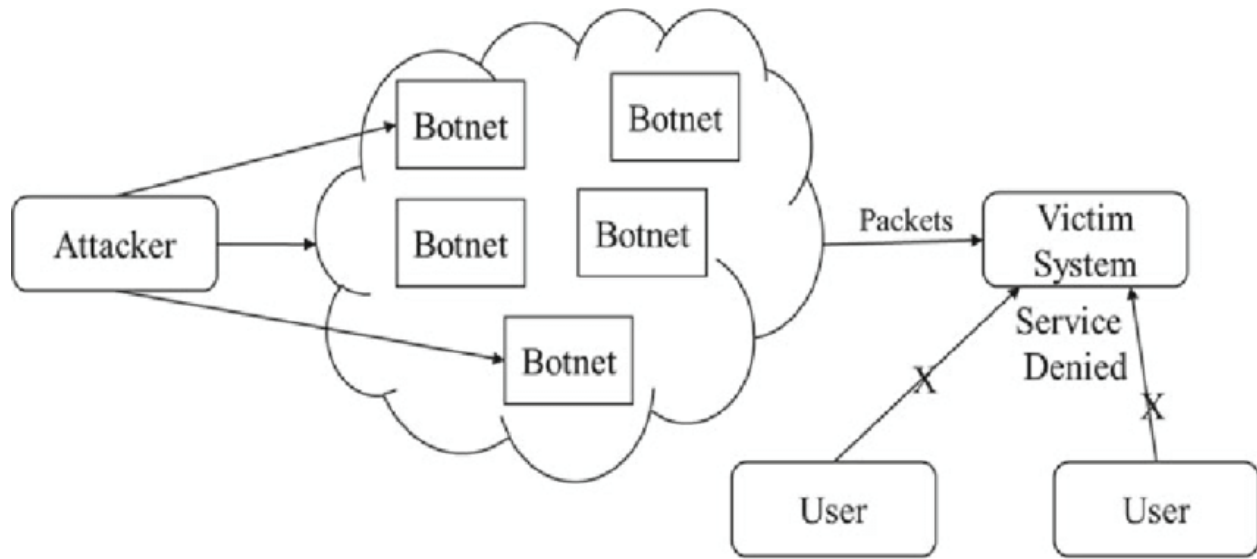


Figure 4. 1: DDOS attack on a victim

4.2 INTRUSION DETECTION IN SDN

This section discusses Intrusion Detection Systems (IDS) in detail and the various types of IDS accessible. Additionally, we'll discuss the two distinct methods intrusion detection systems search for intrusions. An intrusion detection system (IDS) is a tool that continuously monitors network traffic for suspicious packets. It is used to monitor a network for harmful behavior. The main purpose of IDS is to protect the network against intrusions and ensure its security. There are several open-source IDS like Snort and Suricata (Suricata, 2015). The use of IDS in networks makes it possible to collect information on intrusions, to ensure centralized management of alerts and to carry out an initial diagnosis of the nature of the attack, thus making it possible to react actively and quickly to the attack to slow it down or stop it. There is a possibility of IDS detection error which varies from system to system. We are talking here about false positive alarms and false negative alarms. An alert is said to be false positive if the IDS detects legitimate activity as malicious activity while an alert is said to be false negative if the IDS accepts malicious activity as legitimate activity.

4.3 IMPLEMENTATION OF IDS- FOR SDN

Finally, this subsection will present the last diagram used, the Deployment Diagram, which aims to expose the technologies used to develop the proposed application. To perform the capture of

IPs and packets on the network, IDS- is dependent on two other technologies: Nmap and the distribution operating systems, it needs a configuration file with the extension “.so”. This configuration file needs to be recompiled so that it is compatible with the processor architecture of the device in question; the device used has processor architecture. IDS- also depends on the iptables Firewall to block packets from attackers. In order for the rules to be generated, a connection is made to the database, using the TCP protocol, in order to identify the attacker's information — since the rules are generated according to the information extracted from the attack.

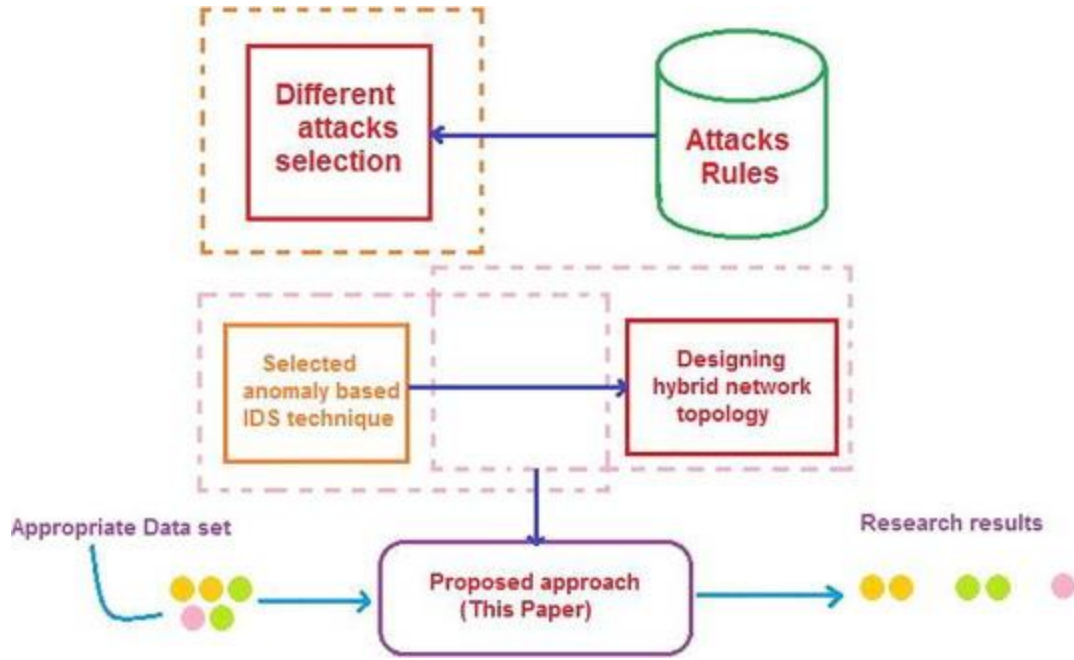


Figure 4. 2: Proposed SDN Intrusion System

This subsection will present some of the pseudo codes used in the IDS- implementation, in which Figure 4.2 is related to the general execution of the developed system. Initially, the variables are defined. Variable true is of type boolean, with assignment true — so that execution enters a loop of execution without stopping condition. The variable time is of type Timer, which stipulates the time for the capture of packets. Initially time is set to 0 (seconds). When starting the analysis phase of the captured packets, the number of packets belonging to the same sender and destined for the same receiver is verified. In fact, this verification is of paramount importance, as the purpose of performing a denial-of-service attack is to send an unnecessary amount of packets to a victim, thus performing a flood attack. Figure 4.3 shows the process of analyzing the captured packets. If it is detected that the number of packets is greater than the established limit, the process of analyzing

the characteristics of packets — extracted in the capture phase — is started, in order to be able to identify the correspondence with the type of attack that the device is in. receiving. After identifying which type of attack is being carried out, the attacker's information, as well as the type of attack, are inserted in the historical table of the database. Once the phase of identifying the attacker's information is finished, the blocking rules generation phase begins.

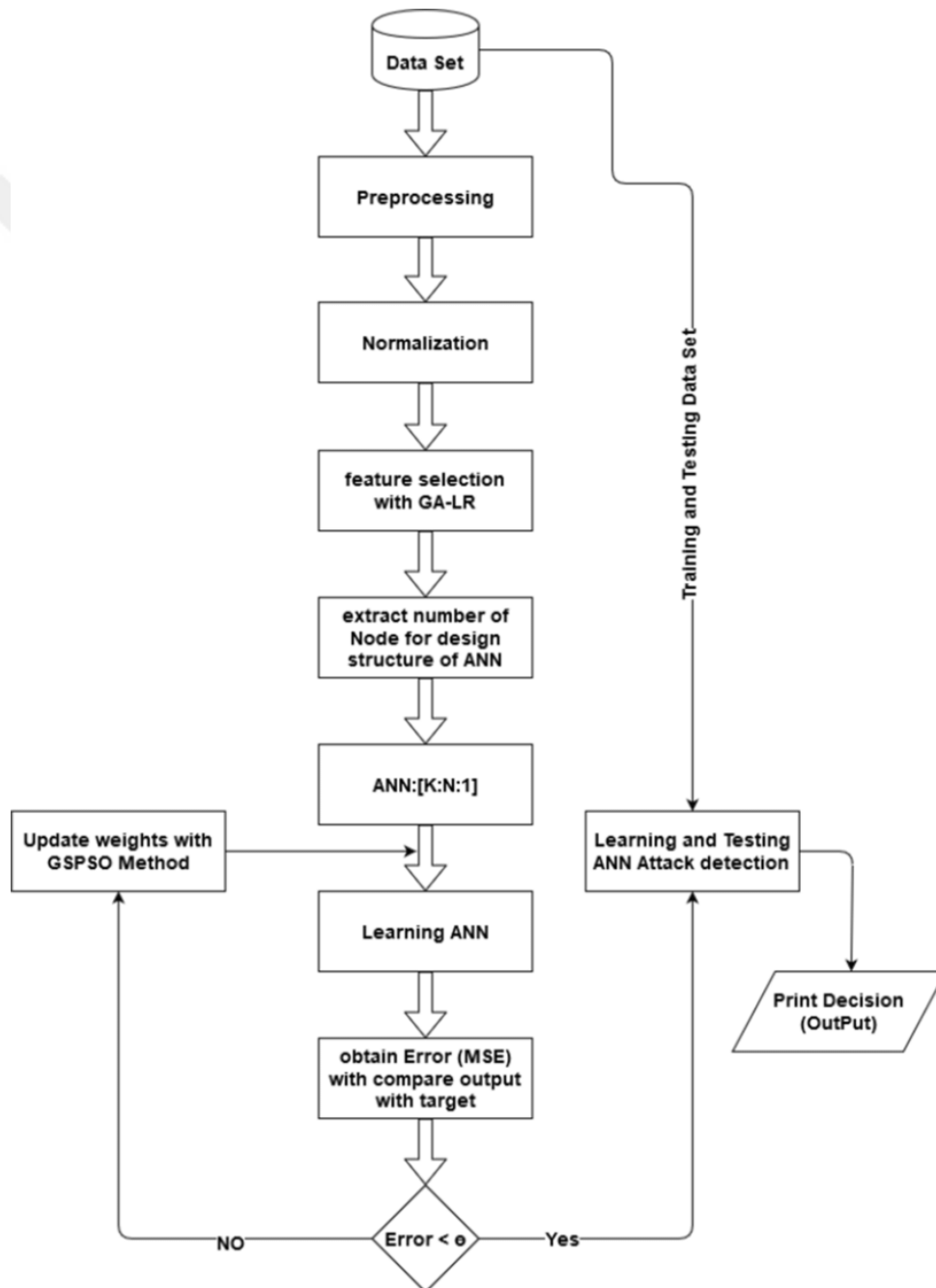


Figure 4. 3: Attack model for the IDS

In the appendix presents the pseudo code for the capture and identification of packets according to the type of protocol corresponding to each packet. Each threat type has the corresponding protocol for carrying out DoS attacks. In order to evaluate the proposal in the exposed IDS, a prototype was developed in order to have a better understanding of how it will behave in a real application. the results obtained when using the IDS- as an Intrusion Detection System in a Raspberry Pi will be presented. shows the messages displayed in the IDS- execution process, which shows the beginning of the IP capture step, showing the time in milliseconds spent for the execution of the nmap command. After the end of the capture, the packet capture start message is displayed, in which the defined capture time is awaited to end the analysis phase. Three analyzes are performed: TCP, ICMP and UDP packet analysis. The time taken to analyze each type of packages is also displayed. The software system used in testing the first scenario was a streaming server (Web server) based on Apache server version and Operating System deployed. According to the scenario presented the web server is running on the device In the tests carried out, all other devices are accessing the web pages and showing the videos provided by the server, thus inserting a significant amount of packets into the network. In this section, the results obtained using IDS- are presented.

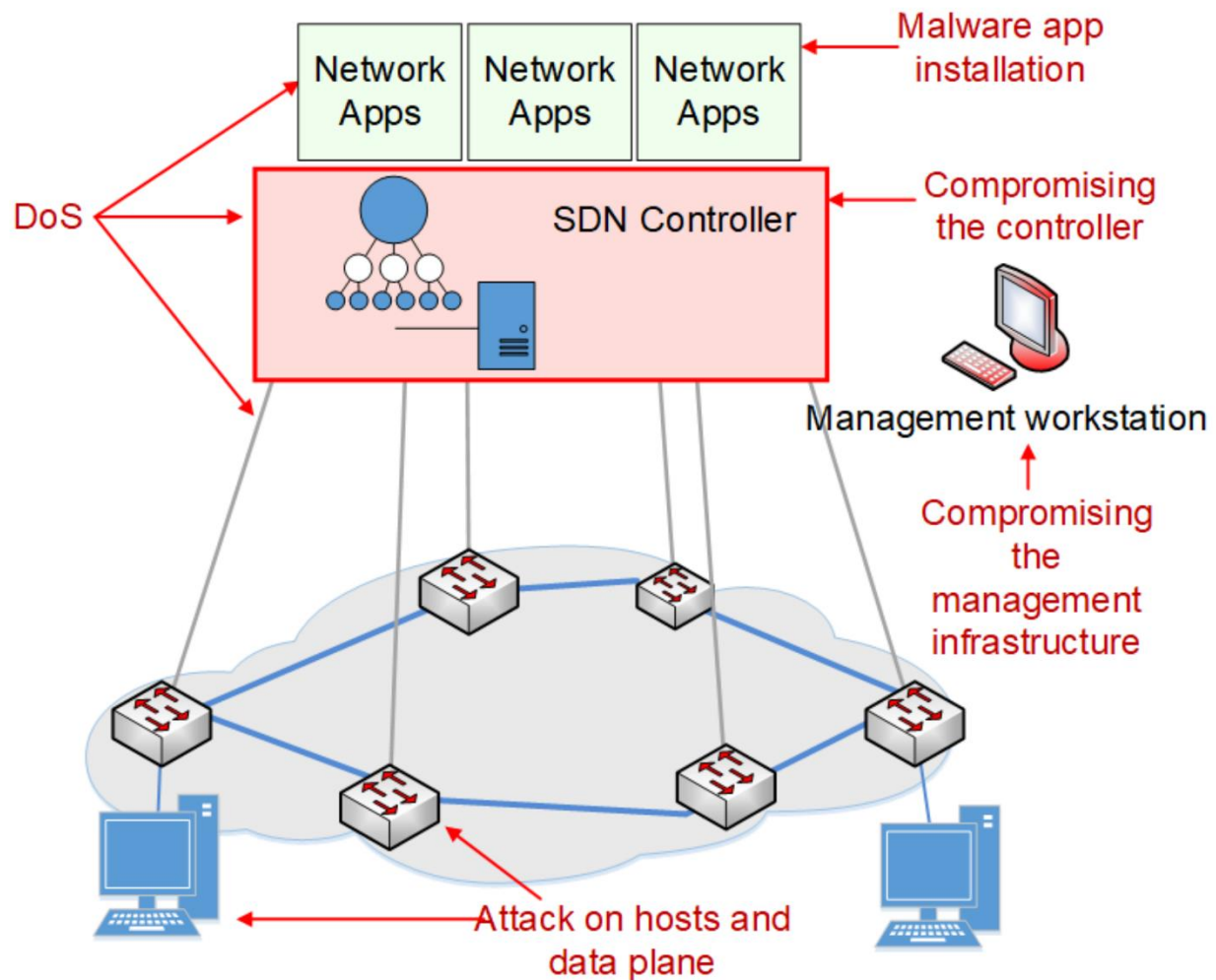


Figure 4. 4: Network Dynamics for the used data

We'll go over the commands for the hping3 tool in this part, which we utilized to launch the attacks. The "Machine 3" gadget was used in each of the attacks detailed in this section. If you're interested in learning more about Syn Flood, you may read the documentation to learn more about the attack's three-way handshake. When a client seeks to interact with a server, the client sends a request with the SYN (synchronize) flag to the server. Finally, the client transmits an ACK flag to the server, eliciting a response from the server in the form of another SYN-ACK flag [7]. a step-by-step guide to the handshake The server communication is used by attackers to launch a Syn Flood attack against the target. SYN packets are delivered in large quantities to the server, and the server gets SYN-ACK packets in response to the massive quantity of SYN packets transmitted. Although the ACK flag keeps the connection open, the connection state is not displayed during a Syn Flood

assault because server resources like as bandwidth, memory, and processing are drained throughout the attack, as illustrated in Figure 1. Following the Syn Flood attack, the intrusion detection system (IDS) accurately identified the perpetrator and noticed typical Syn Flood packet characteristics.

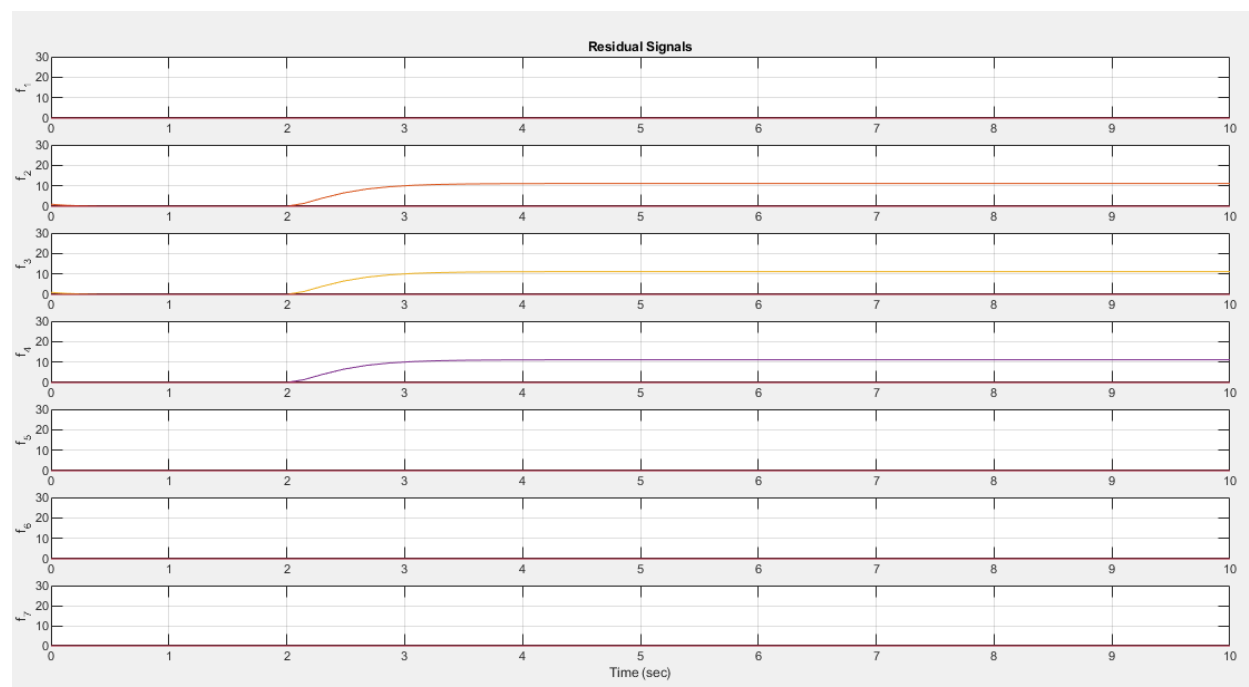


Figure 4. 5: Data rates in different devices

This attack is capable of sending packets to the server that share the server's IP address and/or share the server's source and destination ports, causing the server to enter an unending loop when attempting to react to the client's request as a result of the Land attack. It is easy to determine how much time was spent examining the received packets. The Land Attack strike was carried out twice, for a total of two minutes per attack. The following is an illustration of the blocking rule generated: The A FORWARD and J DROPPING commands enable you to do the same operation in two separate ways. The following are the fundamental rules: Although receiving attack packets with -A FORWARD and transmitting them to the target IP address using -s is possible, it is not advised. Finally, the -j DROP command "discards" all packets from the source IP address to a specified destination IP address. As illustrated in Figure 5.9, the Land Attack attack altered the device's resource allocation. Processing usage began at 3% and peaked at 10%. Memory utilization begins at 196 MB and rapidly increases to a high of 354 MB over time. Open your web browser

and type `hping3 -a -flood` into the URL bar to launch an ICMP Flood attack. The `-1` option specifies that ICMP packets will be transmitted to the target IP address, hence overloading the victim's bandwidth with the denial-of-service attack. As the caption "Non-attack analysis" indicates, it takes time to examine packets in the absence of network attacks. As illustrated in the picture below, the first and second assault analyses correspond to the time required to carry out packets during an ICMP Flood attack, with red representing discovered invaders and green representing those who are no longer in play.

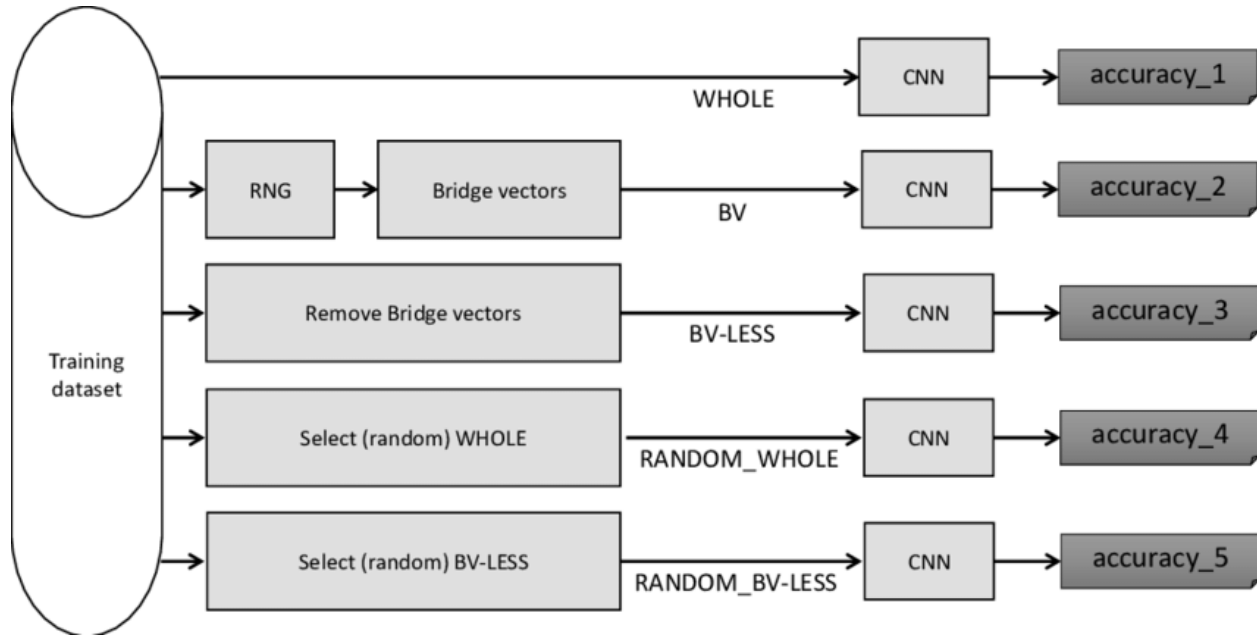


Figure 4. 6: Detected attackers using the CNN-SVM workflow

The ICMP Flood assault had to be carried out twice, with each attack lasting around two minutes in total. For example, when an ICMP Flood attack is performed, the device's resources are allocated differently than usual. The initial processing utilization rate was 3%, increasing to a maximum of 11% at its peak. The device's memory consumption begins at 145 MB and peaks at 314 MB. The Smurf Attack was another ICMP-based attack. This threat is directed at the broadcast address of the network, which is 192.168.1.1. To begin, type `hping3 -flood`, followed by `hping3 -flood` again. The Smurf Attack and ICMP Flood attack commands are quite similar, with the exception that the Smurf Attack floods the network broadcast with fraudulent requests, rather than flooding the device's bandwidth with phony requests. As a result, two distinct sets of IP table rules were created: one for INPUT and one for DROPPING INPUT. The diagram below illustrates the

time required for IDS- to examine the packets sent by this form of assault. As the caption "Analysis without attack" indicates, analyzing packets without resorting to network assaults takes time. "Attack time" refers to the time spent on the "first attack analysis" and "second attack analysis" phases of the Smurf Attack attack. Each Smurf Attack lasted two minutes and was repeated twice. The final attack was a UDP Flood. When an attacker uses UDP Flood, the victim's bandwidth is completely flooded with UDP traffic. At the moment this command was sent, it was decided to initiate an attack on the target. Hping3 has been affected by two consecutive floods. After the attack was discovered, the following rule was established to prevent the sender's packets from being delivered: iptables -A INPUT followed by iptables -A FORWARD. The UDP Flood attack was carried out twice in sequence, with a two-minute pause between each attack.

4.4 MACHINE LEARNING DATA CLASSIFICATION

ML (machine learning) is a field that aims to study algorithms that would be able to learn from their own mistakes. With each new experience acquired, it must become more reliable in its resolutions, to the point that it can be used in order to make accurate predictions with its data [23-24]. The algorithm in question receives the data that must be evaluated, these may be present in a database, for example. The machine will then analyze the information received, and with the use of statistical properties, it will deliver an answer according to the proposed problem. The learning method can be classified in three ways: supervised, unsupervised, and semi-supervised. In the first, during training, for each input to be analyzed by the algorithm, the expected output is also provided. This output is also known as a label. It is expected that at the end of learning, the program has created rules that make it possible to solve similar problems. The categories of issues that can be resolved are called regressions, where the answer is an actual value, such as price or quantity, or ranking issues, where the output is a category, such as malicious traffic or legitimate traffic. As for the unsupervised method, the algorithm receives as input only the attributes that must be evaluated, that is, without the expected output. Unlike the supervised model, this model does not go through a learning phase, the algorithm seeks to learn data patterns by comparing them and organizing them into groups according to their attributes. This model is intended to learn more about the data, so there is usually no right answer. It can be used, for example, to group users by their tastes or behaviors. Semi-supervised methods combine the previous two. Some of the tests follow the expected output, some don't. This scenario usually exists when input is very numerous,

and it would require a lot of time, people, or money to label all expected outputs. Unlike batch learning, these algorithms only receive one example at a time. In this method, the algorithm uses the characteristics of the data sets to organize them according to their similarities, then the known labels are used to predict those unknown in their region. A classic example is image classification, some can be labeled as cars, others as motorcycles, but most do not have a label. In the end the program would be able to identify the images on its own. It is common for learning to take place through databases, this method is called batch learning. In it, the machine must learn from information already stored. In this model, the algorithm receives a set of some or perhaps all of the training examples, then statistics are created on them. The advantage is the possibility of reusing each information several times, however, this technique is not very efficient when dealing with networks.

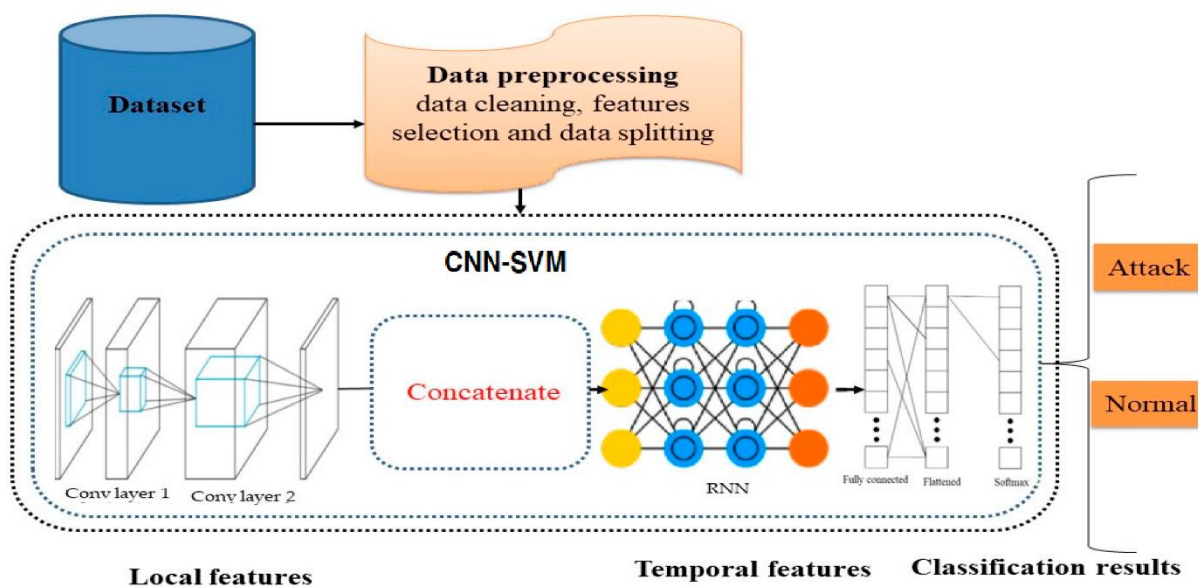


Figure 4. 7: CNN-SVM data classification

The traffic on the network in question can change its behavior, changing the statistical data, so the decisions made from the learning carried out before such change, can be wrong. Also, the data flow is often fast and bulky, which is a problem as not only is storage limited but it makes it virtually impossible to label entries. This traffic from networks can be called Stream or Data Stream. As batch learning is not suitable for these networks, it is necessary to use a different approach. For networks with this traffic, it is recommended to use stream classification algorithms.

Unlike batch learning, these algorithms only receive one example at a time. Stream classification algorithms [26] dispense with the learning phase, have a great performance in networks where drift exists and are especially indicated when traffic is intense. Some examples of these algorithms are presented in [27]. According to [26], the stream classification algorithms must follow four requirements: · Inspect only one element at a time, and do it in a single moment. · Have a maximum limit on the memory that can be consumed. · Work within a maximum time limit. · Be ready to make predictions at any time.



5. CONCLUSION

With the continued growth of DSN-IDS attacks, security monitoring and management mechanisms are increasingly becoming a necessity. As part of this problem, we looked for the missing pieces in the current defense plans. While emphasizing the specificity of the intended application on the one hand, and its financial and confidential sensitivity on the other hand, this thesis is completely complementary to the efforts pursued today in the research community and in the industry. In the first chapter, we introduced DSN-IDS protocols and underlined their distributed and heterogeneous aspect. We justified the security needs of these protocols with a brief overview of the most relevant threats. Finally, we presented the most recent work on the security and intrusion detection of this new technology. The contributions are synthesized in the second chapter. We have proposed interconnected solutions that fill the gap in terms of traffic monitoring, preventive defense and event correlation. Indeed: we have proposed a signaling traffic monitoring system (SIP) based on learning theory. The basic idea is to cut the traffic into small chunks or slots and extract values for predefined statistical variables or attributes. For each slot, the values of these attributes are grouped into a vector which will be given to the learning machine to classify it as normal or suspect. Correlation rules are thus applied to infer a conclusion on the traffic in progress. We used two different approaches: Bayesian networks and support vector machines and compared their performance. We presented an innovative approach to designate a DSN-IDS-specific honeypot. The honeypot has an application interface that allows it to manage a large number of network tools. In addition to receiving and recording attacks, the honeypot is able to conduct real-time investigation of received messages. An important component in its architecture is an inference engine that can judge whether a received message is a routing or enumeration fault, a malformed message issued to exploit some vulnerability, or This is a call that appears to be correct but is not trustworthy. Evaluations of the simulated examples show the efficiency of our investigation procedure and the accuracy of the decisions made by the inference engine. We have presented a multi-level distributed approach for the correlation of security events in a DSN-IDS domain. Our findings, which are based on host- and network-based intrusion detection systems, as well as the outcomes of our research, have been presented. Part One of this chapter discusses in depth how an intrusion detection system's (IDS) location influences its effectiveness and accuracy in recognizing threats. Following that, we examined how sampling can be utilized to reduce the amount of traffic

sent to the IDS by switches while also reducing the time required to process the traffic. We calculated that installing the IDS in the most optimal place possible might eliminate up to 90% of duplicate traffic to the IDS. We discovered that Snort can accurately identify distributed denial of service (DDoS) assaults by analyzing only 11% of the data. This section will examine SDN-overall Guard's performance. SDN-Guard, as proved in testing using Mininet, reduces the processing load on controllers by 32%, further lowering communication between controllers and switches. Additionally, SDN-Guard reduces switch memory consumption for storing flow rules by up to 26%. Additionally, the findings reveal that SDN-Guard dramatically reduces packet loss and average response time in SDN networks due to DDoS attacks.

REFERENCES

- [1] S. C. Mukhopadhyay and N. K. Suryadevara, "Internet of things: challenges and opportunities," in *Internet of Things: Challenges and Opportunities*. Cham, Switzerland: Springer International Publishing, pp. 1-17. 2014

- [2] O. Vermesan and P. Friess, Internet of Things-from Research and Innovation to Market Deployment. Aalborg, Denmark: River Publishers, 2014.
- [3] S. P. Anilbhai and C. Parekh, "Intrusion Detection and Prevention System for IoT," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, no. 6, pp. 771- 776, 2017.
- [4] S. Tanwar, S. Tyagi, and S. Kumar, "The role of internet of things and smart grid for the development of a smart city," in *Intelligent Communication and Computational Technologies*. Singapore: Springer, Singapore, pp. 23-33. 2018.
- [5] M. Anirudh, S. A. Thileeban, and D. J. Nallathambi, "Use of honeypots for mitigating DoS attacks targeted on IoT networks," in *Proceedings of International Conference on Computer, Communication and Signal Processing (ICCCSP)*, Chennai, India, pp. 1-4. 2017.
- [6] W. Meng, "Intrusion detection in the era of IoT: building trust via traffic filtering and sampling," *Computer*, vol. 51, no. 7, pp. 36-43, 2018.
- [7] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song, and K. M. Malik, "NBC-MAIDS: Naive Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5156-5170, 2018.
- [8] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680-698, 2018.
- [9] S. G. Kene and D. P. Theng, "A review on intrusion detection techniques for cloud computing and security challenges," in *Proceedings 2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, India, pp. 227-232. 2015.
- [10] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure internet of things," in *Proceedings of IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria, pp. 84-90. 2016.

- [11] C. Jun and C. Chi, "Design of complex event-processing IDS in internet of things," in *Proceedings of Sixth International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China*, pp. 226-229. 2014.
- [12] A. Patel, Q. Qassim, Z. Shukor, J. Nogueira, J. Junior, and C. Wills, "Autonomic agent-based self-managed intrusion detection and prevention system," in *Proceedings of the South African Information Security MultiConference, Port Elizabeth, South Africa*, pp. 223-234. 2011.
- [13] J. H. Lee, M. W. Park, J. H. Eom, and T. M. Chung, "Multi-level Intrusion Detection System and log management in Cloud Computing," in *Proceedings of 13th International Conference on Advanced Communication Technology (ICACT)*, Seoul, Korea, pp. 552-555. 2011.
- [14] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, article no. 21, 2018.
- [15] P. S. Kenkre, A. Pai, and L. Colaco, "Real-time intrusion detection and prevention system," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*. Cham: Springer, pp. 405-411. 2014.
- [16] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25-37, 2017.
- [17] S. Notra, M. Siddiqi, H. Habibi Gharakheili, V. Sivaraman, and R. Boreli, "An experimental study of security and privacy risks with emerging household appliances," in *Proceedings of IEEE Conference on Communications and Network Security, San Francisco, CA*, pp. 79-84. 2014.
- [18] C. Kolias, A. Stavrou, J. Voas, I. Bojanova, and R. Kuhn, "Learning Internet-of-Things security 'Hands-On'," *IEEE Security & Privacy*, vol. 14, no. 1, pp. 37-46, 2016.
- [19] O. Garcia-Morchon, S. Kumar, S. Keoh, R. Hummen, and R. Struik, "Security considerations in the IP-based Internet of Things: draft-garcia-core-security-06," *Internet-Draft, Internet Engineering Task Force*, 2013.

- [20] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496-3509, 2018.
- [21] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6LoWPAN based Internet of Things," in *Proceedings of IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Lyon, France, pp. 600-607. 2013
- [22] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016.
- [23] A. Abduvaliyev, A. K. Pathan, J. Zhou, R. Roman, and W. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223- 1237, 2013.
- [24] A. Gupta, O. J. Pandey, M. Shukla, A. Dadhich, S. Mathur, and A. Ingle, "Computational intelligence-based intrusion detection systems for wireless communication and pervasive computing networks," in *Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, Enathi, India*, pp. 1-7. 2013.
- [25] H. A. Arolkar, S. P. Sheth, and V. P. Tamhane, "Ant colony based approach for intrusion detection on cluster heads in WSN," in *Proceedings of the International Conference on Communication, Computing & Security, Rourkela, India*, pp. 523-526.2011.
- [26] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the RPL-connected 6LoWPAN networks," in *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, Abu Dhabi, United Arab Emirates*, pp. 31-38. 2017.
- [27] T. Jiang, G. Wang, and H. Yu, "A dynamic intrusion detection scheme for cluster-based wireless sensor networks," in *World Automation Congress, Puerto Vallarta, Mexico*, pp. 259-261. 2012.

- [28] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1 & 2, pp. 1-9, 2009.
- [29] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems: a survey of common practices," *ACM Computing Surveys*, vol. 48, no. 1, Article no. 12, 2015.
- [30] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, and J. Isoaho, "Distributed internal anomaly detection system for Internet-of-Things," in *Proceedings of 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, pp. 319-320, 2016.
- [31] G. Han, J. Jiang, W. Shen, L. Shu, and J. Rodrigues, "IDSEP: a novel intrusion detection scheme based on energy prediction in cluster-based wireless sensor networks," *IET Information Security*, vol. 7, no. 2, pp. 97- 105, 2013.
- [32] T. Sherasiya, H. Upadhyay, and H. B. Patel, "A survey: intrusion detection system for Internet of Things," *International Journal of Computer Science and Engineering*, vol. 5, no. 2, pp. 91-98, 2016.
- [33] I. Alqassem and D. Svetinovic, "A taxonomy of security and privacy requirements for the Internet of Things (IoT)," in *Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management, Bandar Sunway, Malaysia*, pp. 1244-1248, 2014.
- [34] H. A. Abdul-Ghani and D. Konstantas, "A comprehensive study of security and privacy guidelines, threats, and countermeasures: an IoT perspective," *Journal of Sensor and Actuator Networks*, vol. 8, no. 2, p. 22, 2019.
- [35] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of PUF-based hardware security solutions for the internet of things," in *Proceedings of 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi, United Arab Emirates, pp. 1-4, 2016.
- [36] P. Sethi and S. R. Sarangi, "Internet of Things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol., article no. 9324035, 2017.

- [37] D. M. Mendez, I. Papapanagiotou, and B. Yang, "Internet of Things: survey on security and privacy," [Online]. Available: <https://arxiv.org/abs/1707.01879>. 2017.
- [38] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25-37, 2017.
- [39] K. Xing, F. Liu, X. Cheng, and D. H. C. Du, "Real-time detection of clone attacks in wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems, Beijing, China*, pp. 3-10. 2008.
- [40] R. P. Kurbah and B. Sharma, "Survey on issues in wireless sensor networks: attacks and countermeasures," *International Journal of Computer Science and Information Security*, vol. 14, no. 4, pp. 262-269, 2016.
- [41] S. Fosso Wamba, A. Anand, and L. Carter, "A literature review of RFID-enabled healthcare applications and issues," *International Journal of Information Management*, vol. 33, no. 5, pp. 875-891, 2013.
- [42] M. S. Van Devender, W. B. Glisson, M. Campbell, and M. A. Finan, "Identifying opportunities to compromise medical environments," in *Proceedings of Twenty-second Americas Conference on Information Systems*, San Diego, CA, pp. 1-9. 2016.
- [43] J. Deogirikar and A. Vidhate, "Security attacks in IoT: a survey," in *Proceedings of International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, pp. 32-37. 2017.
- [44] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): taxonomy of security attacks," in *Proceedings of 2016 3rd International Conference on Electronic Design (ICED)*, Phuket, Thailand, pp. 321-326. 2016.
- [45] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eysers, "Twenty security considerations for cloud-supported Internet of Things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269-284, 2016.

- [46] A. Patel, S. Jain, and S. K. Shandilya, "Data of semantic web as unit of knowledge," *Journal of Web Engineering*, vol. 17, no. 8, pp. 647-674, 2018.
- [47] D. Jankowski and M. Amanowicz, "Intrusion detection in Software Defined Networks with self-organized maps," *Journal of Telecommunications and Information Technology*, vol. 4, pp. 3-9, 2015.
- [48] D. Jankowski and M. Amanowicz, "On efficiency of selected machine learning algorithms for intrusion detection in Software Defined Networks," *International Journal of Electronics and Telecommunications*, vol. 62, no. 3, pp. 247-252, 2016.
- [49] S. Rathore, P. K. Sharma, V. Loia, Y. S. Jeong, and J. H. Park, "Social network security: issues, challenges, threats, and solutions," *Information Sciences*, vol. 421, pp. 43-69, 2017.
- [50] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of Things (IoT): a review," *Journal of Computer Networks and Communications*, vol. article no. 9629381, 2019.
- [51] S. Hameed, U. M. Jamali, and A. Samad, "Integrity protection of NDEF message with flexible and enhanced NFC signature records," in *Proceedings of IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland*, pp. 368-375, 2015.
- [52] C. Liu, C. Yang, X. Zhang, and J. Chen, "External integrity verification for outsourced big data in cloud and IoT: a big picture," *Future Generation Computer Systems*, vol. 49, pp. 58-67, 2015.
- [53] Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in IoT security system," in *Proceedings of IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China*, 3, pp. 1129-1132, 2013.
- [54] S. K. Singh, S. Rathore, and J. H. Park, "BlockIoTIntelligence: a blockchain-enabled intelligent IoT architecture with artificial intelligence," *Future Generation Computer Systems*, vol. 110, pp. 721-743, 2020.

- [55] I. R. Chen, J. Guo, D. C. Wang, J. J. P. Tsai, H. Al-Hamadi, and I. You, "Trust-based service management for mobile cloud IoT systems," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 246- 263, 2019.
- [56] Z. Zhang, J. Jing, X. Wang, K. K. R. Choo, and B. B. Gupta, "A crowdsourcing method for online social networks security assessment based on human-centric computing," *Human-centric Computing and Information Sciences*, vol. 10, Article no. 23, 2020.
- [57] L. Megouache, A. Zitouni, and M. Djoudi, "Ensuring user authentication and data integrity in multi-cloud environment," *Human-centric Computing and Information Sciences*, vol. 10, Article no, 15, 2020.
- [58] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," in *Proceedings of 7th International Conference on Emerging Security Technologies (EST)*, Canterbury, UK, , pp. 138-143. 2017.
- [59] S. K. Singh, Y. S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, article no. 102252, 2020.
- [60] Y. S. Jeong and J. H. Park, "Security, privacy, and efficiency of sustainable computing for future smart cities," *Journal of Information Processing Systems*, vol. 16, no. 1, pp. 1-5, 2020.
- [61] S. Kumar, T. Kumar, G. Singh, and M. S. Nehra, "Open flow switch with intrusion detection system," *International Journal of Scientific Research Engineering & Technology*, vol. 1, no. 7, pp. 1-4, 2012.