T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

AN IMPROVED AGENT PROTECTION MODEL FOR DISTRIBUTED DENIAL OF SERVICE FLOODING ATTACK TRAFFICS

Haider Kasim Mohammd AL-HUSSEINI

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

AN IMPROVED AGENT PROTECTION MODEL FOR DISTRIBUTED DENIAL OF SERVICE FLOODING ATTACK TRAFFICS

by

Haider Kasim Mohammd AL-HUSSEINI

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled **"AN IMPROVED AGENT PROTECTION MODEL FOR DISTRIBUTED DENIAL OF SERVICE FLOODING ATTACK TRAFFICS"** prepared and presented by **Haider Kasim Mohammd AL-HUSSEINI** was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Sefer KURNAZ	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Mohammed Abubakar	College of Business and Social Sciences, Antalya Bilim University	_____

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

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

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

Haider Kasim Mohammd AL-HUSSEINI

DEDICATION

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

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM 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

AN IMPROVED AGENT PROTECTION MODEL FOR DISTRIBUTED DENIAL OF SERVICE FLOODING ATTACK TRAFFICS

AL-HUSSEINI, Haider Kasim Mohammd,

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

Supervisor: Asst. Prof. Dr.Abdullahi Abdu IBRAHIM

Date: June /2021

Pages: 56

Distributed Denial of Service (DDoS) is the most common and dangers type of attack could targeting the internet, web server, OSI model, and any devices have internet concoctions. DDoS attack try to flood the Internet and make it unusable for real users. In this work, Defense of Flooding Attacks model has been proposed to overcome DDoS attack and control the incoming traffics. Software agent has been employed to identify, classify and control traffics. The CICDDoS 2019 dataset has been used to test and evaluate the performance of the proposed model. According to the obtained results, it is observed the proposed model achieved an excellent results during the comparison with the related work with accuracy of 99.6%

Keywords: DDoS attack, Software agent and Application Layer

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

WWW	:	World Wide Web
OSI	:	Open System Interconnect
ISO	:	International Standards Organization
NALFA	:	Networks Application Layer Flooding Attacks
DDOS	:	Distributed Denial of Service
MA	:	Mobile Agent
DNS	:	Domain Name System
QOS	:	Quality of service
SVM	:	Support Vector Machines
TPR	:	True Positive Rate
TNR	:	True Negative Rate
FPR	:	False Positive Rate
FNR	:	False Negative Rate
CIDDS	:	Coburg Intrusion Detection Data Sets
MCP	:	Model Checking Period
APFA	:	An Adaptive Protection of Flooding Attacks

1. INTRODUCTION

1.1 BACKGROUND

The World Wide Web (WWW) appeared more than two decades ago. Globally it is changing people's lives through the availability of various web applications that are serving billions of web pages consistently [1]. Web applications have evolved long-term in a variety of structures, such as online agreements, email, wikis, online sales, and SMS administration, making them exceptional, smart, and dynamic. Given the significant work it does worldwide, it is critical to ensure that web applications are secure, accurate and of high quality [2]. Correspondence between the user and the web application is also created through the OSI layer. The last layer in the OSI model that represents the interface between the web application and the client is generally called the network application layer.

In order to provide a description of the defined layers within a network operating system, a layered model is known as the Open System Interconnect (OSI) model. The International Standards Organization (ISO) has created it. To illustrate the characterization layers in a network system, a layer model is called the Open System Interconnect (OSI) model. This was done by the International Standards Organization (ISO). With these layers, clearly defined functions that are capable of improving the internetwork connectivity between computer manufacturing companies can be provided. In order to have an insight into the communication of data in Local, Metropolitan or Word Wide Web (WWW), the function of each layer must be understood [3]. Attackers have continued to attack personal computers and networks, and the attacks vary in their magnitude and complexity.

One of the main problems that exist in network applications is Networks Application Layer Flooding Attacks (NALFA) problem, which occurs for several reasons such as financial fraud involving illegal money transactions. A typical example of this kind of financial fraud involves getting a commission from a clicked advertising banner [4]. Other attacks such as political attacks occur in the interest of a nation, while personal attacks are for personal issues involving individuals [2]. In other words, attacks occur for different reasons and at different levels. Moreover, among the NALFA,

The Distributed Denial of Service attack (DDoS) are the most well-known types of attacks, in which a network places so many additional demands on it that normal traffic is reduced or completely disrupted. Similarly, SQL Injection can be seen as a vulnerability that arises in the base of the application layer of an network [5].

Scripting Cross-Site between sites through user web applications executes malicious content (Alssir et al. [6], website defemation occurs when a programmer breaks into a web browser and changes the easy location or secretly makes Richardson et al. [7] words). , which break the path to retrieving mysterious passwords from information normally placed or launched in a computer framework by consistently checking aspects of the password [8].

One of the most feared attacks is referred to as the Distributed Denial of Service (DDoS), which poses several security problems in current networks. They are able to disrupt service availability largely by exhausting a huge amount of resources through the generation of congestions. According to Alomari et al. [9], one of the major threats, which the present-day electronic society is facing with, is DDoS, are well-planned attacks with a significant level of severity that can harm the highest level of information infrastructure [10]. The availability of different hacking tools makes it easy for such attacks to be launched. However, the attacker must possess the experience required to use the available hacking tools. In an ideal situation, one should be able to decipher the difference between abnormal traffic and a normal one. The difference between the activities of an attacker and a genuine user is the intention of either one [8]. In this thesis, the term of ‘demons’ refers to the real DDoS attacker as in the work of Alomari et al. [9].

It has been found in the literature that software agent technology is used to complement filters; this is one of the ways through which the network can keep working even when under attack [10]. Here, an agent refers to any kind of software or hardware that is used autonomously to monitor environmental changes and respond to them accordingly through the aid of features like adaptivity, learning ability and mobility. This agent is incorporated into the network so as to enhance the continuous functioning of the network even when it is attacked while creating scalability and adaptability in solutions that are aimed at addressing the existing and future problems of the network [5].

Several studies have employed the use of agent technology for DDoS attacks as the primary communication entities responsible for communication between source and destination [10]. In order to gather information and disseminate to receivers at the destination, each Mobile Agent (MA) is allowed to move from and within a given network of hosts. The record of the immediate sender is maintained through the history buffer that MA is provided with, and the history buffer is refreshed from time to time [11]. More so, a packet is received by the agent externally, and the agent is also responsible for checking the source addresses for the valid IP address. The communication request is only sent to the next step in the modeling after the address is validated, but if it cannot be validated, then the communication having a suspicious IP will be blocked. Also, it saves the address in history buffer for future [12].

This research focuses on the major threat could facing network application layer which is DDoS attacks. The aim of the research is to improve the NALFA protection model against DDoS attacks traffics targeting network application layer. Subsequently, this research proposes deploying software agent technologies for enhancing defense model against DDoS attack traffics targeting network application layer. The rest of this chapter presents the research problem, objective, scope, research significance and thesis organization.

1.2 PROBLEM STATEMENT

DDoS attack consider as the major threats targeting network application layer. It is attempt to make web server unavailable for a legitimate user by disturbing the hosting servers and consuming their resource. There are many proposed defense methods of DDoS attack traffics in the literature and some examples are Kong et al. [13], Behal and Kumar [14], and Prasad et al. [15]. However, the concerns of current models are with improving the existing technologies without considering the history of previous attacks. Both cases negatively affect the defense mechanisms. The first case neglects useful information regarding the attack. The second case causes major delay and disturbance to the network traffic.

Several protection methods have been proposed to secure the web application layer against DDoS attacks. Some models include AL-DDoS [16] and FCMDPF [17]. In any case, these models can only identify strange traffic and determine the sources of each IP address that distributes inconsistencies, while also logging those IP addresses and removing any strange transactions.

Among these IP addresses, there are many queries that have a location with genuine users. The turnover of these users is declining. Thus, models for processing this type of transaction are lacking. Therefore, the network application layer needs a traffic monitoring tool that can create productive identification and permanent square for attacker IP addresses, while temporary block can enter genuine flooding to prevent traffic from real users.

1.3 RESEARCH OBJECTIVES

The aim of this research is to enhance the network applications layer protection model against DDoS attack traffics. The following objectives are proposed to achieve the research aim:

To design Protection of Flooding Attacks model that employs a software agent to defense against DDoS attack targeting the network application layer.

To test and evaluate the performance of the proposed model by using CISDDoS dataset and based on the performance metrics of accuracy, precision and sensitivity criteria then compare the results with the related work.

1.4 RESEARCH SCOPE

The study focuses on proposing an adaptive traffic monitoring and control module to network application layer. The module is used to defense against DDoS attacks traffics targeting network application layer. The scope of this work is as follows.

- This study focused only on DDoS attack flooding traffics targeting the application-level of the TCP/IP architecture, primarily based on DNS traffic.
- With regards to Application Layer Flooding Attack, this work focuses on the external threats of them which is DDoS attack flooding traffics that is considered as the most common and dangers types of attack could targeting network application layer.
- Because applying the proposed method to a real system is illegal and costly, evaluation is based on simulation model. Simulation is an essential way in implementation environment, which can be used to analyze the network traffics related problems under DDoS attack, and network topologies with low cost.

- Since it is illegal and expensive to implement the proposed method in a real system, the evaluation relies on the simulation model. Network research mapping is an important method that can be used to analyze network-related issues at a much lower cost than DDOS attacks, cross traffic, and topologies.

- In this work, the C# which is available on visual studio 2013 is a well-known programming language is used to build network systems. C# programming language covers a large number of network applications including protocols, network types, network elements, and traffic models. Therefore, C# is used for simulating the proposed work.

- The dataset used in this work is testing dataset services to simulate a web server. CICDDoS 2019 is a labeled flow-based dataset. It is developed primarily for the evaluation purpose of IDS and IPS. It is used to generate a large number of DDoS attack and flooding traffics to be as a real attack traffics that are attempting to flood the server.

1.5 THESIS ORGANIZATION

The thesis is segmented into five (5) chapters. Starting with Chapter 1 which presents the introduction of the research including an overview, problem statement, objective, the scope of the thesis, research significance and the thesis organization.

Chapter 2 presents the concepts of NAL and major threat facing it, which is DDoS attacks and subsequently gives an overview of DDoS attack flooding traffics. Also, present the software agent. Furthermore, review the defense methods against DDoS attack traffics based on AI techniques. It summarizes some of the most common dataset, evaluation methods and pertinent research conducted in detecting DDoS attacks.

Chapter 3 presents the research methodology including the design of the proposed model to detect and prevent DDoS attack flooding traffics.

Chapters 4 present the modeling and simulations of the APFA model including the results, then measure the performance of the proposed model and compare the obtained results with the related work. Finally, Chapter 5 presents the conclusion and future work.

2. LITERATURE REVIEW

2.1 . INTRODUCTION

Distributed Denial of Service (DDoS) attacks try to flood the Internet and make it unusable for real users. In this type of attack, the transfer speed of the network or the resources are depleted when various packets are sent to a targeted server [18]. DDoS attack programs have been around for a long time, and many protection systems are available to thwart previous attacks from a single source. In this way, the source of these attacks can be easily prevented or repelled by enhanced capabilities [19]. Attackers use these weak targets to launch an attack instead of a lone server, which is not yet strong at this stage due to the remarkable increase in Internet use over the years. In any case, attacks on a lone server can be effectively identified [15].

An attacker controls many Internet-connected PCs before starting an attack. This computer machine is known as a handler, which puts these computers at a weak point. At this point, the attacker installs malicious code or updates other hacking strategies to exploit the weaknesses of these PCs and accept the control over the machines [20]. These weak machines will later be mined and become "zombies." The number of zombies can reach hundreds or even thousands. In this way, huge gatherings of zombies begin to form a "botnet." The size of the botnet determines the level and magnitude of the impact force. A huge botnet is carrying out terrible and severe attacks [21]. There are two basic methods to spread DDoS attacks on the Internet. According to the basic methodology, attackers send malicious packets to victims to confuse a running convention or application (e.g., a vulnerability attack). According to the following methodology, an attacker sends network / application / transport-level flood attacks using one of the following: (I) changing network resources, or capacity availability to disrupt availability. ii. Server resources such as memory, CPU, I / O transmission capacity, and data transfer / information base to disrupt the administration of the real user [22].

DDoS attacks are transmitted by powerful, remote botnet PCs that are widely distributed throughout the network. Attackers send countless traffic or administration requests simultaneously or continuously outside the target system. The target system becomes unusable, gradually reacts, or crashes completely due to attack [23]. Trojan horses, backdoors, or are commonly used to engage zombies. Identifying the first attacker is difficult for defense systems because zombies that

are heavily influenced by the attacker and have a stick use fake IP addresses [24]. In 2009, many zombies used to flood the victim with DDoS attacks, which disrupted the network's administration on popular sites such as Facebook, Live Journal, Twitter, and Amazon [25]. The first DDoS attacks were usually manual in nature, and the attackers had to take a few steps, they had to create Internet zombies, scanning of ports, and send scanners to replace the malware before it was last sent. DDoS attacks from now on, have been mechanized and perfected, allowing attackers to perform all or part of the tools with negligible human effort [26]. Attackers can also impose explicit restrictions on the target, although the rest can be controlled by computer generated automatic tools. These cyber-attackers integrate Trinoo, Tribe Flood Network (TFN), TFN2K, and Stacheldraht, best of which work on Internet relay chat, which mostly reduce Internet relay chat, where zombies can broadcast indirectly without revealing identity. Other means of attack are usually special, in which zombies and controllers pass on legally provided information about their characters [27].

2.2 THE OPEN SYSTEMS INTERCONNECTION NETWORK LAYERS

The OSI Reference Model is a theoretical framework that which provides a description of the telecommunication and networking system autonomously, without the technology infrastructure [28]. With the OSI, data communication is divided into ideal groups of networking roles so as to enhance interoperability in the communication system irrespective of the type of technology, model and vendor as illustrated in Figure 2.1.

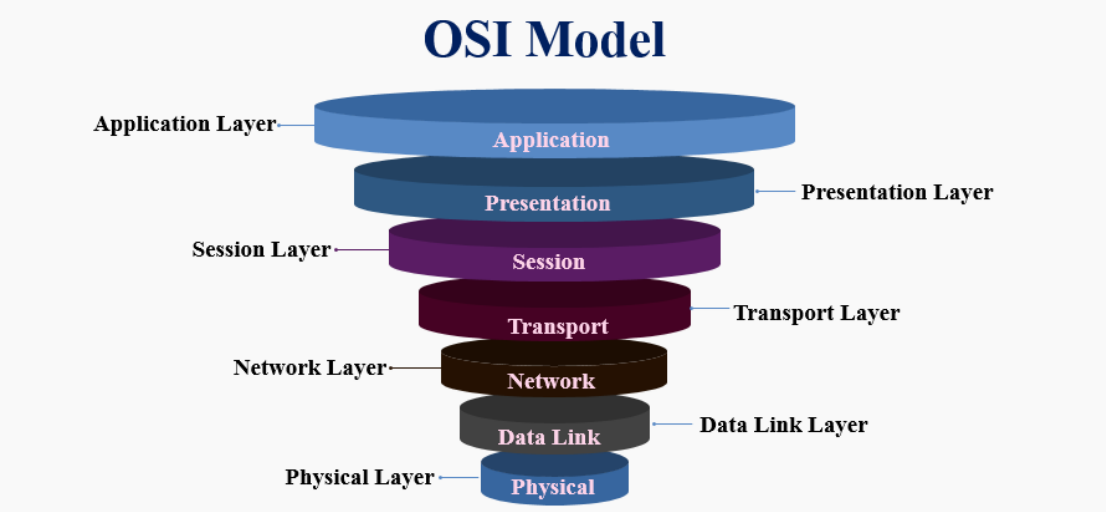

Figure 2.1 :The OSI Model Architecture [28]

The aim of developing the OSI model was to enhance interoperability between vendors while defining the principles of network communications. Nonetheless, the universally used reference framework is the older TCP/IP architecture which is presently being used for internet communication. The following are the key benefits offered by the OSI model [28].

Based on Figure 2.1, the OSI model consist of seven layer starting with the physical layer and ending with the application layer. However, in this work we focusing on the application layer. The Application Layer deals with the networking processes at the level of the application. At this layer, there is direct interaction with end-users to offer assistance, among other distributed information services, for email, network data sharing, file transfers, and directory services. At this topmost layer of the OSI model, there is the identification of networking entities to assist in the networking requests by end-users and determines the availability of resources, synchronization of communication and management of application-specific networking requirements [18]. There is also the identification of constraints such as the ones linked to authentication, privacy, quality of service (QoS), networking devices and data syntax. Among the most frequent Application Layer protocols are the Domain Name System (DNS), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) and Domain Name Server (DNS) [29]. The application layer offers the ways for the generation and reception of data that can be transported on the network.

There are many types of attack targeting the application layer such as DDoS attack, FE flooding traffics, Malware, IP Spoofing, IP Sniffing, Password-Based Attack, and Man-in-the-middle Attack [18]. DDoS attacks is the most dangerous type of attack facing network application layer because there is no need to access to the system, only send heavy traffic to flood the system hence useless for legitimate users.

2.3 DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACKS

DDoS attacks have become a global threat to the current Internet. Moreover, these attacks are of dexterous nature, and ordinary physicians use a similar method of external attacks that is carried out by the previous botnets [30] over a more significant scope than the last. A botnet chain consists of hundreds or thousands of traded hosts (zombies, bots, or slave agents) who attack a particular victim with at least one gate crosser, as shown in Figure 2.2.

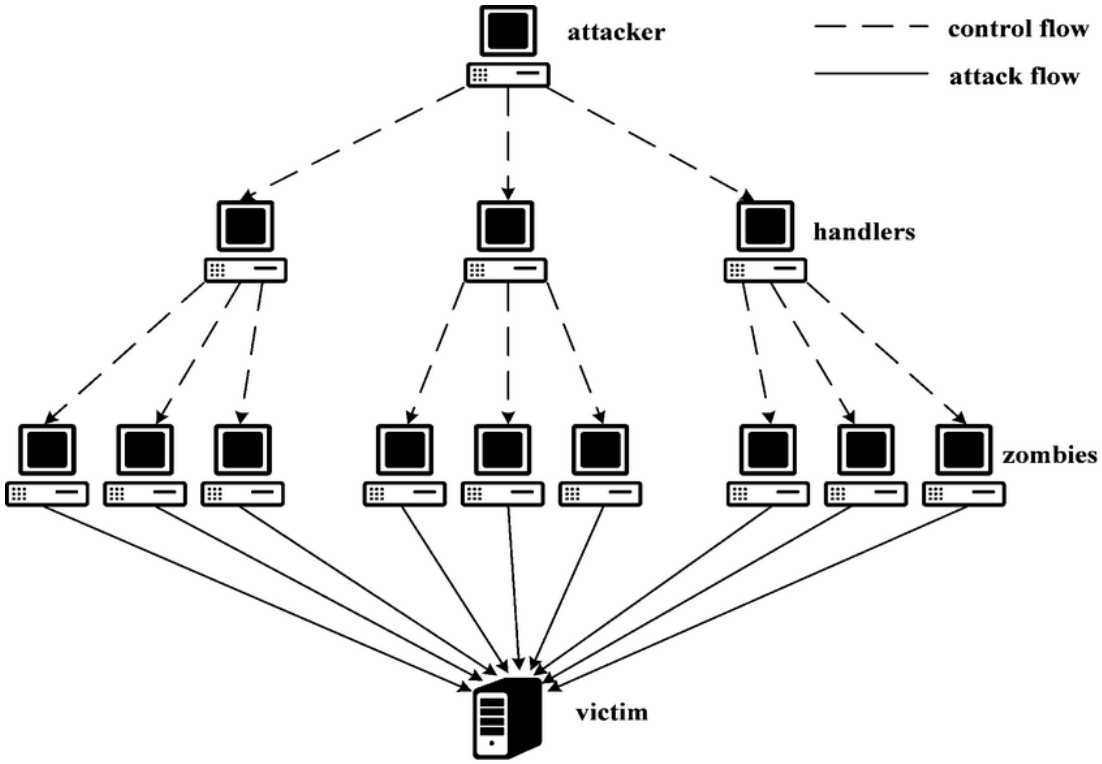

Figure 2.2 : Architecture of DDoS Attack [18]

For attackers, any computer connected to the Internet offers a chance to create zombies without even informing their users. Zombies are lured by worms, Trojans or secondary backdoors by sending an impressive link, e-mail content, or trusted sender address to vulnerable machines [31]. Furthermore, one zombie could provides a limited amount of information. However, the combined turnover of the various zombies that emerge among the end users is huge and therefore runs out of resources.

Low frequency of DDoS attacks are dangerous and difficult to detect because they can usually contain traffic that can be restricted by a particular link [24], [32]. Therefore, effective authentication techniques can lead to a rapid increase in high frequency DDoS attacks. DDoS attacks are currently being launched in the form of links and packet flooding. Such attacks are certainly prevalent on the internet, as attackers certainly know what, where and how to obtain data. Of course, attackers such as Internet conventions, system and web applications can always be exposed to vulnerabilities. Such attacks are planned with thought processes such as collusion exploitation of blackmail, hacktivism (gaining media coverage), financial causes, individual

causes (questions or revenge) and political reasons [33]. Web applications, games, media and programming activities are best suited for these attacks [34], [35].

DDoS attacks come in two kinds: flooding and vulnerability [36], [37]. In the case of flooding attacks, an invader sets an army of zombies to transmit attack packets toward the destination to increase the traffic by an amount that cannot be handled by the victim, and the system of the victim is crashed or brought down [40], [41]. Kaur et al., [42] classified flooding attacks into indirect and direct (via reflectors) DDoS attacks on the basis of the attack method. These attacks are categorized into App-DDoS (application level) and Net-DDoS (transport/network level) flooding attacks [39]., according to the affected protocol level. Flooding attacks, such as UDP, TCP, and Internet control message protocol (ICMP), are classified as Net-DDoS attacks, whereas the flooding of DNS falls under App-DDoS attacks. App-DDoS flooding attacks presented and reviewed in [18], and it shows the incapability of the techniques of network-level detection for detecting such attacks. These kinds of attacks are increasing at an alarming rate, and they are more difficult to detect and produce worse problems in gaining access to particular Internet services (or web servers) in comparison with Net-DDoS flooding attacks. In the case of vulnerability attacks, the invader browses exposed openings in the implementation of the software and takes their advantage to cause the system to shut down or to transmit zombies for subsequent attacks. Such attacks use the exacted performance of several protocols (e.g., DNS and TCP) to damage the resources of the victim's server and stop it from processing requests or events from the sanctioned users.

The Internet holds several vulnerable protocols, insufficient authentication plans, and exposed OS and computer systems that are used to initiate DDoS attacks. The use of prevention techniques is the best means of protecting vulnerable systems, and protocols are required to raise security at the global level [43]. Techniques for attack detection expose any attack after they actually take place, and characterization advances the packets of attack from the usual traffic, e.g., an anomaly and signature-based detection methods. Meanwhile, traceback techniques attempt to prevent the sending of malicious packets at their place of origin and discover the source of the attack in spite of the spoofed source IPs prior to or after the occurrence of the attack. Attack reaction techniques aim to reduce the damage triggered by the DDoS attack during its occurrence. This phase reduces the influence of the attack and maximizes the service quality delivered to the authorized users who are under attack.

2.4 SOFTWARE AGENT TECHNOLOGY

A software agent is a product substance or a mix of equipment or programming element, which can follow up for the benefit of its clients in an autonomous manner. It has numerous helpful attributes like collaboration, learning capacity, and it is reactive and proactive. There are two kinds of programming operator around; there are agents as an assailant group and as a protective group [44]. This work consolidates the agent with a protective or defensive archetype.

The software agents, which offer great benefits in terms of enhancing the quality of the process, are also the integration of specialized expertise and new technology. The use of agent technology is employed in several areas which include information retrieval, user interfaces, filtering, mobile computing, smart messaging, electronic and telecommunications marketplace [45]. There are different ways through which the agents interact with one another within a multi-agent system. Agent clusters within a multi-agent framework are collaborative, competitive, task-oriented and have the ability to provide users with the interface. The use of software agents in DDoS attacks is driven by the fact that they possess some good characteristics which enhance the monitoring of security. Some of such characteristics include fault tolerance, autonomy, information providers, robustness, scalability and dynamic configuration [18], [46].

2.5 DDOS ATTACK DEFENSE METHODS

DDoS attacks are difficult to detect and resolve. In essence, extraordinary DDoS attacks do not have the normal credit to identify them. In addition, due to the widespread nature of DDoS attacks, they are very difficult to counteract or track, and the programming tools that launch DDoS attacks are easy to find. Similarly, intruders can exploit IP address to disguise their identities, thereby gradually complicating the location of DDoS attacks. Finally, machines connected to the Internet do not have an adequate level of security, and the Web is full of certain security conditions. Various professionals have prescribed safeguards to protect victims from DDoS attacks. In subsequent techniques, the system of detection and identity management may regulate its execution cycle based on information collected late [13]. The framework can update your presentation in a number of experiments based on past results. This strategy responds to the AI process, which focuses on obtaining decisions that generate new information [18], [22], and [40]. It provides properties such as parallelism; robustness; and tolerance of faults, inaccuracy, and uncertainty [38]. Artificial

Intelligence based strategies are called artificial intelligence procedures. Artificial intelligence includes advances such as the Bayesian choice hypothesis, perennial procedures, clustering, the multilateral perceptron, direct segmentation, neighborhood models, and classification trees, learning support, and protected Markov models. [46]. Some AI-based recognition techniques, specifically, Bayesian networks fuzzy logic, genetic algorithms, K-nearest neighbor (K-NN) algorithm, neural networks, software agent technology, and Support Vector Machines (SVM), are outlined as follows

2.5.1 Bayesian Networks

Garcia-Teodoro et al. [46] characterized the Bayesian network as a strategy that determines the probabilistic relationship between variables. This method is commonly used to identify attacks along with measurable plans that create a number of favorable conditions, such as the ability to code interdependencies between factors, assess opportunities, and incorporate prior information data and knowledge.

Kim et al. [47] proposed a pocket scoring appropriation that can be identified as a programmed attack characterization, selective packet removal, and means of congestion control. The basic idea of this score is to arrange the packages according to the score per package, which determines the authenticity of the package in the light of the estimation of the characteristics. At this point, a score-based explicit packet destruction measure is performed at the target when the disposal packet score is determined to identify the circuit breaker separation by a hypothetical Bayesian evaluation. The drop limit of the packet emptying measure is gradually modified based on (1) the score allocation of recent inbound packets and (2) the present level of system congestion. Kim gradually did not provide information about the tests to show that the evolution of update events with score logs, total circulation work score, and dynamic evacuation limit can affect response time and the purpose of this package is to propose destruction measures for coordinated synchronized DDoS attacks.

Gonzalez et al. [48] use Bayesian inference prototype to evaluate the reliability of an input switch based on sending circles that do not change the source IP settings. In this strategy, the reliability indicators of the access switches are determined by a router called a judge router that collects all the traffic passing through the access routers. The main purpose of these methods is to use trust

algorithms, trust agreements between managers and routers to discriminate against threatening hostile routers.

2.5.2 Fuzzy Logic

The fuzziness concept is used together with DDoS attack identification methods to emphasize anomalies or attacks on a network [49]. The reasoning is approximated on the basis of fuzzy set theory instead of being accurately obtained from traditional predicate logic [18]. Fuzzy sets and their rules are used to deal with a large number of input parameters (activity rate, CPU usage time, connection duration) that can be unclear in nature and have incomplete datasets [33].

Shiaeles et al. [49] achieved DDoS attack detection with improved time limits using non-asymptotic fuzzy evaluators. The evaluator is deployed on average packet inter-arrival durations. The problem is divided into two parts: detection of the actual DDoS attack and recognition of the IP addresses of the victims. The former task is accomplished by using strict, real-time boundaries for DDoS attack detection. The latter goal is achieved using comparatively lenient constraints, which identify the IP addresses of the victims promptly, thereby starting added anti-attack applications on the affected hosts using the arrival time of the packet as the primary statistic of DDoS attack detection.

2.5.3 Neural Networks

Neural networks are introduced as an alternative method that prescribes the guidelines that follow them, as evidenced by the user's precursors. Neural networks are adequately prepared, forward-looking, and are common in emerging networks that prefer outcomes rather than basic signaling methods [18].

Braga [54] suggested choosing a mild detection measure for DDoS attacks. It acquired a set of attributes for DDoS attacks using independent neural computational design to identify attack spikes with SDN traffic data work. Meanwhile, Niyaz et al. [55] a multi-vector, in-depth DDoS attack framework within SDN. However, it is difficult to identify these low percentage attacks using these procedures as it seems logical that there will be more traffic from the network at the end of the crash. Meanwhile, long-term DDoS attacks must be established in connection with victims. Otherwise, it does not endanger the assets of the network or the framework. The procedure

used for progress recognition uses unexpected packages in the network and can understand the visible distinction between authentic and offensive traffic. You will find redesigned plans that describe DDoS attacks and trace them in a long-term traffic sequence.

Chambers et al. [56] proposed a natural language according to the invention that prepares an NLP modeling application to detect DDoS attacks using only web-based media only. Private networks usually report attacks late. For example, a detection framework that uses public information may respond better to a comprehensive attack on some administrations. NLP models examine whether online media is used as a perceived part of a network's administrative status. For this work, two learning models have been elucidated: neural network and an incompletely labeled LDA framework. In addition, the material-dependent model enables society as a whole to thoroughly evaluate the response to current network attacks and thus 'organize' a number of perceptions. Our plan is the first full plan to identify DDoS attacks with widespread results, but at the same time evaluate how and when the general community cleans up for service outages. The models are typical: they examine the management of the largest Twitter DDoS corpus to date and assess society's overall response based on the return of the scientific community.

2.5.4 Software Agents

Kotenko and Ulanov [57] point to a agent-based methodology and software which relies on the OMNeT ++ INET system designed to provide appropriate safeguards that can be placed on the Internet to support network attacks. The proposed approach is the cybernetic neutralization of “malicious guys”, security devices, and the connection between different specific groups. The essential parts of the production framework are discussed. One of the tests for protection against DDoS attacks is described.

Junija [44] proposed a multidisciplinary structure to identify, secure and track the source of DDoS attacks. This locates a source of DDoS attack, but it needs some experts to get the best results. Kesavamoorthy & Soundar [58] propose another way to differentiate and protect against DDoS attacks with an independent multidisciplinary framework, in which experts use particle swarm optimization together to create robust communication and dynamic dynamics to have. DDoS attacks are recognized by several interconnected companies and the helper is trained on a new

attack. The recommended framework at cloud level can protect against various types of DDoS attacks with 98% accuracy.

Singh [59] introduced a common conspiracy, specifically based against DDoS attacks that recognize and prevent such attacks within ISP boundaries. The important effort is led by experts and expert network in all ISPs. The security framework works by analyzing the incoming traffic on the margin change and identifying the term of DDoS attacks. Experts share information about the attack with nearby ISPs to achieve centralized security. On the positive side, steps were taken to map the display of the protective frame, and the effectiveness of the protective frame in terms of illuminating the frame without such a defense system was evaluated.

2.6 EVALUATION METHODS

Evaluation studies examine defence frameworks by presenting them qualitatively and quantitatively. Some cases identify test situations that rely on specific assessment measures and measures. This section covers ranking measurements at different angles according to measurement measurements.

2.6.1 Performance Metrics

Evaluation methods is an important stage have to take into account in designing the network systems. In this subsection, the most common performance metrics in the filed of information security have been illustrated bellow:

True Positive Rate (TPR): True positive rate is the percentage of attack instances that are identified and reported correctly as attacks. This metric is useful for validating the effectiveness of the detection tool [18], [19], 27] and [33]. It can be considered the same as the recall (or sensitivity) parameter, which stands for the retrieved fraction of relevant instances. Equation 2.1 is used to express DR:

$$TP = \frac{TP}{TP + FN} \tag{2.1}$$

True Negative Rate (TNR): True positive rate is the percentage of instances that are classified as legitimate.

False Positive Rate (FPR): False positive rate is the number of legitimate instances that are classified as attacks. Its goal is to measure the effectiveness of the system in distinguishing fake and legitimate requests [58]. It can be stated over the ROC curve (DDoS DR v/s FPR graph) to demonstrate the possible trade-offs between TPR and FPR [45], [51].

False Negative Rate (TNR): The true negative test result is one that does not detect the condition when the condition is absent.

2.6.2 The Precision

The precision metrics is represents the number of attack that are correctly detected and reported, divided by the total number of reported attacks. However, it is suitable for measuring the quality of the attack detection system [18], [19]. Furthermore, the precision can be calculated as the bellow Equation 2.2 as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

2.6.3 Accuracy

This is the proportion of correctly classified instances. Moreover, it is represent a basic parameter used to compare the performance of various detection methods [18], [19], [20], [57] and [58] and can be stated as in Equation 2.3, where TN is the number of TNs.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.3}$$

Whereas, Error Rate: represent the reflection of the detection system's accuracy, and it can be calculated as the following formula 2.4.

$$Error\ Rate = \frac{FP + FN}{TP + FP + TN + FN} \tag{2.4}$$

Finally, F-score: it represents the reflection of the trade-off between two metrics: precision and recall. The range of its values is between 0 and 1. High values are used to indicate remarkable

performance in terms of precision and recall [18]. Moreover, it can be calculated as in Equation 2.5.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2.5)$$

2.6.4 System Sensitivity

The sensitivity also is the most common performance metric that used to evaluate the performance of the defense methods against the network application layer flooding attack. It is suitable for measuring the quality of the attack detection system. The sensitivity of the system can be calculated as in Equation 2.6 as follows.

$$Sensitivity = \frac{TP}{TP+FN} \quad (2.6)$$

It is used to evaluate the performance of the system as in [18], [19] and [57].

2.7 THE TESTING DATASETS

A number of real datasets are available publicly and have even been widely used for DDoS-related research [18]. A summary of these real datasets is as follows.

2.7.1 The CAIDA DDoS Dataset

This dataset contains traffic reports of a DDoS attack lasting approximately 1 hour. This attack aims to reduce processing resources, and IP addresses have and pseudonyms PINs. In addition, the usefulness of this data is limited by the fact that their payment fees and unsupervised traffic are excluded from the database for security reasons. This data can be used for low levels of privacy and high levels of DDoS attacks. The attack segment in the analysis is the CAIDA dataset, and the normal traffic component is provided from the SSE group information. Normal traffic and attacks are characterized by the use of an open source tool called Konstanz Information Miner version 3 [62].

2.7.2 DARPA DDoS Attack Dataset 2009

The DARPA dataset from 2009 refers to a mixed dataset that can be reactivated for real network attacks and internet traffic. It took 10 days, ie from 3 November 2009 to 12 November 2009. The database is approx. 6.4 TB and is divided into a large number of pcap files measures 954 each. The traffic provided basic information about SMTP, HTTP and DNS. Attacks are considered large network attacks that include DNS worms, DNS worms, and DDoS attacks. Worm and DDoS attacks are defined to show specific multiplication effects. This strategy focuses on the latest DDoS attack-based data obtained from the MIT Lincoln Laboratory. Captured Traffic and SYN Flood DDYNS attacks have base traffic on a target. Enter approximately 100 unique IP addresses as the source of DDoS traffic. These hosts were used to carry out a malware DDoS attack on a nearby target [63].

2.7.3 The Coburg Intrusion Detection Data Sets (CIDDS)

Coburg Intrusion Detection Data Sets (CIDDS) are a streamlined subset of datasets that have been analyzed. It was mainly done for IDS and IPS evaluation. The database contains OpenStack and External Servers traffics. The top positions for Attack ID and Attack Description are not included in this review as they provide more information on performed attacks. Consequently, these functions did not contribute fundamentally to the research. Approximately 153,026 events from external staff and 172,839 events from OpenStack server data were collected for evaluation. All events in the database were named or marked for format, attacker, horror, suspicious and obscure class [64]. In addition, the experts evaluated the presentation of their proposed approach with publicly available CIDDS data. A proper discussion of the commonly used DDoS databases was provided, a summary of the dataset is given in Table 2.1.

2.7.4 CICDDoS 2019 Dataset

CICDDoS 2019 is a dataset has been created in university of New Brunswick. It is used to test and evaluate the performance of the proposed system that have been design to defense against DDoS attack and related flooding attack. Furthermore, it contains benign and the most up-to-date common types of flooding attacks, which resembles the true real-world data (PCAPs) [65]. It also includes the results of the network traffic analysis using CICFlowMeter-V3 with labeled flows

based on the time stamp, source, and destination IPs, source and destination ports, protocols and attack (CSV files).

Table 2.1 : Summary of DDoS Datasets.

No.	Ref.	Dataset	Scope	Traffic type	Layer
1	CAIDA, [62]	CAIDA	DDoS	ICMP	Network
2	DARPA, [63]	DARPA	DDoS	HTTP, DNS and TCP	Application, Transport
3	CIDDS, [64]	CIDDS	DDoS	HTTP, DNS	Application
4	UNB [65]	CICDDOS	Flooding Attack	ICMP, DNS, HTTP, TCP and UDP	Application, Transport

2.8 ANALYSIS AND DISCUSSION

The overview of DDoS attack flooding traffics characteristics help to develop a good intuition about those types of attack and how they come to existence. Therefore, to allow for systematic studies of DDoS attack targeting network application layer in particular, such a formal definition is essential to the defense against those types of attacks and comparing the performance of different defines strategies. Recently, observed a big problem and serious nuisance for network security could be a DDoS attack.

This chapter presents a review of various methods to detect and prevent DDoS attack, according to the classification of artificial intelligence techniques that are feasible at the OSI layered model. The aim of this review is to provide guidelines for developing improved DDoS defense methods, strategies and integrating effective solutions. The OSI network layers model have been demonstrated with explaining the role of each layer in the OSI model with emphases on the network application layer. Furthermore, the motivations that prompt the attackers to send a large number of fake requests to flood the network application layer and make it useless for legitimate users have been highlighted.

Related research activities in this area focus on specific parts of DDoS threats and protection controls, such as attacks, security techniques, assessment strategies, and test datasets. Various procedures have been created, recommended, and implemented to protect network application layers from DDoS attacks; Most of them are unable to identify the actual traffic and harmful consequences in terms of creating false positives and positives. This work has been reorganized according to the most recognizable strategies and features that rely on AI methods for countering DDoS attacks with a focus on the network application layers.

The model of Zhou [16] is able to efficiently detect and prevent DDoS attack and block the abnormal traffics. However, this model arbitrary block the abnormal traffic without considering that of the legitimate traffic in which the abnormal traffic contains legitimate and illegitimate traffic. Additionally, according to the literature review, the agent architecture is proposed to defense against DDoS attack. The results show that the architecture is efficient and flexible in defending against DDoS attack. Furthermore, it has the adaptive mechanisms to distinguish legitimate traffics from attack traffics. Based on the literature, the most common evaluation criteria of the DDoS defense methods are quantitatively measured, therefore accuracy, precision and sensitivity have been used in this work.

2.9 CHAPTER SUMMARY

This chapter attempts to achieve a clear view of the DDoS attack problem and the numerous defense solutions that have been proposed in the literature. This comprehensive picture of the DDoS attack problems clarifies the view, thereby allowing the work to find possible solutions to the DDoS attack problems. Among these defense methods and according to the discussion above, the agent technology is found to be most suitable to combine with the reactive adaption for defense against DDoS attack targeting network application layer.

3. METHODOLOGY

3.1 INTRODUCTION

As mentioned earlier in the previous chapters, the DDoS attack is a serious problem for many individuals and applications operating on the Internet today. The attack is not only hard to identify, but its methods also change constantly [10]. Furthermore, new ways of attacks are emerging at a rapid rate, as a reactive response, new protection techniques are needed to be developed. Moreover, the overview of DDoS attacks characteristics helps to develop a good intuition about those types of attack and how they come to existence. Therefore, this thesis offered a thorough and detailed review of various methods to detect and prevent DDoS attack.

3.2 RESEARCH STAGES

This work proposed the Defense of Flooding Attacks model which has the ability to controls the network traffics and defense against DDoS attack. Then, this chapter describes the research methodology of the proposed model. The methodology consists of four parts, which are the identification of the research problem, experimental research planning, the solution model Protection of Flooding Attacks, and testing and evaluations the performance of the proposed model. Figure 3.1 shows the research methodology. The following subsections illustrate the research methodology in the four parts.

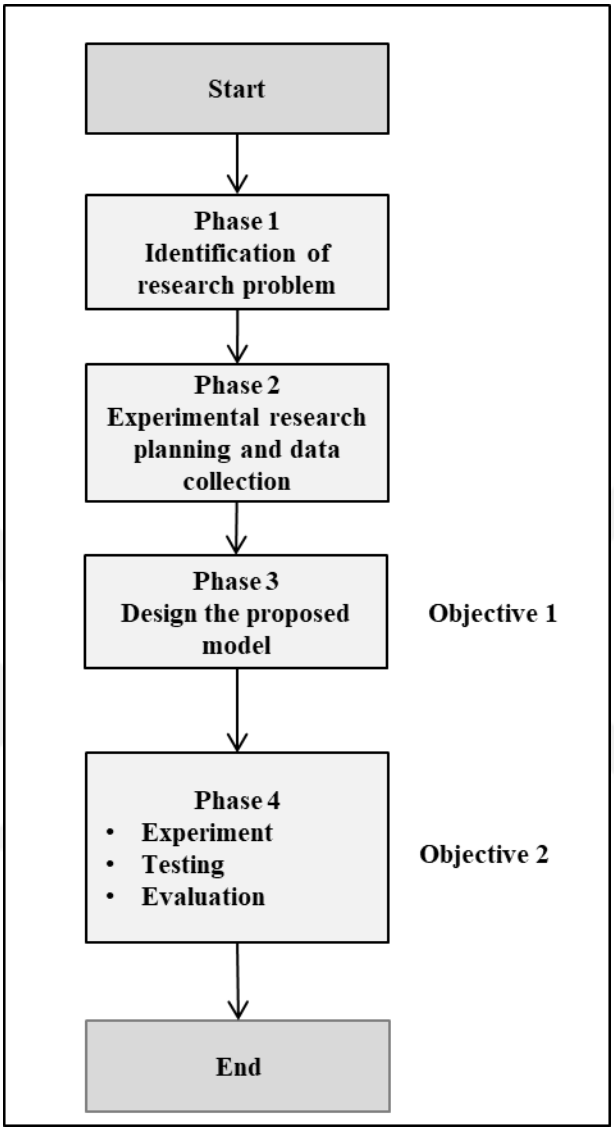

Figure 3.1 : Research Phases.

3.2.1 Identification of Research Problem

Recently, a big problem of DDoS attack is observed which implicate serious nuisance for network security. The advent of the science and technological era created a threat of such attacks for a whole range of public administrations and applications using the Internet [17]. Businesses can have colossal financial accidents during a short period of disruption in the services of administration. Analogous to a DDoS attack is the abnormal flooding traffics, which is a particular service is assessed by many legitimate users concurrently, which results in the denial of service

[18]. Active overload is a problem associated with both cases, where the CPU, available data transmission, and memory fluctuate to real users, who then request limited availability [33].

The DDoS attack involves a large number of sources that the messenger attacks against the website server expect to deny the management of real users. Before finally sending an attack, the attacker is responsible for some computers via the internet and puts them in a weak position. At that time, the attackers begin to exploit the flaws of these computers by introducing malicious code or some other method of hacking to target the machines. The number of these defenseless machines also called zombies. These can be hundreds or thousands. Such zombie collections allow the formation of a botnet. The size of the botnet determines the size of the attack; a larger botnet is usually associated with more extreme and unfortunate attacks.

The issues above result in those individual entities easily fall behind in the development of the field and, thus, resulting in inadequate protection against DDoS attacks. The main challenging issue in detecting DDoS attacks based research is to build an effective model in identifying a DDoS attack using necessary and valid features. Hence, this thesis is meant to produce an artifact that addresses these issues.

Various procedures have been proposed, created, and implemented to protect website servers against DDoS attacks; in any case, masking malicious traffic behind the actual traffic remains a typical scourge for these protective models. Most of them do not clearly recognize the true traffic and harmful imitation of negative and false positive creations. Recently analysts have called specialists for a DDoS attack. For example, it was developed by professional carriers as an element of protection against DDoS attacks [16].

The aim of the model which is designed in the current study is to provide protection for application layer against DDoS attack. In flood attack, DDoS attackers are able to make packets that resemble normal packets, however, the traffic it generates is different from that which occurs in normal circumstances. In addition, the rate at which resources are used in the case of an attack is different from that of a normal situation. Therefore, their certain characteristics can be used to determine each kind of attack; these features are associated with the use of resources and the traffic that occur during an attack.

3.2.2 Experimental Plan

One of the most relevant steps involved in research is the planning of experiments because, without the experimental plan, the researcher has no guide. When there is a plan, the researcher is able to validate the results of the experiments. The first step in this research involves problems identification, and afterward, the research objectives are formulated in a way that addresses the problems that have been identified. The research objectives are formulated based on the review of past studies. The experimental plan involves selecting a proper dataset that can be used in evaluating the solution model. The data should be able to demonstrate real-time processing to the model implementation. This ensures the quality of the performance evaluation.

3.2.2.1 Datasets Description

Benchmark datasets are required for testing, evaluating and validating assumptions for the researcher's methodology. However, the hardest issue for working on DDoS attacks is to find suitable datasets [14]. It is a considerable challenge since the availability of such datasets is very rare. Subsequently, there are several types of security datasets such as CIDDS, DARPA, KDD and CICDDoS 2019 available to be utilized [14]. These datasets can be utilized in the detection of several types of attacks as well as for systems assessment. However, existing datasets also post additional limitation as most of them are outdated. Due to numerous new types of attacks and security technology for detecting DDoS attack flooding traffics-based network anomalies, there is a need to find a new and suitable dataset to test and evaluate the proposed model. However, CICDDoS 2019 is a conception used for the creation of datasets that will be used for evaluation of anomaly-based network intrusion detection systems [64].

Therefore, the primary objective of using CICDDoS 2019 in this study is that it is current and customizable. The additional program of C# programming language is used in a virtual environment to regenerate customized datasets that can be used in this work. Basically, the CICDDoS 2019 dataset was chosen for the following reasons:

- It is most recent in 2019.
- It is available online and free.
- It can simulate real-time processing because.

- It has the attributes of several types of Flooding attack.

3.2.2.2 Implementation Platform

C# programming language is regarded as a general-purpose and subject-oriented modern programming language that is available on visual studio 2013. C# is a high-level language which has some commonalities with Java and C++ and to some extent, languages like Delphi, VB.NET, and C. Also, the C# programs are object-oriented and made up of a set of standards in categories containing methods, and in these methods, program logic and instructions executed by the computer are contained [66]. C# is designed and developed jointly by Microsoft and ASP.NET platform. Meanwhile, a wide range of software applications is developed using this programming language on ASP-NET. The C# is active in the websites, mobile applications, office applications, desktop applications, games, web applications, etc. [66].

3.2.2.3 Data Preparation

The following subsections describe the steps of data preparation and collection.

- Real-time Data Processing

The processing of data in real-time involves executing data within a short period of time and providing near-instantaneous output. Once the data is inputted, the processing begins, and it is required that the data input be continuous so that a continuous output can be provided.

For the traffic to be processed in real-time, real-time frequency vector $RFV = (favg^1, favg^2, \ldots, favg^m)$ was created. The amount of resources like the images and web pages in the website is denoted by m. Each RFV item represents the average frequency of one resource being visited. Its value is calculated by Equation 3.1 as follows. Where $1 \leq k \leq m$.

$$\begin{cases} favg_n^k = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{te_i - te_{i-1}}\right) \\ favg_{n+1}^k = \frac{1}{n+1}\left(favg_n^k \times n + \frac{1}{te_{n+1} - te_n}\right) \end{cases} \quad (3.1)$$

In Equation 3.1, 〖favg〗_n^(k) denotes the average frequency for each resource, and is defined as the average reciprocal difference between every pair of successive arrival times te. The parameter n is dependent on the frequency of visits. The algorithm's complexity is denoted by O (C), where C is constant. In addition, the algorithm has a spatial complexity of 2m with current 〖favg〗_n^(k) and the last arrival timete being recorded by two parallel vectors. Through the use of Equation 3.1, a dynamic description of the visiting status of web users can be given. Nevertheless, in real-time, there are two problems that need solutions; 1) how can the items in RFV be localized? and 2) how can the length of the RFV be dealt with?

The items in RFV can be localized by making the complexity of identifying a particular item in a vector O (n), where the vector's length is represented by n. A server often contains tens of thousands of 'Get' requests. Thus, the requirements of real-time processing may not be met by this method of localization. In a large scale website that has massive resources, the length of RFV is dealt by obtaining the statistics on the traffics. The reason for this is that a large number of the resource will be visited; the length of RFV is quite large. Therefore, it is almost impossible to perform real-time processing. The traces of real-world traffic have been analyzed in past studies, and a conclusion was drawn that 90% of requests came from less than 10% of the most popularly visited web pages. Thus, models can be mined from users' visit history without requiring unit digit level accuracy in the algorithm. Thus, the length of RFV can be greatly reduced by not considering the records that have an average frequency of small values. For instance, a website that has 34,000 resources, the length of RFV can be reduced to less than 34000.

- Traffic Models

In this study traffic models are derived from web traffic by building a model set V { v1…..vp}, which includes the traffic models (tModel) and the models between models (bModel). In order to get tModel, it is agreed that a self-similarity distribution is followed by internet traffic. Based on the observation of the traffic, it was found that the traffic was a period of a day. In the current study, the Model Checking Period (MCP) is set at 24 hours, while the RFV is selected as a tModel for the 24-hour traffic.

The initial state of V is ∅. Subsequent to the addition of first RFV to the model set, the checking of every successive RFV in a MCP should be done so as to ascertain if current RFV is a new

tModel. Equation 3.2 is used in calculating the coefficient of the correlation between the current RFV (v) and tModels ({ vi|1≤i ≤p}) in the model set:

$$r = |\ cos\ \theta\ | = \left| \frac{v\ \times v_i}{||v||\ \times ||v_i||} \right| \tag{3.2}$$

Where the parameter θ is the included angle. The range of the correlation coefficient value r is within 0-1. In an event that r is close to one, the two vectors are more correlated and they may belong to the same model. Conversely, if r is close to zero, the current RFV can be considered as a new model within the set. In to make a decision, Model Similarity Threshold (MST) is set.

For the web traffic which follows the self-similarity distribution, the stability of V tends to be achieved after the mining of models occurs for a given time frame. This shows that the duration has passed the Self-Similarity Period (SSP). An example will be the MCP and SSP of the CISDDoS traffic which are set at 24 hours. Then, V achieves stability when p ≤24. The MCP can be adjusted, while the use of a smaller time frame can be employed for the models. The RFV is the approximation of a real-time model in the web visitation traffic. Nevertheless, the traffic of the network is noticeably unstable, and it is almost infeasible to present the model. The MST can be changed so that the results of the model matching processes can be adjusted.

For the derivation of the bModels, real-time models can be constructed by the RFV. However, detection cannot be adequately performed by these tModels. In fact, one of the key temporal characteristics of web security is the users' visitation. There are various tModels available for all sort of temporal patterns. In this study, a different kind of model known as bModels is introduced.

The production of the bModels is done in a dynamic way, thereby leading to the dynamic growth of the diameter's matrix. Therefore, the SSM is a dynamic spanning matrix. The event of v_i emerges prior to v_j which is denoted by v_i→v_j, and therefore the following properties: (1) items along the SSM's diagonal line represent the number of instances for each tModel; (2) each item in the up-triangle of the SSM represents the number of instances v_x→v_j after v_j the production of (x ∈(1.i)); and (3) each item in the down-triangle of the SSM indicates the number of instances v_i→v_x after v_j has produced (x ∈(1.i)).

For each v_i→v_j, the window w (l) is set, wherein the window length is denoted by l. Figure. 3.4 shows that, for any sequences inside the window, a (bModel v_i→v_j, is defined regardless of the presence of other models between them. Therefore, it can be seen in Figure.3.4 that the sequential relations are obtained $\alpha \rightarrow \eta$. $\beta \rightarrow \eta$. $\gamma \rightarrow \eta$. This gives three instances of bModels, whereby the number of appearances made by each bModel in the SSM is recorded, and the new bModel is pushed into the window. Therefore, bModels could be dynamically obtained.

To evaluate each tModel and bModel produced in the proposed system, SSM is used to compute the possibility of each bModel by using Equation 3.3.

$$p(v_i) = \frac{\sum_{i=1}^{|V|} S_{ij}}{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} S_{ij}} \tag{3.3}$$

For each v_i→v_j, its support can be calculated by Equation 3.4, which denotes the possibility of v_i→v_j having the same sequences originating from v_i:

$$P(v_i \rightarrow v_j | v_i) = \frac{S_{ij}}{\sum_{i=1}^{|V|} S_{ij}} \tag{3.4}$$

Moreover, the confidence of the bModel can be calculated by Equation 3.5. This denotes the possibility of the sequencev_i→v_j:

$$P(v_i \rightarrow v_j) = \frac{S_{ij}}{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} S_{ij}} \tag{3.5}$$

3.2.3 Protection of Flooding Attacks Model

There are many methods proposed to defend against DDoS attack as discussed in the literature but the performance is still not good enough. This work proposes the Protection of flooding attack model to protect network application layer against DDoS attack. The propoed model monitors and controls both of DDoS attack and abnormal traffics. It consists of four stages, which are Abnormal

Traffic Detection Stage (ATDS), DDoS Attack Detection Stage (DADS), Traffic Control Stage (TCS) and Kalman and Bloom Filters Stage (KBFS) as described in the following subsections.

3.2.3.1 Abnormal Traffic Detection Stage (ATDS)

The main module in the APFA model is the Abnormal Traffic Detection Stage (ATDS). The ATDS is the Abnormal Traffic Detection Stage (ATDS), which can be a product application or device that can filter frame exercises or network traffic to detect strategic violations or harmful exercises and report to a management station. This cycle helps to differentiate the intrusion because the potential of the intruder's behavior is unique compared to the actual user's behavior. Its use is largely close to other security measures that activate countermeasures [16].

This stage was designed to filter analyses and gradually distinguish abnormal traffic in real-time. The presence of this stage is mainly in the frontend sensor. It dispels the fundamental skepticism about how standard traffic behavior relates to the use of a typical framework, or at least that the main deviation from common behavior may be interesting. Therefore, the goal is to detect any unexpected change that is occurring. The fundamental emphasis behind this module is that when the modeling is completed, the output of the model can be related to the typical behavior of the dog and a strange understanding can be gained by distinguishing between the output of the model and the observed behavior [16]. This signature is calculated as a signal that shows where it ended up as an identifiable component of DDoS attacks, and then helps separate whether this could be DDoS traffic or unusual trading. There are many fruitful applications based on such ideas in online traffic analysis.

The main purpose of this module is to detect sudden changes in DNS GET requests, i.e., anomaly detection, sent to the front-end sensor. This module does not take any action if no anomalies are detected. If abnormal information is detected from the incoming DNS traffics, an "attention" signal is sent to the next phase (DADS), which gives further analyses for the traffics. Several steps are taken before sending an attention signal, these are described below.

The first measurement is to analyze incoming traffic. This can be done in many different ways but the APFA model predicts traffic intensity by using an Auto Regression (AR) model [16]. In regression, previous values have an effect on future values; therefore, the AR model uses previously observed traffic to predict the change of traffic intensity in the future.

Initially, the DNS GET traffic stream is monitored. A time-series {y_(1), y_(2), . . . , y_t} is formed by the traffic intensity which is studied in constant time intervals. The traffic intensity is calculated in this thesis “by the total number of packages received in a time interval” [16]. The traffic intensity is predicted from earlier observations with the help of the Shannon entropy [14]. If major changes are detected, it can potentially be network application layer DDoS attack. The AR model that predicts the current traffic intensity can be calculated by Equation 3.6.

$$y_1 = \sum_{k=1}^{p} a_t^k \; x_{t_1 + e_t} \tag{3.6}$$

The variable y_(t) is the prediction of x_(t), which is the observation value at a certain time t. The variable a_t^k is a “stationary model parameter”, which means that it does not vary when time changes, and e_(t)is the observation error [16].

Secondly, at a certain time t, the difference between the observation x_t and the prediction y_t gives the residual or model error x_t , which can be seen in Equation 3.7. From that specific residual under time t, a standard deviation σ_d^2 can be calculated, as seen in Equation 3.8.

$$d_t = |y_t - x_t| \tag{3.7}$$

Thirdly the standard deviationσ_d^2:

$$\sigma_d^2 = \frac{\left(\sum_{i=t\_p}^{t} \left(d_t - avg\left(d_{t\_p}^{t}\right)\right)^2\right)}{p} \tag{3.8}$$

Now, a threshold was seen in Equation 3.9 and Equation 3.10 can be made for determining if the traffic is abnormal or not. If d_t is higher than kσ_d^2 abnormal traffic is detected and an attention signal is sent to the DADS for further analyses. In the opposite case when no abnormal traffic is detected, the ATDS module sends a dismiss signal to the DADS module which inactivates itself. The constant k adjusts the sensitivity of the threshold and is not set to a specific value.

$$d_t > k\, \sigma_d^2 \tag{3.9}$$

And

$$\sum_{i=1}^{n} favg_i Log(favg_{i)} \tag{3.10}$$

3.2.3.2 DDoS Attack Detection Stage (DADS)

The DDoS Attack Detection Module (DADS) is the second unit of the propoed model. It has many stages of preparation. Said conversation is presented by ATDS; nevertheless, DADS is the second part of the proposed model in this category. DADS is a policy to determine the reliable origin of bundles on the Internet. DADS recognizes the current source of presented sets in the section. While there are a number of different response plans, DADS provides an incredible highlight for resources after IP bundles, as well as the contrast for getting better tracking skills in others. In addition, it can versatile change its control speed, participating router's load via a flexible flow-based marking scheme [16].

In addition, the DNS agreement follows a policy of tripartite coalition. It is also known as "SYN-SYN-ACK" because TCP sends three messages to schedule the start of a TCP meeting between two computers in a typical DNS format. In a DNS flood, DNS clients, such as a web browser, work with an application or a server to send DNS requests. The request can be "GET" or "POST". The essence of the exploit is to force the server to designate the number of devices that can be expected to serve the offense under these circumstances and to deny credible user's access to the server resources. An evaluation of both the reproduction and the current implementation of a frame, including a sealing concealed bearing position, enables that frame to provide a satisfactory tracking result in any event that a load important is imposed on the server. This framework was inspired by DDoS protection. In addition to tracking DDoS attack packages, this provides additional assistance to improve the isolation factor used to attack the traffic [16].

The activation of the DADS occurs upon arrival of an 'attention' signal from the previous module. Afterward, each incoming source IP address is traced by the DADS, while the average frequencies in the related RFV are recorded. Within the context of a huge amount of traffics, the average is regarded as the probability each required webpage. In the actual sense, RFV processing is to a

large extent efficient for a large amount of traffic and is also able to produce a sequence of traffic models. For tModels, Equation 3.3 was used in computing their probabilities. On the other hand, Equation 3.4 and Equation 3.5 are used to calculate the support and confidence values, and (confidence-support)/2 are used in place of the probability. A comparison of the current model with normal traffic models within the model set is then carried out by the DADS, and if it is found that the current model's probability is above the assumed threshold, then this abnormal traffic will be regarded as an AL-DDoS attack. In any event, that the current model demonstrates a probability which is smaller than the assumed threshold, then this abnormal traffic is regarded as normal traffic.

Moreover, in order to differentiate the DDoS traffic from normal traffic for each IP address, the calculations for entropy values on each tModel or bModel are done. The purpose of these calculations is to provide a description of the distribution of incoming sources and the target URLs. Next is to determine the source of each IP address that sends anomaly traffic in the DADS module. Then, it will send to the TCMS for further analyses.

3.2.3.3 Traffic Control Module (TCM)

This work contributes the Adaptive Traffic Control (TCM) Stage. The TCM is agent in which it is deployed to perform the adaptively decisions. The autonomous capabilities of the agents are supported by facilitating the selection of tasks based on states of beliefs. These beliefs help the agent to make decisions on the course of actions that are required to complete the tasks. The tasks involve monitoring the behavior of the incoming traffics data and controlling the flow of the traffic.

The TCM agent has a reactive adaption in which it performs adaptation to the traffic through implementing three functions which are the anomaly traffic identification ati function, anomaly traffic diagnosis atd function and anomaly traffic handling ath function. These functions process according to the values of three pre-existing parameters which are traffic attack behavior, normal traffic intensity and history log of IP address. Figure 3.2 shows the architecture of the proposed ATC and the related adaptive functions.

The belief constituents include information about traffics in the normal case as well as in the abnormal case. Desires also referred to as goals, are reflective of what the agents intend to achieve. The agent can create desires or goals explicitly or generating them during runtime. However, in

the TCM agent, the desires are pre-determined by the corresponding tasks which are explained in the following. Lastly, intentions are interwoven with plans, which are sequences of actions structured towards achieving the goals if there is a means of achieving them.

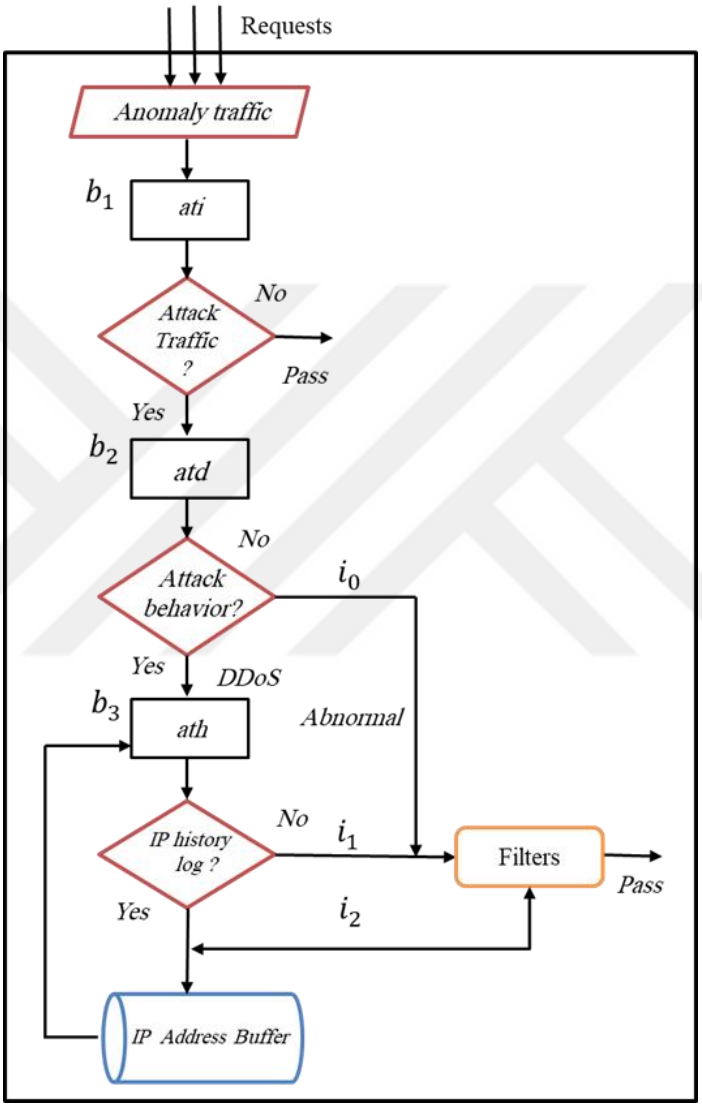

Figure 3.2 :TCM Architecture

Additionally, the agent's goals are differentiated form plans. There may be several plans prepared for achieving a goal, so that is one plan fails, the agent considers other plans according to the reasoning cycle. In a case whereby, there are multiple plans to achieve the goal, there is a selection function so that the most adequate and appropriate plan can be selected, rather than following the random selection approach.

Let , B, represents the network traffic parameters: normal intensity, b_0; traffic intensity, b_1; traffic behavior, b_2; and IP history log, b_3. The agent beliefs motivate set, D, to accordingly reacts based on the traffic conditions. The desires are filtered via three functions: ati, atd , and ath. The desires are translated to intentions, I , of filters, i_0, block, i_1 , and lock, i_2 traffic actions. The i_0 represents a temporarily filtering of traffic by randomly dropping network requests. The i_0 is applicable in the cases of the Abnormal traffics detection. Whereas, the i_1 represents a temporarily blocking to DDoS zombie network requests. The i_1 is applicable to cases of DDoS zombie IP addresses detection. The i_2 represents a permanent locking to the DDoS demon network requests. Finally, the i_2 is applicable for cases of DDoS demon IP addresses detection.

When the anomaly traffic with the source IP address is reached to the agent coming from the DADS, the TCM agent will control the incoming traffic according to the three predefined parameters. In the first stage of an agent, the ati function checks the current traffic intensity with the b_1, in case of that the current traffic intensity more than b_1 that means there is an attack traffic state, in case of the current traffic intensity less than b_1 that means it is a normal traffic state and the traffic passes to the web service.

In the second stage of the agent, and after it determines the incoming traffic is attacked then the second function which is the end and according to the b_2 diagnosis of the type of attack traffic if it is DDoS or abnormal traffics. In case of the DDoS, the traffic is sent to the last function which is ath for further analysis. In case of the abnormal traffics, the traffic is sent to Kalman filter to temporary block some of it. The traffics DDoS or abnormal state is configured according to the Equation 3.11.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}} \tag{3.11}$$

In the last stage of the agent, and after it determines the traffic is DDoS, then, the last function which is ath and according to b_3 separates the DDoS traffic into demons and zombies. Then, the demons' IP addresses are sent to bloom filter to permanent block, and the zombies' IP address is

sent to Kalman filter to temporary block. Finally, the demons' IP address is saved in the buffer IP address for future processes.

3.2.3.4 Kalman and Bloom Filters Module (KBFM)

The last module of the proposed model is the Kalman and Bloom Filters Module (KBFM) module. The KBFM module consists of two filters. The first one is the Kalman filter which is used to temporarily block of abnormal flooding traffics and zombies IP addresses. The second one is the Bloom filter which is used to permanent block of the demons IP address.

- Kalman Filter

The Kalman filter contains a handful of numerical equations that allow the condition of processing with high-performance computer equipment to be assessed with the ultimate goal of limiting the average error. The capabilities of the filter can be viewed from several perspectives: it evaluates the past as it does today, regardless of whether the exact type is unclear in the frame presented. In the proposed model, the Kalman Strait is under the control of an expert. The specialist sends commands to the Kalman Filter to start or turn them off after measuring predefined limits and nearby traffic, after imperfect irregular flood meters and the IP address of the zombies.

- Bloom Filter

The Bloom filter can be considered the most spatially probable form of information created by Burton Howard Bloom in 1970. This filter can be employed to test and determine if the element is a set member. For example, a question might come back as "not in a set" or "maybe in a set". Components can be added to the equipment that has not yet been switched off. At the end of the day, the likelihood of false positives increases and the number of components added to the equipment increases [14], [15]. In the proposed model, the pathogen is controlled by the specialist. The expert sends comments to the Bloom filter to start or stop it, depending on predefined limits and the amount of incoming traffic on Demon persistent IP address fields, as previously explained.

3.2.4 Evaluation Methods

The evaluation of the proposed model uses measurements such as sensitivity, accuracy and precision. These measurements are obtained from the parts of the matrix elements. The elements

that make up the chaos frame include true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In general, TN is characterized by the number of events that are usually called effective. TP is represents the number of samples accurately referred to as attacks. CE is a level of sampling of attacks incorrectly organized as routine events. In essence, the UN is the scene of common events falsely classified as aggression. The accuracy is characterized by the fact that the proportion of all events is accurately classified (TP, TN) (TP, TN, FP and FN) in all cases. The ratio between TP and total TP and CE is exact positive predictive value. Accessibility is the ratio of TP to the sum of TP and FN. In this work, an accuracy calculation is given in Equation 3.12:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.12}$$

Wherein the precision was calculated as Equation 3.13.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.13}$$

3.3 CHAPTER SUMMARY

This chapter describes the methodology of the proposed model for monitoring, analyzing, detecting and blocking DDoS attacks and FC flooding traffics which is an Adaptive Protection of Flooding Attacks (APFA) model. It is consists of four modules which are the ATDS, DADS, TCMS and the KBFM. However, the TCMS and KBFS represent the contribution of this work. Also, the evaluation methods that have been used in this work is illustrated in this chapter.

4. RESULTS

4.1 INTRODUCTION

In this research, the flood protection paradigm was developed, which autonomously handles network traffic and carries out traffic management and mitigation activities for DDoS attacks. Simulation software for evaluating and analyzing the performance of the proposed concept is presented in this chapter. Furthermore, the simulation system is explained with an emphasis on the context for applications, the simulation setting, and the simulation design.

In this Chapter, a variety of tests are carried out to test the theory and assess if DDoS safety from flood traffic is a positive. The experimental setting includes utilizing the data collection of the 2019 CICDDoS, which is important for the verification and validation of DDoS-protection techniques. In addition to the above-mentioned assessments, output steps are eventually taken to assess the performance of the proposed model in the simulated environment. Finally, this chapter discusses the experimental findings and explores the derived results.

4.2 IMPLEMENTATION

Several tests were conducted to replicate DDoS attacks for the efficiency and protection of DDoS flooded traffic against them to be checked satisfactorily. Moreover, the following subsections describe in detail the simulation conditions, the setting, and the simulation's design, and the experimental results.

4.2.1 Implementation Environment

- Software: C#, accessible for 2013 visual studio, Windows 10, was used in this study. In comparison, various libraries, groups, and functions can be found in C#, which implies it is chosen.

- Hardware: The simulation is implemented by using one laptop with specifications of Intel (R) Core (TM) i7-5500U processor, 2.40GHz, and 16 GB RAM.

4.2.2 Implementation Modelling

The purpose of this study is to strengthen the defence mechanisms that target DDoS attacks. The protection model is critical to be able to identify, categorise and track the attack traffic. In the following subsection, the dataset setting. The CICDDoS 2019 dataset attributes are evaluated to better understand the stimulation level. A large number of DNS requests, including normal DNS requests and irregular DNS requests, are generated using the CICDDoS 2019 dataset. The CICDDoS dataset is then divided into five different groups of traffic (normal, abnormal, unknown, attacker, and victim).

In addition, the total transactions were split into two parts, with the first portion being used for system planning and comprising 60% of the dataset, and the second segment representing 40% of the dataset for tracking. Furthermore, the CICDDoS dataset is separated into four weeks. The Week 1, Week 2, and Week 3 datasets are used for training purposes and contain 20,374 IP addresses submitted by 402,744 applications. The first week consists of 9,412 IP addresses sending 172,838 requests, and the second week consists of 8,357 IP addresses sending 159,373 requests, and the third week consists of 2,605 IP addresses sending 70,533 requests. Whereas the fourth-week dataset used for analysis comprises 13,583 IP addresses and 268,496 requests have been sent.

In addition, as a consequence of the benchmark analysis, the validation stage incorporated the categories of accused, unknown, attacker and victim to act as attack traffic, which can only cope with two modes of traffic that are normal traffic and attack traffic. This job tackles four types of traffic are attacking, abnormal, demons, and zombies while complementing the structure. As a result, attack traffic is considered an attacker, unknown and suspicious traffic, whereas normal traffic is assumed to be attack traffic. By the end of the day, it is understood that victim traffic is sporadic flood traffic. There are many ways to protect against DDoS attacks and each technique uses specific parameters. Even so, the study used several criteria and variables. The latter half of this chapter explores these two additional requirements.

The proposed model is designed to protect the DDoS attack layer of the application. The overall simulation processing model, starting with the ATDS (Step 1-Step 4) which obtained the data set to set the threshold by monitoring and evaluating the incoming traffic imported from the data set,

as shown in Figure 4.1. In addition, according to the ATDS analysis, the threshold is obtained. Whereas, in the case of incoming traffic, less than the limit may have elapsed. Conversely, if the incoming traffic is higher than the stage, the traffic would submit an important signal to the DADS for further analysis. However, the DADS will activate and start tracking the source of each IP address that sends anomalies to the traffic when an attention signal is obtained (Step 5). The root of these IP addresses will then be specified by the DADS and forwarded to the TCS agent for further analysis. Whereas the contribution to this study is made by the TCS agent. In comparison, these elements allow it possible to differentiate what an agent wants (goals) from how the target is reached (plans).

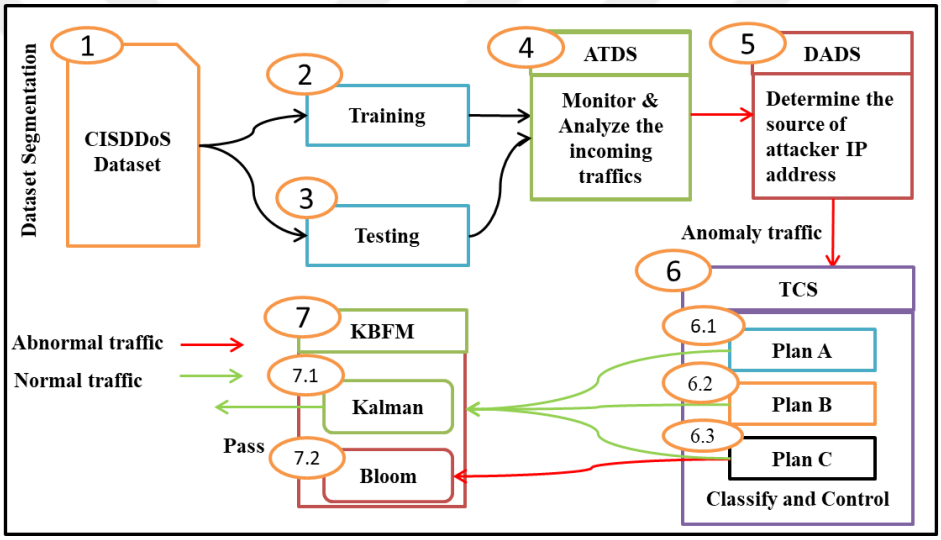

Figure 4.1 : Steps of the Simulation Design

As a result, there are three plans to identify, classify and control traffic. First, plan A will be passed in the case of incoming traffic below the threshold (Step 6.1). On the other hand, if the incoming traffic exceeds the threshold, the next plan will be forwarded for further analysis. Second, in the case of abnormal traffic following Plan B (Step 6.2), which is used to distinguish and separate attack traffic, attack traffic is classified as DDoS and abnormal traffic following traffic behaviour. However, the abnormal traffic sends the temporary block to the Kalman filter (step 7.1) and the DDoS traffic sends further analysis to the next plan. Finally, the DDoS traffic will be separated into the Zombies and Demons by the IP history log parameter (Step 6.3). Whereas the Zombie traffic will send a temporary block to the Kalman filter (step 7.1) and the Demon traffic will send a permanent block to the bloom filter (Step 7.2). Note that the system is running twice during the

implementation phase, one for training and the other for testing. At the training stage, the data set for Week 1, Week 2, and Week 3 was used to set the parameters and to obtain the conditions. Whereas the dataset for Week 4 was used to test and evaluate the system at the test stage.

4.3 EXPERIMENTS

The models and measurements used to determine the effectiveness of the proposed model are described in this section. The studies examine the impact of the Distributed Denial of Service and Forced Causality attacks to evaluate the proposed model performance. The proposed model is evaluated and compared to tasks in the dataset and the results are then tested against this dataset directly. The proposed technology is compared to the solution of Zhou et al. [14], as well as Chen et al. [29], and finally Sreeram & Vuppala [67].

4.3.1 Training Stage

The CISDDoS 2019 dataset configuration is 60 percent of the external server dataset used for system training, and 40 percent is used for system testing. In this experiment, the first stage of implementation analyses the dataset and the average frequency of incoming requests. In addition, the system is implemented based on the parameters of the period is set to 7 and the SSM is set to 130. The M traffic model represents the fourth week's traffic.

4.3.2 Results of AL-DDoS Base Model

The AL-DDoS implementation model by Zhou et al. [14] was tested by using a discrete random sequence, and its probability of an attack was evaluated by using the AL-DDoS attack model. The above is tested using the CISDDoS dataset to compare and measure the effectiveness of the AL-DDoS based model, where the CISDDoS dataset produces two forms of traffic, regular and DDoS.

In addition to Segment 4 of the study, the robot is also being fitted with a new brain and operating out of a new location. In addition, 60 percent of the data for the study's dataset is from the sampled IP addresses. Finally, the robot, which has been programmed by the study, has made a combined total of over a hundred thousand requests, all geared toward keeping the database growing. At the stage when the specifications are available, the requirements of the device are obtained. On the other side, 40 percent of the external CISDDoS 2019 dataset for the fourth week was used to

validate this model. The fourth-week dataset comprises 13,583 IP addresses and 268,496 requests, as seen in Figure 4.3, the average of incoming requests.

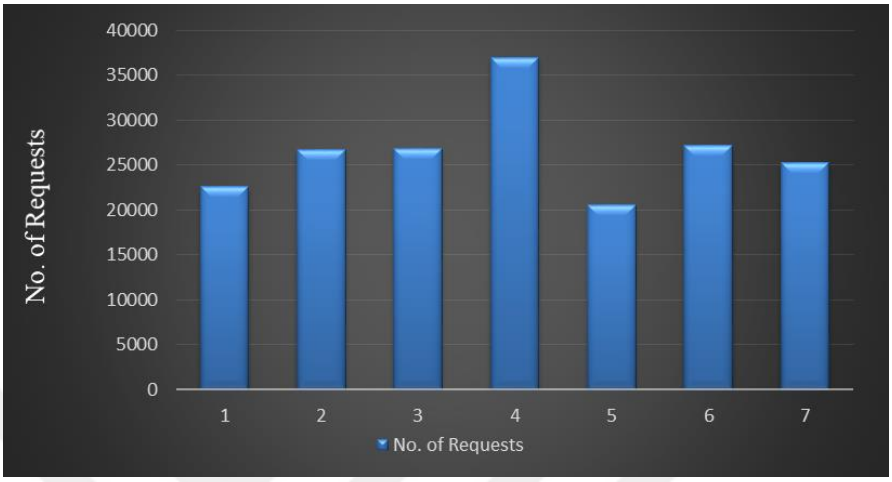

Figure 4.2 : Average of Incoming Requests for the 4th Week

The AL-DDoS base model by Zhou et al. [14] is used for the analysis of external server traffic data. In addition, it should be noticed that the characteristics of the Attack ID and Attack Description are overlooked, as they only provide more descriptions of the attacks carried out. The performance of the AL-DDoS base model is calculated based on the window height, duration, and SSM parameters for each external traffic data execution. The utility of the AL-DDoS base model is calculated based on the correct classification of the traffic instance for attack and daily traffic.

The number of the attack traffic recorded from the AL-DDoS base model is 13,583 IP addresses and 268,496 queries. Traffic was split into two classes, assaulting traffic and normal traffic, according to the system analysis. The ordinary traffic is then transmitted, while the attack traffic is sent to the adaptive agent for further analysis. Just anomaly traffic can be detected by the AL-DDoS base model and the source of each IP address that sends anomaly traffic is determined such that the IP address is deposited in the bloom filter and all anomaly traffic is dropped. Subsequently, some of the abnormal overwhelmed traffic, and zombie requests belonging to legitimate users have been dropped and interrupted forever by those IP addresses.

The result demonstrated that the AL-DDoS base model was effective in detecting and blocking DDoS attack IP addresses with 99.13 percent precision, 99.14 percent accuracy, and 99.99 percent sensitivity, as seen in the following section. However, the AL-DDoS model is failing to deal with

abnormal traffics and zombie flow. In contrast, in addition to DDoS traffic, the proposed model can define and track those types of traffic and to create a temporary block for them. The findings and analysis of the proposed comprising the agent module are discussed in the next chapter.

4.3.3 The Results of the proposed Model

The analysis, validation, and evaluation of the AL-DDoS base model were discussed in previous sections, while this section is supported by the TCS. After evaluating the traffics at the last stage of the attack, the agent can provide more analysis based on the three functions ati, atd, & ath.

The DADS anomaly traffic is received by the adaptive agent and the amount of traffic received is 13,583 IP addresses and 268,496 requests. The traffic was classified by the TDS and DADS and the attack traffic was blocked after the system was detected. To classify the incoming anomaly traffic into DDoS and abnormal based on the BDI architecture, the agent will receive the anomaly traffic to perform further analyses, and then the DDoS traffic has been categorized into demons and zombies. The agent then controls the traffic and decides to send the traffic to the filter to be intended.

The observations involve three examples, normal, abnormal, and DDoS traffics, as well as the agent's response for all three instances. The parameters of the traffic anomaly are categorised to be unusual according to the traffic conduct, t_0^ ; discrete, t_1^ ; and continuous, t_2^ , as shown in Table 4.1. This classification helps to identify the attack type.

Table 4.1 : Scenario Setting

Traffic conditions	Traffic	Behavior
Anomaly	13,583	t_0
Abnormal	2,745	t_1
DDoS	10,838	t_2

4.3.3.1 Normal Traffic Processing

The first feature ati has elementary objectives of determining if the incoming traffic is usual or irregular. In a case where 7.6428 can be regarded as a threshold value for traffic strength according to the TDS analyses, the actual incoming traffics value is changed in b1, then it is contrasted with b 0. If b 1 value is smaller than that of b 0 (i.e., 7.6428), then the situation is known as normal traffic. If the b1 'value is greater than b0, then the next step of calling atd function is triggered to evaluate the traffic state.

4.3.3.2 Abnormal Flooding Traffic Processing

For abnormal traffics, in which traffic severity is 7.6428, the same measures as the first case are taken. The irregular traffic is diagnosed by atd according to b2. In this case, according to the correlation coefficient, the conduct of the traffic is discrete flow, b2, t 1, and atd: b2 and i0 . As a consequence, the Kalman filter will be directed by the agent to operate with x functionality, which will then momentarily filter out an IP address of 2,745. Although the sum of DDoS traffic observed in this process is 10,838 IP addresses, a random number of requests are sent to each IP address. This number of IP will be sent to the agent to evaluate, identify and block according to the agent function.

4.3.3.3 DDoS Traffic Processing

When incoming traffic is 10838, it has a constant flow, b2, t2. Then, the atd: b2, i1and i 2, which is regarded by DDoS as an attack, which means invoking ath. The goal of ath is to tackle the undead, r1 and ghosts, r2 demands, ath: r1, i1, r2 and i2. The zombies IP address is forward to Kalman filters for temporally blocking, while the demons IP address is sent to Bloom filters for permanent blocking. Later, the IP addresses for abnormal traffics and DDoS traffics will be stored and redirected to the IP Address buffering for future security actions.

In Figure 4.4, the graph offers descriptions of the number of anomaly requests for the fourth week from the dataset selected to pass through the first function of the agent that is ati. The traffic in the fourth week was divided into seven days starting on day 1, with a demand of 37,919. Also, it was noticed that there was a minor decrease in traffic on the second day. However, the highest traffic was reported on the fourth day with 56,991 requests. It is also clear that traffic has declined over the last 3 days. As seen in Figure 4.11, the total number of requests for the AL-DDoS base model

for the week under examination was 268,496. The agent in the ati function even detected the same total number of requests according to the normal traffic strength parameter. Then, the cumulative number of requests for further analysis was sent to the second function of the agent.

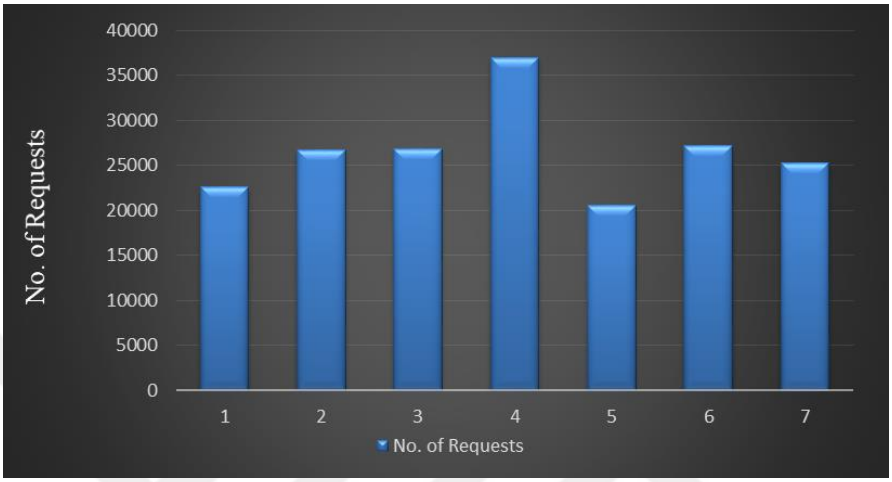

Figure 4.3 : Average Anomaly Traffics for the 4th Week in the First Stage of the Agent Cycle

Figure 4.5 displays the sum of attack requests from the selected data set for the fourth week from the atd function of agent. The function is to break attack traffic between the DDoS and the abnormal based on traffic actions, then to apply the abnormal traffic for further analysis into Kalman filter and DDoS traffic at the final function of the adaptive agent, ath. Furthermore, 31.502 DDoS requests and 759 abnormal requests were observed on the first day of traffic. On the second day, traffic increased marginally and only DDoS was seen on the third day. On the contrary, traffic grew and fell marginally over the last four days. The total number of requests reported by the proposed model for the week evaluated for the DDoS assault was 264,551, while the observed abnormal traffic was 3945. Then, for more study, the total amount of DDoS requests was submitted to the third feature in the adaptive agent, ath.

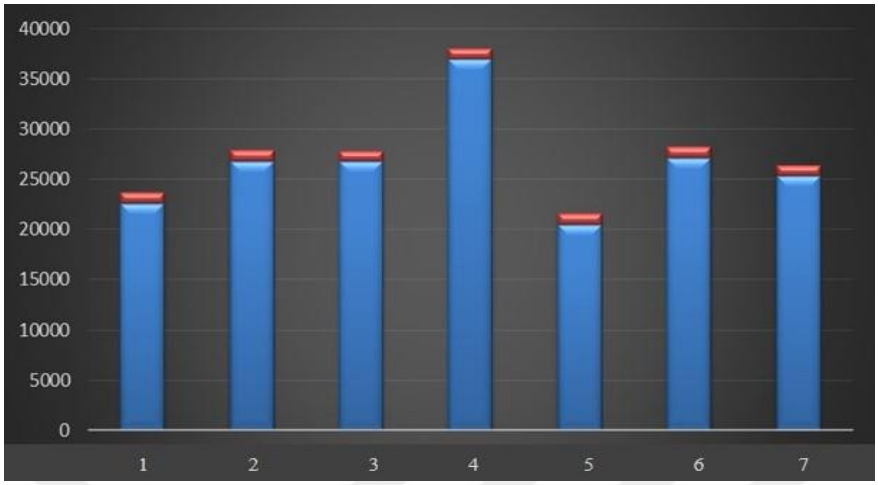

Figure 4.4 : Average traffics for 4th Week in the Second Stage of the Agent

Figure 4.6 indicates the amount of abnormal, zombies, and demons demanded from the selected dataset for the fourth week, going through the third agent function, which is an ath. The task of this fanction is to classify the DDoS traffic into demons and zombies as seen in table 4.5. Then traffic from the Zombies was sent to the Kalman filter and traffic from the demons to the bloom filter. In comparison, 19,257 Demons requests and 13,731 Zombies requests were identified by the first traffic. The last six days have seen a small spike in traffic.

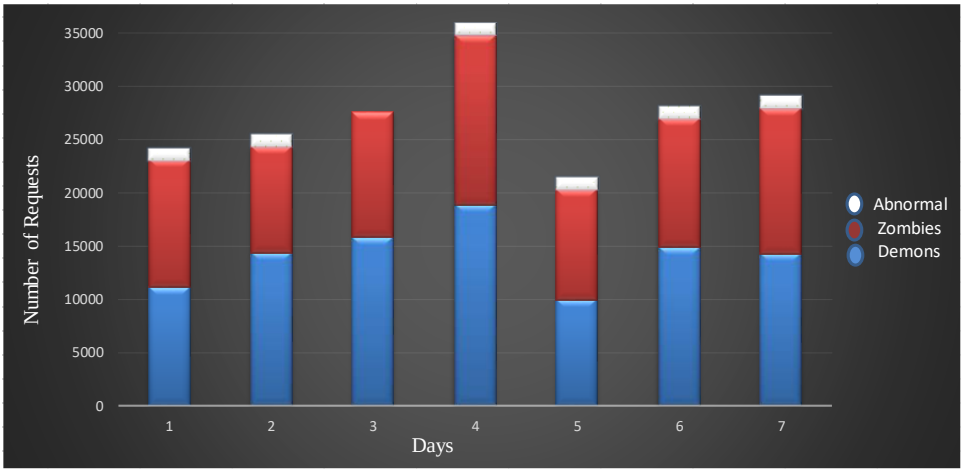

Figure 4.5 : The average of traffics for the fourth week in the third stage of the agent

The total number of requests listed by the last function in the TCS was 264551, including 175624 Demon requests and 88927 Zombie requests, based on the table, while the abnormal traffic registered was 3945, as seen in the figure above. However, the figure reveals that the proposed

model was capable of defining and distinguishing the traffic of DDoS and abnormal and then determine the traffic into demons and zombies.

The proposed model has been desined to guard against DDoS attacks and abnormal floods impacting application layer. The results obtained in the first stage of the agent showed 99.11 percent accuracy, 99.14 percent precision, and 99.99 percent sensitivity for DDoS attack detection, while 99.92 percent accuracy, 99.85 percent precision, and 99.96 percent sensitivity were achieved in the second stage of the agent. Besides, the results hit 99.91 percent accuracy, 99.89 percent precision, and 99.93 percent sensitivity in the last function of the adaptive agent. However, the proposed model produced an excellent performance in DDoS attacks and abnormal traffics with an average accuracy of 99.64 percent, 99.62 percent accuracy, and 99.96 percent sensitivity. The proposed model is therefore distinguished it self from the related work in that the assumption and sensitivity of DDoS attack and abnormal flood detection have been determined. Moreover, seven days of the fourth week were fitted and tested from the selected data set to achieve these effects. As can be seen in the chart, the proposed model attained a high degree of accuracy, presision, and sensitivity, increasing day by day with the willingness of the machine to learn.

4.4 DISCUSSION

A disruptive incident that does not require internal system access is a danger to DDoS. It's often difficult to detect in the early stage. This involves the deployment of a significant number of zombies, who are at risk of possible injury. A majority of nodes were redirected to a particular goal by the critical forms of the attack, which may have catastrophic effects for its victims and plague the networks. There is another form of traffic flooding, the abnormal. Abnormal traffic is defined as a type of network traffic comparable to DDoS traffic but originating from legitimate users. It is a traffic like a DDoS attack, which enables multiple illegitimate users to reach the website at the same time.

Several techniques have been formulated, built, and applied to defend application layer against DDoS attacks. However, the concealment of malicious traffic behind legitimate traffic as well as abnormal traffics appears to be a common scourge in these protection models. Any of these can not readily discriminate between actual traffic and deceptive mimicry in terms of negative performance and false positives. The research proposed, implemented, evaluated, and contrasted

with the results of the proposed model revealed that the proposed model was effective in the protection against DDoS attacks targeting application layer.

By comparing the proposed model with that of Zaho et al., [14]; Chen et al., [29]; Sreeram & Vuppala, [67], the results of the work were evaluated. For this reason, 60 percent of the dataset was used in model training, while the remaining 40 percent was used to test the model, as the dataset splitting configuration is widely accepted. A comparison of the proposed model is rendered based on the dataset slicing setup. Note that there is no improvement in precision, but it is so successful when DDoS attacks and abnormal floods sense traffic.

The model, though, differs from previous methods by utilizing the decision module, which utilizes the agent to modify and recognize the DDoS (demon and zombie) and the abnormal flood traffic. The agent is equipped with three analysis functions, which operate on three normal traffic intensity parameters, traffic attack behavior, and an IP address history file. The agent responds differently to each of the attacks with different forms of filtering behavior. The comparison results are presented in Table 4.2. It can be observed from this table that, compared to other simulations utilizing the same dataset, the overall accuracy of the proposed model was 99.64 percent.

Table 4.2 : Comparison Results of Proposed Model with Related Work

Model	Techniques	Accuracy %
Zhou et al., [14]	Statistical	99.1
Chen et al., [29]	Random Forest	99.2
Sreeram & Vuppala, [67]	Bat Algorithm	94.8
Proposed model	Agent and Statistical	99.6

4.5 CHAPTER SUMMARY

Besides explaining the results of the installation process, the purpose of this chapter is to describe the simulation measures. This chapter addresses the implementation environment for the proposed system, the dataset in which the method is centered, and the specifics concerning the dataset.

Simulation architecture, simulation environment, simulation specification, and simulation implementation are discussed in depth to render the simulation complete. Therefore, the AL-DDoS test culminated in a less likely outcome relative to the proposed model. In the next part, the author will discuss forthcoming works in the area.

5. CONCLUSION

5.1 CONCLUSION

Several studies have been established in the field of DDoS attacks detection and prevention and find the best solutions to overcome this type of attack. These attacks make it challenging to distinguish between normal traffic and harmful traffic. Furthermore, blocking the attack traffic is a more difficult task to accomplish, because it can hardly be done with having an effect on the server's performance. Against this backdrop, this work was conducted with the aim of investigating the different approaches and methods of detecting and countering DDoS attack targeting application layer. However, the proposed model distinguishes itself from the previous methods by employing the agent which has the ability to control the attack traffics. The simulation runs the CICDDoS 2019 datasets to test and evaluate the performance of the proposed model. The simulation results show that the proposed model is able to achieve high accuracy of 99.64%, and the precision of 99.62%

5.2 FUTURE WORKS AND RECOMMENDATIONS

In the future, this work is hoped to be continued in the direction of detection and prevention to various anomalies targeting web servers that are not addressed yet in this thesis, such as

Since each anomaly has a different impact on the QoS of a web server, it is important to differentiate between them and make corresponding countermeasures.

In addition, testing the proposed model by real data could provide more confidence in its capability to perform in real-time.

.

REFERENCES

[1] Aamir, M., & Arif, M. (2013). Study and performance evaluation on recent DDoS trends of attack & defense. International Journal of Information Technology and Computer Science (IJITCS), 5(8), pp. 54- 65.

[2] Alssir, F. T., & Ahmed, M., (2012). Web security testing approaches: comparison framework. In Proceedings of the 2011 2nd International Conference on Computer Applications and Computational Science, Berlin, Heidelberg, Springer, pp. 163-169.

[3] Suresh, P. (2016). Survey on seven-layered architecture of OSI model. International Journal of research in computer applications and robotics, 4(8), 1-10.

[4] Farhat, V., McCarthy, B., Raysman, R., & Knight, L. L. (2011). Cyber-attacks: prevention and proactive responses. Journal of network security and Cryptography, 16 (11), pp. 1-12.

[5] Suresh, P. (2016). Survey on seven layered architecture of OSI model. International Journal of research in computer applications and robotics, 4(8), 1-10.

[6] Alssir, F. T., & Ahmed, M., (2012). Web security testing approaches: comparison framework. In Proceedings of the 2011 2nd International Conference on Computer Applications and Computational Science, Berlin, Heidelberg, Springer, pp. 163-169.

[7] Richardson, R., & Director, C. S. (2008). CSI computer crime and security survey. Computer Security Journal, 1 (3) pp. 1-30.

[8] Jovičić, B., & Simić, D. (2006). Common web application attack types and security using Asp. Net. Computer Science and Information Systems, 3(2), pp. 83-96.

[9] Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfaris, R. (2012). Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art. International Journal of Computer Applications,49 (7), pp. 1-9.

[10] Khalaf, B. A., Mostafa, S. A., Mustapha, A., Mohammed, M. A., & Abduallah, W. M. (2019). Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. IEEE Access, 7, 51691-51713.

[11] Juneja, D., Chawla R., & Singh, A. (2009). An Agent-Based Framework to Counter attack DDoS Attacks. International Journal of wireless Networks and communications, 1(2), pp. 193-200.

[12] Singh, K., Singh, P., & Kumar, K. (2017). Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. Computers & Security, 65 (10), pp. 344-372.

[13] Kong, B., Yang, K., Sun, D., Li, M., & Shi, Z. (2017). Distinguishing Flooding Distributed Denial of Service from Flash Crowds Using Four Data Mining Approaches. Computer Science and Information Systems journal, 14 (3), pp. 839-856.

[14] Behal, S., Kumar, K., & Sachdeva, M. (2017). Characterizing DDoS attacks and flash events: Review, research gaps and future directions. Computer Science Review, 25 (2), pp. 101-114.

[15] Prasad, K. M., Reddy, A. R., & Rao, K. V. (2017). BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web. Journal of King Saud University, 14 (3) pp. 1-15.

[16] Zhou, W., Jia, W., Wen, S., Xiang Y., & Zhou, W. (2014). Detection and defense of application-layer DDoS attacks in backbone web traffic. Future Generation Computer Systems, 38(19), pp. 36-46.

[17] Saleh, M. A., & Abdul Manaf, A. (2015). A novel protective framework for defeating http-based denial of service and distributed denial of service attacks. The Scientific World Journal, 21(3), pp. 1-20.

[18] Khalaf, B. A., Mostafa, S. A., Mustapha, A., Mohammed, M. A., & Abduallah, W. M. (2019). Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. IEEE Access, 7, 51691-51713.

[19] Aamir, M., & Zaidi, M. A. (2013). A survey on DDoS attack and defence strategies: from traditional schemes to current techniques. Interdisciplinary Information Sciences journal, 19(2), pp. 173-200.

[20] Bandara, K. R., Abeysinghe, T. S., Hijaz, A. J., Darshana, D. G., neez, H., Kaluarachchi, S. J., & DhishanDhammearatchi, M. (2016). Preventing DDoS Attack Using Data Mining Algorithms. International Journal of Scientific and Research Publications, 6(10), pp. 390-400.

[21] Barrionuevo, M., Lopresti, M., Miranda, N., & Piccoli, F. (2017). An Anomaly Detection Model in a LAN Using K-NN and High-Performance Computing Techniques. Argentine Congress of Computer Science journal, 790 (17), pp. 219-228.

[22] Chang, R. K. (2002). Defending against flooding-based distributed denial-of-service attacks: a tutorial. IEEE communications magazine, 40(10), pp. 42-51.

[23] Chhabra, M., Gupta, B., & Almomani, A. (2013). A novel solution to handle DDOS attack in MANET. Journal of Information Security, 4(3), pp. 165-179.

[24] Lemon, J. (2002, February). Resisting SYN Flood DoS Attacks with a SYN Cache. Journal of Applied Science, 21(7) pp. 89-97.

[25] Acohido, B., & Swartz, J. (2009). Hacker attack takes down Twitter, Facebook, LiveJournal. Retrieved on August 6, 2009, from https://phys.org/news/2009-08-hacker-twitter-facebook.html.

[26] Kumar, S., & Sangwan, S. (2011). Adapting the software engineering process to web engineering process. International Journal of Computing and Business Research, 2(1), pp. 42-63.

[27] Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE International Symposium on Secure Software Engineering, Mitropoulos, IEEE, pp. 13-15.

[28] Howser, G. (2020). The OSI Seven Layer Model. In Computer Networks and the Internet (pp. 7-32). Springer, Cham.

[29] Zhou, W., Jia, W., Wen, S., Xiang Y., & Zhou, W. (2014). Detection and defense of application-layer DDoS attacks in backbone web traffic. Future Generation Computer Systems, 38(19), pp. 36-46.

[30] Jazi, H. H., Gonzalez, H., Stakhanova, N., & Ghorbani, A. A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Computer Networks, 121, pp. 25-36.

[31] Jyothi, V., Wang, X., Addepalli, S. K., & Karri, R. (2016). Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect DDOS attacks. In Proceedings of the 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems, Washington, IEEE pp. 587-588.

[32] Kaur, P., Kumar, M., & Bhandari, A. (2017). A review of detection approaches for distributed denial of service attacks. Systems Science & Control Engineering, 5(1), pp. 301-320.

[33] Hamad, H., & Al-Hoby, M. (2012). Managing intrusion detection as a service in cloud networks. International Journal of Computer Applications, 41(1), pp. 35-40.

[34] Jung, J., Krishnamurthy, B., & Rabinovich, M. (2002). Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In Proceedings of the 11th international conference on World Wide Web, Hawaii, IEEE, pp. 293-304.

[35] Li, S. H., Kao, Y. C., Zhang, Z. C., Chuang, Y. P., & Yen, D. C. (2015). A network behavior-based botnet detection mechanism using PSO and K-means. Transactions on Management Information Systems, 6 (1), pp. 1-30

[36] Mousavi S. M. (2014). Early detection of DDoS attacks in software defined networks controller. Electrical and Computer Engineering Journal, 19 (7), pp 65-77.

[37] Arbor Networks (2018). DDoS attack report. Retrieved on March 6, 2018 https://www.netscout.com/report/.

[38] Verisign, (2014). Network security technical report. Retrieved on April 24, 2014, from https: //www.verisign.com/en_IN/ security-services/ddos-protection/ddos-report/index.xhtml.

[39] Khandelwal, S. (2016). Cybercrime Technical Report. Retrieved on Jnu 10, 1018, from http://thehackernews.com/2016/ 01/biggest-ddos-attack.html.

[40] Woolf, N. (2016). DDoS attack that disrupted internet was largest of its kind in history, Retrieved on April 24, 2014, from URL https://www. theguardian. com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

[41] Bhandari, A., Sangal, A. L., & Kumar K. (2015). Destination address entropy based detection and traceback approach against distributed denial of service attacks. International Journal of Computer Network and Information Security, 7(8), pp. 9-20.

[42] Kaur, H., Singh, G., & Minhas, J. (2013). A review of machine learning based anomaly detection techniques. International Journal of Computer Applications Technology and Research, 2 (2) 1307-1319.

[43] Saharan, S., & Gupta, V. DDoS Prevention: Review and Issues. Advances in Machine Learning and Computational Intelligence, 579-586.

[44] Juneja, D., Chawla R., & Singh, A. (2009). An Agent-Based Framework to Counter attack DDoS Attacks. International Journal of wireless Networks and communications, 1(2), pp. 193-200.

[45] Mostafa, S. A., Ahmad, M. S., Mustapha, A., & Mohammed, M. A. (2017). A concise overview of software agent research, modeling, and development. Software Engineering, 5(1), pp. 8-25.

[46] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security Journal, 28(7), pp. 18-28.

[47] Kim, Y., Lau, W. C., Chuah, M. C., & Chao, H. J. (2006). PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. IEEE transactions on dependable and secure computing, 3(2), pp. 141-155.

[48] Gonzalez, J. M., Anwar, M., & Joshi, J. B. (2011). A trust-based approach against IP-spoofing attacks. In Proceedings of the Ninth Annual International Conference on Privacy, Security and Trust, Paris, IEEE, pp. 63-70.

[49] Shiaeles, S. N., Katos, V., Karakos, A. S. & Papadopoulos, B. K. (2012). Real time DDoS detection using fuzzy estimators. computers & security, 31(6), pp. 782-790.

[50] Li, Y., Guo, L., Tian, Z. H., & Lu, T. B. (2008). A lightweight web server anomaly detection method based on transductive scheme and genetic algorithms. Computer Communications Journal, 31(17), pp. 4018-4025.

[51] Rahul, A., Prashanth, S. K., Suresh, B., & Arun, G. (2012). Detection of Intruders and Flooding in Voip Using IDS, Jacobson Fast and Hellinger Distance Algorithms. IOSR Journal of Computer Engineering, 2(2), pp. 30-36.

[52] Nguyen, H. V., & Choi, Y., (2010). Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework. International Journal of Electrical, Computer, and Systems Engineering, 4(4), pp. 247-252.

[53] Barrionuevo, M., Lopresti, M., Miranda, N., & Piccoli, F. (2017). An Anomaly Detection Model in a LAN Using K-NN and High-Performance Computing Techniques. Argentine Congress of Computer Science journal, 790 (17), pp. 219-228.

[54] Braga, R., Mota, E., & Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In Proceedings of the 35th International Conference in Local Computer Networks (LCN), Colorado, IEEE, pp. 408-415.

[55] Niyaz, Q., Sun, W., & Javaid, A. Y. (2016). A deep learning based DDoS detection system in software-defined networking (SDN). Computer Science and Cryptography, 24 (9), pp. 1-13.

[56] Chambers, N., Fry, B., & McMasters, J. (2018). Detecting Denial-of-Service Attacks from Social Media Text: Applying NLP to Computer Security. In Proceedings of the 18 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis IEEE, pp. 1626-1635.

[57] Kotenko, I., & Ulanov, A. (2006). Agent-based Simulation of Distributed Defense Against Computer Network Attacks. In Proceedings of the 20th European Conference on Modelling and Simulation, Berlin, Springer, pp. 1-6.

[58] Kesavamoorthy, R., & Soundar, K. R. (2018). Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system. Cluster Computing journal, 7 (11), pp. 1-8.

[59] Singh, K., Dhindsa, K. S., & Bhushan, B. (2018). performance analysis of agent based distributed defense mechanisms against DDOS attacks. International Journal of Computing, 17(1), pp. 15-24.

[60] KDD Cup Dataset, (1999). KDD DDoS Attack Dataset. Retrieved on July 10, 1999, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[61] Kim, Y., Lau, W. C., Chuah, M. C., & Chao, H. J. (2006). PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. IEEE transactions on dependable and secure computing, 3(2), pp. 141-155.

[62] CAIDA dataset (2011). The caida ddos attack dataset. Retrieved on August 3, 2011, from http://www.caida.org/data/passive/ddos-20070804dataset.xml.

[63] DARPA Dataset, (2009). The DARPA DDoS attack dataset. Retrieved on April 3, 2009, from http://www.isi.edu/ant/traces/DARPA2009DDoSatt05.README.

[64] CIDDS dataset (2017). The CAIDA DDOS attack dataset. Retrieved on August 3, 2017, from https://www.hs-coburg.de/forschung-peration/forschungsprojekte-oeffentlich/ingenieurwissenschaften/cidds-coburg-intrusion-detection-data-sets.html.

[65] UNB, CICDDOS 2019, available online on: https://www.unb.ca/cic/datasets/ddos-2019.html, accessed on: 2019.

[66] Rajon, S. A. (2016). Fundamentals of Computer Programming with C. SA AHSAN RAJON, Prentice/Hall International, pp. 99-145.

[67] Sreeram, I., & Vuppala, V. P. (2017). HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. Applied Computing and Informatics,3(4) pp. 59-66.