Computing and Informatics, Vol. 42, 2023, 1060–1090, doi: 10.31577/cai_2023_5_1060

CYBERSECURITY: BOTNET THREAT DETECTION ACROSS THE SEVEN-LAYER ISO-OSI MODEL USING MACHINE LEARNING TECHNIQUES

Rami Mohammed Baazeem

MIS Department, University of Jeddah
Saudi Arabia
e-mail: rbaazeem@uj.edu.sa, rami_mohammed78@outlook.com

Abstract. The Open System Interconnection (OSI) model, consisting of seven layers, has become increasingly important in addressing cyber security issues. The rapid growth of network technology has led to a rise in cyber threats, with botnets taking over fixed and mobile computers. The widespread availability of mobile devices has led to increased app consumption, with 60 % of Android malware containing major or minor botnets. The ease of accessibility of mobile devices has accelerated the adoption of mobile apps in various use cases. This article aims to identify and reduce botnets in operating systems, focusing on identifying them faster and reducing attack impact. The study analyzes botnet characteristics under controlled conditions and creates four traffic flow components across multiple time ranges. Using machine learning, flow vectors are created to identify internet flows as botnet flows or predicted flows. The method uses a combination of Boosted decision tree ensemble classifier, Naive Bayesian statistical classifier, and SVM discriminative classifier to accurately identify both well-known and novel botnets, reducing false positives and improving detection accuracy.

Keywords: Spoofing, hijacking, SYN flood, DoS, DDoS, cybersecurity threats, ISO-OSI model, machine learning

1 INTRODUCTION

The contemporary communication and computer infrastructures are very susceptible to many of various kinds of attacks. Modern communications and computing infrastructures are vulnerable to cyber-attacks like phishing, malware, DDoS, and advanced persistent threats due to system vulnerabilities. Cloud and mobile device use has increased these risks. Robust security measures and staying up to date with security trends are crucial for organizations to mitigate these risks. The traditional starting point for these kinds of attacks is the installation of malicious software, often known as malware, which may take the form of computer viruses, worms, and Trojan horses [1]. These attacks harm on the Internet and result in many issues for users, including data corruption, delays, and severe capacity waste on the network. In addition, some of these assaults are used to obtain control of Internet hosts, which may then be used to carry out denial-of-service attacks on other websites. These attacks can be carried out in many different ways. If an attacker is successful in getting access to network hosts [2], this might result in considerable harm to the network. This damage could include the interruption of e-commerce websites, news sources, network infrastructure, routers, and root name servers, amongst other things. An adversary might potentially speed up the process of gaining access to systems and improve their chances of being successful by making use of malicious software. The term "bot" is used to refer to every one of these compromised machines. A collection of computers that have been compromised and are now referred to as bots might be referred to as a botnet [3]. Using a Command and Control (C & C) channel that has been specially set up, an intruder known as a botmaster manages these bots remotely from a faraway location. This newly discovered facet of the danger presented by botnets puts the lives of millions of people and the essential network infrastructure of countries all over the world at hazard [4]. The exponential growth of botnets poses a substantial threat to the authenticity of both computer networks and data. Botnets are a major threat to computer security, with some of the most common threats including Distributed Denial of Service (DDoS) attacks, spamming, click fraud, theft of sensitive information, spreading malware and viruses, and unauthorized access to infected machines. Consequently it is of the utmost necessity to move fast in the direction of developing a dependable detection system against botnets. This chapter begins with a brief overview of botnets, followed by a discussion of the motivation for doing the research, followed by a rundown of the various research projects that were carried out to accomplish the goals, and finally, the chapter concludes.

Open System Interconnection, commonly known as a reference model, is what is meant when someone uses the phrase "OSI model". In the year 1984, the International Organization for Standardization (ISO) was the organization responsible for developing it. This seven-layered architectural model was created to assist vendors and developers in the process of creating interoperable network systems [5].

The Open Systems Interconnection (OSI) model has seven layers: the physical layer, the media access control (MAC) layer, the network layer, the transport layer, the session layer, the presentation layer and the application layer. It offers an overview of how packets flow through the network when communication is taking place [1] and discusses the many roles that are performed by each tier. At each of these tiers, different protocols are put into operation. Using this OSI model with seven layers, each job in the network was broken down into its component elements, which were referred to as layers. Each layer is responsible for carrying out the particular duty that has been delegated to it. Every layer in the network will provide a service to the layer above it and will make use of the services offered by the layer below it [2]. One can have a better understanding of the visual depiction of what is taking place in internet communication and how the devices are functioning inside the network with the assistance of this model.

At this point, the development of internet technology is increasing at an alarmingly quick rate, which in turn leads to high-profile hacking tactics. This OSI model not only helps one understand how network communication works, but it also enables one to examine how each layer is impacted by potential risks to one's privacy and security. A hostile act that aims to harm, steal, or disrupt electronic data and equipment are referred to as a cyber threat. The proliferation of online dangers have led to an increase in both the importance of network security and its status as a primary source of anxiety [6]. Social media can increase anxiety and depression due to factors such as social comparison, cyberbullying, fear of missing out, information overload, exposure to negative news, and addiction. It is crucial to acknowledge these triggers and take measures to minimize their adverse effects on mental health. The design of the internet itself makes it easier for many security breaches to take place. Discovering and addressing network security issues will be made easier by gaining an understanding of the vulnerabilities and different forms of attacks that might be launched against the system.

The authors of [7] researched the potential online dangers posed by social networking services. The fact that the number of people using social networking sites is growing daily provides countless opportunities for cybercriminals, who may target victims and steal critical information by using these platforms. When accessing social media networks, a person is exposed to a fast increasing number of risks to both their privacy and their security. Since Wireless Sensor Network (WSN) uses the OSI model, several kinds of attacks and countermeasures are described in [8]. These are carried out to maintain significant aspects such as confidentiality, authenticity, and integrity.

A breach of security that takes advantage of a weakness in the system or program is what we refer to as a threat. In this context, the term "vulnerability" refers to flaws or openings in the design. Vulnerability in a network may often be traced to faulty design of the network's hardware or software [9]. A danger may be either purposeful or inadvertent, such as those posed by nature or those that humans have created. An assault is always prompted in response to a threat. Unlawful activity carried out on a system is referred to as an attack. When anything causes harm, it is usually done on purpose and with malice. Controlling the vulnerabilities allows for protection against both potential threats and actual assaults. In general, attacks may be broken down into two categories:

1. the assault from inside, and
2. the attack from outside [10].

A passive attack often involves listening to and analysing the traffic on a network without interfering with the devices' ability to communicate with one another. Defending against this kind of assault may be challenging. On the other hand, an active attack is one that intentionally seeks to alter or remove the data that is being sent across the network. This attacker will often cause problems for the functioning of the network.

DoS and DDoS are the most destructive attacks that may be performed at any of the OSI model levels [11]. DoS stands for denial of service, while DDoS stands for distributed denial of service. A single system launches an assault on the victim of a DoS, however in a Distributed Denial of Service (DDoS), several bots referred to as Zombies are utilized to deliver an overwhelming volume of traffic to the victim. DDoS refers to an assault that uses a large number of dispersed devices to perform a DoS.

Security breaches and assaults on the network layer are very perilous and might bring the system's performance to a lower level overall. Because the network layer is responsible for handling source and destination IP addresses, there is a high probability that spoofing attacks will be initiated [12]. In addition, the protocols that make up the transport layer, such as TCP and UDP, contain security flaws that may be exploited to launch flooding assaults.

The "Man in the Middle" attack enables the perpetrator to listen in on and steal information from a conversation taking place between two devices on a network [13]. MITM may either be passive or active, depending on the situation. This attack may be carried out in a variety of ways, including session hijacking, ARP poisoning, and DNS spoofing, among others.

The main contribution is discussed as follows: This article aims to improve the identification and mitigation of botnet attacks on popular operating systems like Microsoft Windows and Android. By analyzing botnet characteristics under controlled conditions and using machine learning techniques, the article develops a method that accurately classifies internet flows as botnet or normal flows. The approach reduces false positives, enhances detection accuracy, and minimizes the impact of botnet attacks.

2 RELATED WORKS

Bots have the potential to be highly helpful pieces of software when they are designed to assist a human user in some manner, such as by automating an easy process or by simplifying the user's control over a range of programs and systems. During the time in question, the Eggdrop bot was widely acknowledged as a valid IRC bot and a significant number of individuals made substantial use of it in this capacity. Unfortunately, subsequent IRC bots are also built to undertake destructive activities, which compromise the system and enable the attacker to acquire sensitive information and participate in other malicious operations. This allows the attacker to forward their malicious agenda. Both the Trojan Sub7 and the worm were responsible. Pretty Park is contending for the title of the piece of software that is credited with being the one that began the process of creating the botnet. The malware known as Sub7 is a Trojan horse, while Pretty Park is a worm. These malicious programs were the first to propose the concept of connecting to an IRC channel to listen for commands to execute harmful code [14].

This bot was developed on top of the mIRC client, which can access raw TCP and UDP sockets and run user-defined scripts in response to activities that take place in an IRC channel. Because of this, it is well suited for implementation in simple denial-of-service (DoS) 4 attacks [15]. There are at least a hundred distinct types of GTbot that are currently active and comprise an IRC client. These bots are all a member of the botnet. Before that, in 2002, the commercialization of SDBot made it feasible to build a new botnet that was built on top of existing ones. This was previously not conceivable. This bot was made up of a solitary binary file that was very short and was written in C++. The features of its backdoor make it possible for additional instructions and functions to be carried out on a machine that has been hacked. Among these operations include verifying the current condition of the infection, removing the bot from IRC [16], coming up with a random username, sending ping attacks, forcing a bot to join a channel, carrying out SYN flood or DDoS attacks, and a great deal more. In the same year, Agobot revealed the concept of a modular staged attack. It was packaged as extra malicious payloads that were put on top of an underlying backdoor. The initial phase of the attack is to install the backdoor, which prepares the way for following attacks that are segmented into several smaller modules. Once this step is complete, the route is clear for the subsequent assaults. The makeover that SDBot went through in 2003 ultimately led to it being rechristened as Spybot. These additional capabilities included the recording of keystrokes and the sending of unwanted instant messages in bulk (SPIM) [17]. Rbot, a programe that worked via the use of a SOCKS proxy, was presented to the public the same year. In addition to that, it was able to carry out distributed denial-of-service assaults and steal data using a variety of different techniques. This family of bots was the first to make use of technologies such as compression and encryption to avoid being identified.

Later on, botmasters came to the realization that IRC-based C & C servers were a single point of failure because it was very simple to delete them or just ban them, which would effectively terminate the botnet. This realization was reached because that it was very simple to do either of these things. It is possible that the IRC port has been blocked on the majority of current firewalls, and it is not difficult to identify the protocol when one examines network traffic. The botmasters were forced to alter the way how they controlled the botnets to prevent themselves from being found. They took advantage of a Peer-to-Peer, or P2P, architecture in order to construct the next generation of botnets. Within the context of P2P, a botmaster will issue an instruction to one or more bots, and those bots will then communicate the instruction to the users who are located in proximity to them. If a particular bot cannot be reached, the communication will continue with the next bot on the list. As a consequence of this, the botnet grew into a tremendously strong network, making its elimination extremely difficult [18]. The term "hypertext transfer protocol" (HTTP) traffic refers to the kind of traffic that makes up most of all traffic on the Internet. Since that HTTP traffic is still quite common across all networks, the command-and-control servers for these botnets are still operational [19].

3 THREAT ANALYSIS

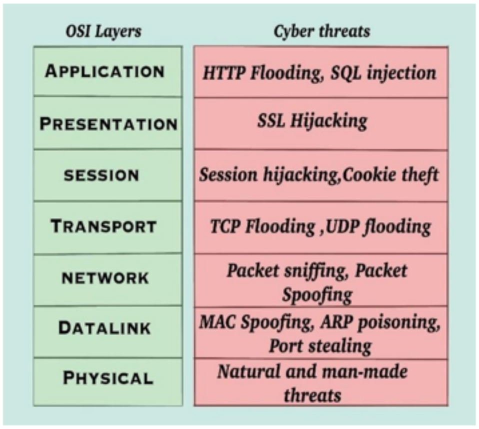
3.1 Dangers Posed by the Application Layer

The application layer in HTTPS interacts directly with the user because it manages requests and responses such as web pages, emails, and file transfers. It provides the user interface and indicates data in an approach that is understandable by individuals. HTTPS encrypts data transmission between the client and server at the application layer, enabling end-to-end encryption and secure communication between the user and server. The only layer that communicates directly with the end user is the application layer, which is also known as the presentation layer. Because it is situated in the OSI model so near to the layer that serves the end user, this layer is the most susceptible to attack. Numerous services, including e-mail, file transfers, chatting, and web surfing, are carried out at this location. The most frequent types of attacks that may take place at this layer are known as HTTP floods and SQL injections. Hackers use HTTP floods and SQL injections to gain unauthorized access to networks or systems by overloading servers or exploiting vulnerabilities in web applications. Protecting against these attacks involves implementing security measures such as firewalls, intrusion detection and prevention systems, regular security audits, and keeping software up to date. Educating users on secure computing practices is also important [20]. The objective of the Distributed Denial of Service (DDoS) attack known as an HTTP flood is to freeze the application and prevent the end user from gaining access to it. HTTP flood is a DDoS attack that overloads an application or server with excessive traffic, causing damage and financial loss. Attackers use botnets or compromised devices to launch the attack easily. Load balancing, rate limiting, and traffic filtering are mitigation measures, while regular security assessments can identify vulnerabilities. When a client wishes to connect with the server, it will often submit an HTTP request, such as GET or POST, in order to get access to the resources. This allows the client to communicate with the server.

HTTP GET Flood – A GET request is used to get and see the usual material like photos and files. This request may be used to flood websites. HTTP GET requests are important as they allow users to retrieve data or resources from a web server efficiently. GET requests are used to retrieve web pages, access APIs, and can be cached by web browsers for faster access. They are also used by search engines to index web pages, impacting a website's SEO. As a result, HTTP GET requests are a vital component of the HTTP protocol. Unauthorized sources will submit repeated HTTP GET requests to a particular server in an attempt to access the files and pictures stored on that server [21]. The server is unable to react to the request coming from the authentic source because it is too busy dealing with the many requests that are coming from the fraudulent source.

HTTP POST – A request known as a POST is used whenever there is a need to dynamically update the resources. To bring the server down, the maximum number of resources that may be allotted to a single request will be used [22].

SQL Injection is a kind of attack that is also known as SQLI. During this assault, the attacker takes advantage of application weaknesses in order to inject malicious SQL code. This gives the attacker access to the database, enables them to edit the information that is already there, allows them to carry out administrative tasks, and even grants them access to sensitive material that is stored in the database [23].



OSI Layers	Cyber threats
APPLICATION	HTTP Flooding, SQL injection
PRESENTATION	SSL Hijacking
SESSION	Session hijacking,Cookie theft
TRANSPORT	TCP Flooding ,UDP flooding
NETWORK	Packet sniffing, Packet Spoofing
DATALINK	MAC Spoofing, ARP poisoning, Port stealing
PHYSICAL	Natural and man-made threats

Figure 1. Cyber threats in a seven-layered OSI model

3.2 Presentation Layer Threats

This layer is also sometimes referred to as the Syntax or Translation layer. Its purpose is to provide the application layer with data that is presented in a way that is easier to understand. Encoding and decoding, data encryption and decryption, and data compression are the operations that the presentation layer is responsible for carrying out [24]. The Secure Socket Layer (SSL) is a means for encrypting data that is sent over the internet. SSL is a protocol which provides secure and encrypted connections between web servers and client browsers to avoid attackers from stealing or copying sensitive data. Data is encrypted using public key encryption, and the web server's identity is examined. SSL improves website SEO, encourages customer loyalty, and establishes trust between websites and users, all of which increase sales and revenue for businesses. SSL is prioritized over HTTP in HTTPS. SSL is not preferred over HTTPS because, due to known vulnerabilities, SSL has been deprecated in favour of TLS. HTTPS is the recommended method for secure data transmission over the internet because it provides authentication, encryption, and trust between a web server and a user's browser. HTTPS validates the identity of a website, encrypts data transmission, and is recognized by many as a reliable and secure protocol. At the beginning of a connection between a client and a server, the server will give the client a certificate that includes the digital signature to demonstrate that it is an authorized identity [25]. Therefore, in a hijacking, the communication that is supposed to be going between the client and the server is instead intercepted by the attacker, who then attempts to pose as a genuine service. Therefore, the client and the server no longer have a secure connection, which is indicated by the HTTPS protocol [26].

3.3 Dangers at the Session Layer

Synchronization is handled at the OSI session layer, which is where the responsibility lies. This layer is responsible for the initiation of sessions between end-user application processes, as well as their management, acceptance, and closure. It also serves the purpose of a token manager by prohibiting two different users from carrying out the same activity at the same time [27].

Session hijacking is one of the risks that are present at this layer. Session hijacking occurs frequently at the OSI session layer because it manages session-related tasks, such as authentication and authorization, and uses session IDs or tokens that can be intercepted by attackers. To prevent session hijacking, it is important to use secure protocols, implement secure coding practices, and take precautions such as using strong passwords and avoiding public Wi-Fi networks. An adversary has taken control of the session and hijacked the user's session in this instance. The user has lost control of the session. Cross-site scripting is an attack technique that hackers use to take over users' sessions. An example of a client-side injection attack is known as cross-site scripting (XSS). This kind of attack involves injecting malicious JavaScript code into the program that runs on a web page [28]. The server delivers the code for the page, which now contains the injected script, when the victim attempts to view it. The browser used by the end-user is the one that runs the malicious script since it believes it came from a reliable source. The session cookie is automatically sent to the attacker by the browser being used by the victim. The script that is launched has the potential to access cookies, see sensitive information stored in the browser, and even change the text of an HTML page [29].

Theft of cookies – a cookie is a piece of data that is shared by a user session between a website and that user's session. A cookie will always be tracked and stored by the server for user personalization. In this scenario, the attacker prevents the cookies from reaching the server by seizing control of the session ID that is now valid [30]. Cookies can be used to track a user's online activity, such as the websites they visit, the products they view, and the advertisements that they click on. This is typically accomplished using third-party cookies to collect data and establish a profile of the user's actions, which can then be used for personalized advertising or audience insights. However, users can limit cookie tracking by eliminating their browser history or selecting website tracking.

3.4 Dangers at the Transport Layer

Since it provides an end-to-end link between the two processes that are running over the network, the transport layer is sometimes referred to as the "heart" of the OSI model [31, 32]. It does it by segmenting data and incorporating error-control methods into its operations. Error-control methods in the OSI model ensure reliable data transmission by detecting and correcting errors at the data link layer. They can detect errors using checksum or CRC, correct errors with forward error correction, or request retransmission of the data frame. These methods are necessary to minimize errors and ensure correct data transmission over unreliable communication channels. The connection-oriented and connectionless modes of transmission are the two options available to users at the transport layer. Connection-oriented transmission establishes a dedicated logical connection before data transfer, ensuring reliable delivery of data but with increased overhead. Connectionless transmission, sends data as independent packets without a dedicated connection, providing faster delivery but with less reliability. The mode used depends on the application's specific requirements. Transmission Control Protocol, often known as TCP, and User Datagram Protocol (UDP) are the two protocols that are used the most frequently in the transport layer. Both of these protocols are susceptible to attack by malicious cyber actors. TCP and UDP are both vulnerable to attacks from malicious actors due to errors in their design and the applications that use them. While UDP lacks error checking and flow control, making it easier for attackers to take advantage of application vulnerabilities, TCP is susceptible to SYN flooding, TCP reset attacks, and session hijacking. Attackers can also intercept and modify data packets by using packet sniffing and spoofing. Strong security measures are required to ensure the confidentiality, integrity, and availability of data transmitted over these protocols to minimize these risks. These measures include firewalls, encryption, and intrusion detection and prevention systems.

An example of a DDoS attack is known as a TCP SYN flood. A SYN flood attack takes advantage of an error in the TCP three-way handshake process, causing a server to become overloaded and unable to respond to legitimate requests. This may result in server downtime, resource depletion, financial damage, and increased security risks. Mitigating these attacks necessitates the implementation of appropriate security measures such as firewalls, intrusion detection and prevention systems, and load balancers. The primary objective of this attack is to prevent genuine traffic from accessing the server by using all available resources in a manner that renders the service inaccessible. The SYN flood attack is effective because it takes advantage of the three-way handshake that occurs inside a TCP connection.

1. The attacker uses the faked IP address to send many SYN packets to the server that is the target of the attack. Attackers use IP spoofing, botnets, amplification,

and reflection techniques to send a large volume of SYN packets to a server using a fake IP address. Implementing access control lists and anti-spoofing technologies can help prevent or mitigate these attacks.

2. As a kind of reciprocity, the server will reply to all requests and will ensure that an open port is available to receive the ACK. The server is now waiting for the ACK from the client, but the client never sends it.
3. The server keeps the port open for an extended length of time, which eventually results in the server going down and causing genuine clients to be prevented from receiving service.

The DoS attack known as UDP Flood involves sending a large number of IP packets containing UDP datagrams to the server that is being targeted in order to render it inaccessible. UDP is more susceptible than TCP because, unlike TCP, it does not need an initial handshake to create a connection. TCP and UDP are transport layer protocols used for communication between devices on a network. TCP is connection-oriented, provides reliability, is commonly used for applications that require error-free transmission and has a larger header size. UDP is connectionless, faster, does not provide reliability, is commonly used for applications that require speed and real-time communication and has a smaller header size. The choice between the two protocols depends on the specific requirements of the application being used. The server will employ its resources and look for the applications that are presently listening for the request with a certain port whenever it gets a UDP packet. The port number is supplied in the request. After receiving a large number of UDP packets, all the resources will be used up, which will cause a Denial of Service condition for the traffic that is legal.

3.5 Dangers Posed by the Network Layer

The network layer is the most important part of the OSI model since it is responsible for determining the most efficient logical route that a data packet may take to go from its origin to its destination. The network layer is susceptible to several types of risks, which can compromise network security and stability. Common risks include network intrusion, DoS attacks, routing attacks, ARP attacks, IP spoofing, and malware infections. To prevent potential threats, robust security measures and protocols such as firewalls, encryption, and intrusion detection systems are necessary to safeguard the network. Since the network layer is directly connected to the internet, it is more susceptible to harmful attacks than the lower layers. Attacks like IP spoofing, route poisoning, and denial-of-service attacks can be carried out by attackers by using the advantages of these vulnerabilities. To reduce these risks, network administrators need to put in safety features such as firewalls, intrusion detection and prevention systems, and VPNs, as well as conduct regular security audits to identify and address vulnerabilities.

The process of detecting and monitoring network traffic while simultaneously collecting data packets that are being sent over a network is known as "packet sniffing". Packet sniffing involves capturing and filtering network traffic to analyze data packets and identify both lawful and malicious activity. The process includes analyzing the contents of each packet and can be used to diagnose network issues, optimize performance, and detect security incidents. However, it is important to use packet sniffing for legitimate purposes only and to implement appropriate security measures to prevent unauthorized use. In order to get access to secret information, attackers will sniff the network in search of sensitive data that is either not encrypted or just partially encrypted and will steal this data. Depending on the accessibility mode, packet sniffing may or may not be considered lawful activity.

Spoofing the identities of data packets is what is meant by the term "packet spoofing". The data that is sent over the network will be divided up into many different packets. Each packet has its own IP header, which includes information about the packet as well as the source IP address and the destination IP address. Hackers make use of the programme to generate IP packets with a changed source address, after which they attempt to interact with the server by masking its IP address and using an IP address that is known to be trustworthy. IP spoofing like this will always result in DDoS assaults. IP spoofing involves falsifying the source IP address in IP packets to impersonate another computer system. Cybercriminals use this technique to carry out malicious activities undetected and bypass security measures that rely on IP blacklisting.

3.6 Dangers Posed by the Data Link Layer

The primary function of the data link layer is to carry out framing, which is the process of attaching the data's source MAC address and its destination MAC address. Also guarantees the flow of data that is free of errors. The data connection layer is composed of two sublayers in its structure:

1. Logical Link Control (LLC),
2. Media Access Control (MAC).

The MAC layer manages access to the physical network and adds a header to data packets, while the LLC layer provides flow control, error checking, and sequencing of frames. Together, these layers control data flow and ensure reliable and efficient transmission between devices on the same network.

There are a few potential dangers that might occur at this layer, including ARP poisoning, MAC flooding, and port theft.

ARP poisoning is a kind of the Man-in-the-Middle attack that enables the attacker to disrupt the communication which is taking place between the devices. ARP is used to convert Internet Protocol (IP) addresses into Media Access Control (MAC) addresses. This is the program's primary purpose. An attacker using ARP poisoning will broadcast bogus ARP packets over a Local Area Network (LAN) in order to connect their MAC address to a valid IP address. Now the attacker is in a position to receive any communications that are sent to valid MAC addresses. As a consequence of this, the adversary is in a position to be able to intercept, change, or stop the communication directed to the MAC address of the victim.

The next step is known as MAC flooding. In most cases, switches will keep a "MAC table" that is comprised of the MAC addresses of the hosts that are connected to each port on the switch. The attacker will continue to send bogus source MAC addresses to the switch until the table is completely filled. Now the switch is confused, and instead of functioning as a switch, it is acting as a hub by broadcasting the frames to all of the associated hosts.

The term "port stealing" refers to an attack in which the perpetrator takes the traffic that is intended for another section of the ethernet switch. In most cases, switches are able to discover the MAC address and associate it with the appropriate port. Attacks that take use of ethernet switches may be called port theft attacks. The attacker sends a flood of counterfeit frames from a separate port that include the victim's MAC address disguised as the source address. At this point, the data frames that were supposed to be transferred to the genuine port are being sent to the attacker's port instead.

3.7 Dangers Posed by the Physical Layer

The "physical" layer is the one that manages all of the "physical" components. Wires, copper cables, fibre optic cables, routers, endpoints, and sockets are the components that link everything to make a real network. Other components include fibre optic cables. The dangers posed by this layer include the interruption of electric signals that are sent between the network nodes, the cutting of cables in a physical manner, insufficient power, and damages caused by natural catastrophes such as short circuits. In this layer, there is also the potential for man-made disasters on purpose, such as the destruction of the devices, theft, and signal jamming.

4 PROPOSED MODEL

After doing an extensive research on HTTP botnets such as SpyEye, Zeus, Athena, BlackEnergy, and Andromeda, we have discovered that some Simple Network Management Protocol - management information base SNMP-MIB variables undergo substantial transformations whenever an HTTP bot make contacts with its command and control server.

Feature Extraction. The C & C server will receive a significant number of HTTP requests from the HTTP botnet as they are sent automatically. These HTTP requests use valid forms and are sent over standard TCP connections; as a result, it might be difficult to pinpoint where these requests are coming from. In addition, the queries made by bots are often created at random or by repeatedly

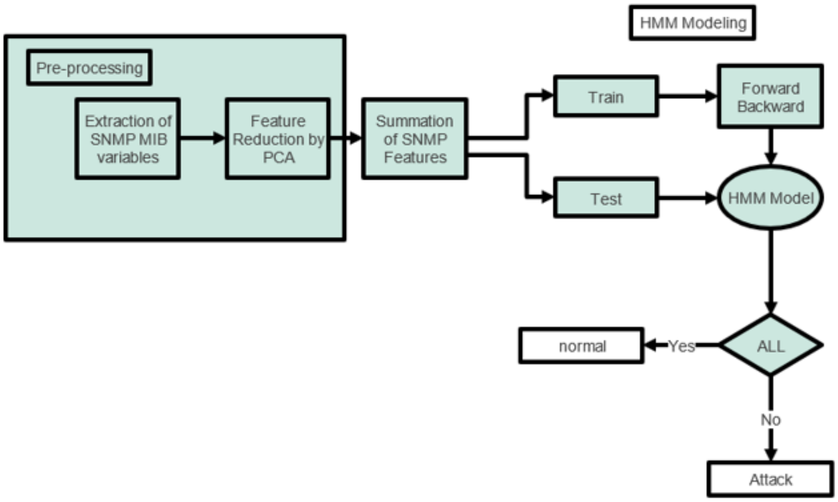


Figure 2. Block diagram of the proposed HsMM model

making a small number of straightforward HTTP requests across TCP connections.

Principal Component Analysis (PCA) is a statistical method that transforms correlated variables into uncorrelated ones called principal components. The components are ranked by importance, with the first accounting for the most variance in the data. PCA is used to simplify data, reduce noise, and identify patterns in machine learning, data analysis, and preprocessing. PCA is performed on the gathered MIB variables, in order to determine which SNMP-MIB variables are important. It is among the dimensionality reduction methods that is commonly quite often utilised for analysis of data and compression. Dimensionality reduction is popular because it simplifies complex data by reducing the number of variables. It improves analysis and visualization, and helps identify patterns and relationships that may not be visible in high-dimensional space. It is useful in fields such as image and natural language processing. It works by identifying a small number of orthogonal linear relationships of the set of variables that had the highest variance in order to reduce a relatively large number of variables into a smaller number of uncorrelated variables. Dimensionality reduction, which involves reducing a high number of variables into a smaller number of uncorrelated ones; it aims to simplify and optimize the analysis of complex datasets while retaining important patterns. It improves machine learning performance, reduces computational costs, and enhances data interpretability and visualization. Additionally, it eliminates redundancy and noise in the data, leading to more accurate analysis. One of the most widely popular methods for dimensionality reduction is this one.

$$\left.\begin{aligned} p_1 &= \mu A_1(x) \\ p_2 &= \mu A_2(x) \end{aligned}\right\} \tag{1}$$

The output response of layer 1 for feature set 2 is

$$q_j = \mu B_j(y), \tag{2}$$

where

- j represents number of nodes in layer 1 for feature set 2 and in this research work,
- j varies from 1 to 2 and hence Equation (3) is represented as the output responses of layer 2:

  $$w_i = \mu A_i(x) \cdot \mu B_i(y), \tag{3}$$

  where

  - i represents the number of nodes in layer 2 for feature set 1 and in this research work,
  - i varies from 1 to 2 and hence Equation (4) is represented as

  $$\left.\begin{aligned} w_1 &= \mu A_1(x) \cdot \mu B_1(y) = p_1 \cdot q_1 \\ w_2 &= \mu A_2(x) \cdot \mu B_2(y) = p_2 \cdot q_2 \end{aligned}\right\} \tag{4}$$

  The output responses of layer 3 are

$$\begin{aligned} \overline{w_l} &= \frac{w_l}{w_1 + w_2}, \\ \overline{w_1} &= \frac{w_1}{w_1 + w_2}, \\ \overline{w_2} &= \frac{w_2}{w_1 + w_2}. \end{aligned} \tag{5}$$

The output response of layer 4 is

$$\left.\begin{aligned} k_i &= \overline{w_l} \cdot f_i \\ k_1 &= \overline{w_1} \cdot f_1 \\ k_2 &= \overline{w_2} \cdot f_2 \end{aligned}\right\} \tag{6}$$

The output response of layer 5 is

$$\begin{aligned} f &= \sum_{i=1}^{2} \overline{w_l} \cdot f_i, \\ f &= \overline{w_1} \cdot f_1 + \overline{w_2} \cdot f_2. \end{aligned} \tag{7}$$

Constructing a HsMM in order to generate a profile of the typical behaviour of MIB traffic is the first step in developing the model that will be used to identify the botnet. MIB traffic analysis can identify botnets by detecting specific network communication patterns between infected devices and their command-and-control servers. The MIB provides a standardized way to monitor and manage network devices, and analyzing MIB traffic patterns can help detect botnet infections. Security analysts can detect and mitigate botnets by monitoring and analyzing MIB traffic. Because this model only takes into account two distinct states, the state space is denoted by the notation $F = 0$ and 1, with 0 denoting the state of the system when it is operating normally and 1 indicating that the system is operating under the direction of the botmaster.

Among the different clustering techniques, the k-means algorithm is one of the major and well-known that uses distance measurements to determine the groups (Euclidean distance metrics). This strategy, among others, is straightforward yet effective. In accordance with the needs of the user, it divides the data into k clusters. The k-means algorithm is a popular and efficient clustering method that outperforms other techniques in terms of scalability, ease of implementation, flexibility, and performance. Its ability to handle large datasets, customize to different applications, and produce accurate results has made it an important tool in data analysis and machine learning. With k-means, data is processed spatially, and after each cluster, the centroid is determined using a statistical technique (mean). Such clustering methods choose random centroids and adjust themselves based on the cluster's circumstances and its divergence (objective function). These algorithms are known as unsupervised because of this. Iterations continue once the procedure starts until the convergence state is reached. The minimization of the distance between the measurement points and the cluster centres is shown by this convergence stage. The number of sub-optimal states in the k-means approach varies depending on the initial value picked.

The k-means' objective function J is given by

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ij} \left\| x^i - \mu_k \right\|^2, \tag{8}$$

where μ_k is the k^{th} cluster midpoint, m is number of data points, x^i is the data point in i^{th} vector, K is total number of clusters well-defined by the user, $w_{ij} = 1$ for data point x_i if it fits in to cluster k; else, $w_{ij} = 0$.

The FCM (Fuzzy C-Means) method, which has a benefit over k-means for its ability of detecting mutual clusters, was developed using the k-means technique as a base (i.e., identifying data points that may belong to two or more clusters). FCM is an iterative technique that advances on minimizing the objective function, similar to the k-means approach. The square error analysis is used to determine the convergence process. The convergence process in k-means clustering is evaluated using square error analysis, where the sum of squared distances between data points and assigned cluster centroids is calculated. The k-means algorithm aims to minimize the SSE by iteratively updating cluster assignments and centroid positions until convergence. The SSE is used to evaluate the clustering solution and improve its quality. This method is commonly used to assess the convergence process in k-means clustering. This approach activates and starts to cluster utilizing random picks since it follows the essence of unsupervised learning. The three steps of the FCM process are as follows:

1. Cluster Center,
2. Membership function, and
3. Objective function.

The FCM's objective function is provided by

$$J_{\text{iter}} = \sum_{i=1}^{N} \sum_{j=1}^{nc} \left[M^f \left\| D_{x_j - c_j} \right\|^2 \right], \tag{9}$$

where nc is the number of clusters, and N denotes the number of data points in the data vector. (The user may provide this in accordance with their needs.) M^f is a membership function given by

$$M^f \left(\frac{d_{ji}}{d_{ki}} \right) = M_{iter} = \frac{1}{\left[\sum_{k=1}^{nc} \left(\frac{d_{ji}}{d_{ki}} \right)^{\frac{1}{f-1}} \right]}, \tag{10}$$

where $\|D_{x_i} - C_j\|$ and $\left\|D_{x_j} - C_k\right\| - f$ is a fuzzification factor, and d_{ji} and d_{ki} are the distance measurements (Euclidean distance technique is employed) among the data point D_{x_i} and the cluster center C_j. C_j is a cluster centre that is often taken from the M^f provided by

$$C_j = \frac{\sum_{j=1}^{N} M^f D_{x_j}}{\sum_{i=1}^{N} M^f}. \tag{11}$$

The Fuzzy C-Means (FCM) and k-means clustering algorithms differ in their approach to cluster membership, centroid calculation, sensitivity to outliers, computational complexity, and tuning parameters. While FCM allows data points to belong to multiple clusters based on membership degrees, k-means assigns each point to a single cluster; meantime, FCM is less sensitive to outliers and more complex than k-means.

The FCM procedure is as described below.

1. The parameters nc, f and condition of the convergence state S_c are defined in order to initiate the FCM process.
2. Because the approach is unsupervised, the FCM starts by choosing a random cluster centre M^f_{rand} or a random membership function C_j. In this instance, M^f_{rand} is picked for additional processing.

3. The current M^f's Equation (11) is used to determine C_j.
4. At this point, the objective function J_{iter} is determined utilizing Equation (11) and its convergence is examined as described below.

$$J_{\text{iter}} - J_{\text{iter}-1} < S_c, \tag{12}$$

   i.e., the difference among the value produced in the current iteration's goal function and the prior iterations is measured after every iteration. The S_c is used to validate this difference.
5. The loop is terminated and the current M_f is taken into consideration as the output if the convergence state has been reached; otherwise, the current M_f is replaced with the new M_f (i.e., M^f_{new} derived utilizing Equation (12) based on the current C_j). Up till the goal is accomplished, the iteration is performed.

Even while FCM yields superior outcomes, the constraint brought on by local optima makes it more difficult to achieve the convergence state and, occasionally, it never does. This is the basis for the introduction of the fuzzification factor f.

Numerous algorithms were developed based on FCM to get around the drawbacks.

$$J_{\text{iter}} = \sum_{i=1}^{N}\sum_{j=1}^{nc}\left[M^f \left\|D_{x_i} - C_j\right\|^2\right] + \frac{a}{N_R}\sum_{i=1}^{N}\sum_{j=1}^{nc}\left[M^f \left\|D_{x_r} - C_j\right\|^2\right], r \in N_i, \tag{13}$$

where

- a is a constant that is employed for boosting or adjusting and
- N_R is the cardinality.

Let the collection of recognizable symbols, that are nothing more than the summation of SNMP-MIB variables at various time intervals, be denoted by the notation V = "v0, v1, v2ams XM-1". The visible symbol distribution is denoted by the equation $B = bi(k)$, where $i > F$ and $k > M$

The distribution at the beginning of the state. P is equal to $[P0, P1]$, and it is a given that $P0 = 1$ and $P1 = 0$ due to the fact that the model was developed for a perfect normal process at the beginning.

5 TRAINING PHASE

Following the construction of the model, we will need to train the model so that we can determine the parameters of the HsMM. In order to keep the most accurate parameters of a legal HsMM model, the typical forward-backward training procedure is utilised. The forward-backward training procedure, or backpropagation algorithm, is a method used to train neural networks. It involves propagating input data forward through the network, computing the output, calculating the error between the actual and desired output, backpropagating the error through the network to update neuron weights, and repeating the process iteratively until the error is minimized or a stopping criterion is met. This procedure helps the network learn from errors and improve its accuracy. The parameters of the HsMM are first calculated, and then they are re-evaluated in such a way that the ALL function gradually grows until it reaches its highest possible value. ALL may be calculated as follows, assuming that we have the sequence O1: T and the model: parameter.

$$M^f = \frac{\left(\|D_{x_i} - C_j\|^2 \frac{a}{N_n} \sum_{r\in N_i} \|D_{x_r} - C_j\|^2\right)^{\frac{1}{f-1}}}{\sum_{M=1}^{c} \left(\|D_{x_i} - C_j\|^2 \frac{a}{N_R} \sum_{r\in N_i} \|D_{x_r} - C_j\|^2\right)^{\frac{1}{f-1}}}, \tag{14}$$

$$C_j = \frac{\sum_{i=1}^{N} M^f \left(D_{x_i} + \frac{a}{N_n} \sum r \in N_i D_{x_r}\right)}{(1+a)\sum_{i=1}^{N} M^f}. \tag{15}$$

Equations (14), (15), and (16) provide the cluster centre membership function, and goal function for the FCM_S variant. The settings of FCM_S are changed to create new versions of the algorithm, such as FCM_{S1} and FCM_{S2}.

$$\varepsilon_i = \frac{1}{1+a}\left(D_{x_j} + \frac{a}{N_R}\sum_{j\in N_i} x_j\right). \tag{16}$$

In order to create an EnFCM (Enhanced FCM), the D_{x_i} is further substituted with ε_i where ε_i lessens the clustering process's temporal complexity, where the i^{th} data matrix value is taken into account as the centre of the local window (p_i, q_i), and the j^{th} data matrix value is taken into account as the neighbour of the centre. For efficient operation, the scaling factors (λ_s and λ_g) are introduced in this work.

$$\sigma_i = \sqrt{\frac{\sum_{j\in N_i} \|x_i - x_j\|^2}{N_R}},$$

$$\varepsilon_i = \frac{\sum_{j\in N_i} s_{ij} x_j}{\sum_{j\in N_i} s_{ij}}, \tag{17}$$

where S_{ij} indicates how equivalent a data point is to its neighbour in order to identify the gradient within it. After you have obtained the parameters, you can use HsMM to assess if the supplied observation sequences of SNMP-MIB variables belong to a normal profile or a bot profile by calculating their respective average log likelihoods. This may be done after you have obtained the parameters. The forward-backward technique is used in order to compute the probability that a given sequence would match a particular model. COMPUTATION is performed on EVERY value of the training sequences for BOTH the standard system and the botnet system. In the event if the data from all of the observations

If the figure in question is found to fall within a certain confidence interval, the profile will be regarded as normal. In a similar manner, the confidence interval for botnets is determined. If the whole observation sequence falls within this confidence interval, then the event in question will be categorised as botnet communication.

6 DATASETS USED

The experimental setup allows the deployment of SpyEye, BlackEnergy, Zeus, Athena, and Andromeda botnets. Zeus, ZeuS, or Zbot is a Trojan horse that steals financial information through form snatching and man-in-the-browser keyboard recording. It is commonly used for banking theft and CryptoLocker malware installation. Most Zeus infections occur through drive-by downloads and phishing scams. Since 2011, Zeus malware construction kit's capabilities have been combined into SpyEye, a sophisticated and deadly malware kit. SpyEye, a banking virus, poses global problems and is difficult to identify and eliminate on infected Windows machines. Athena is a C++ DDoS botnet that targets Windows PCs with sophisticated strategies, targeting web, gaming, VoIP, and residential connections.

The Russian hacker underground employs BlackEnergy, a web-based DDoS network, to launch various attacks. The botnet is user-friendly and command-friendly, with minimal syntax and structure. It connects to its controlling servers via HTTP, enabling easy communication between the attackers and the servers. Andromeda, a botnet using the HTTP protocol, is a popular distribution channel for malicious actors. It is only downloaded on 3.8 % of infected computers, with PoS malware being primarily created for stealing credit card data from POS systems. The hackers behind GamaPoS are targeting POS systems among the massive number of compromised PCs, as most machines infected with Andromeda backdoors do not currently run PoS software. The feature extraction component is built using Java, and the SNMP setup tool is installed. MIB variables are gathered from zombie machines for seven and twelve hours daily, with outgoing and incoming traffic contributing to their updating. Table 1 provides information on the datasets.

Botnet MIB Traces			
Botnet	MIB Trace size	Botnet	MIB Trace size
SpyEye	1.25 GB	Athena	2.73 GB
BlackEnergy	2.96 GB	Andromeda	4.59 GB
Zeus	2.57 GB		
Normal MIB Traces			
FTP service	4.95 GB	Web service	4.27 GB
E-mail service	3.28 GB	Remote service	2.90 GB

Table 1. Description of the MIB datasets

The fluctuation of MIB variables seen during regular and botnet communications may be seen in Figures 3 and 4. On the x-axis, each of the figures is shown at a time interval of thirty seconds.

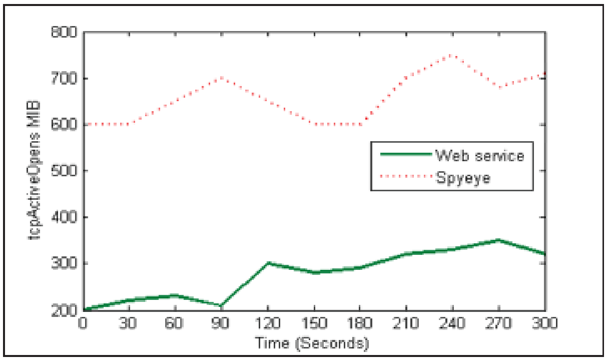


Figure 3. SpyEye and the web service's tcpActiveOpens MIB

Figure 3 demonstrates that there has been a substantial shift in the value of the tcpActiveOpens MIB variable held by the SpyEye botnet in comparison to the value held by the typical web service.

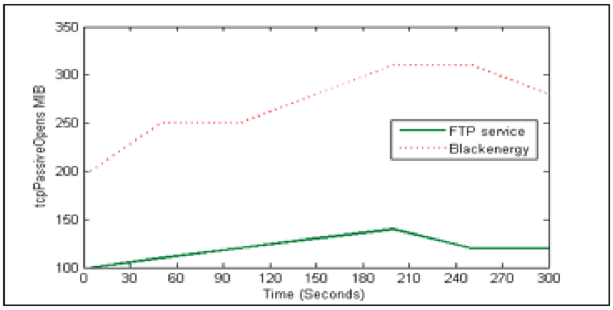


Figure 4. tcpPassiveOpens MIB in FTP service and BlackEnergy

When compared to the values of the standard FTP service's tcpPassiveOpens MIB variable, the tcpPassiveOpens MIB variable values of the BlackEnergy botnet show a substantial difference from those of the normal FTP service. This can be deduced from Figure 4. During the first stages of the bot's spread over the network, the zombie machine attempted to establish a line of communication with a distant site in the hopes of obtaining the address of the C & C server or information from the botmaster. The value of the MIB variable known as tcpPassiveOpens has significantly shifted as a consequence of this behaviour.

Comparison of the tcpCurrEstab MIB variable used by the Email service and the Athena HTTP botnet can be seen in Figure 5. One may deduce, based on the

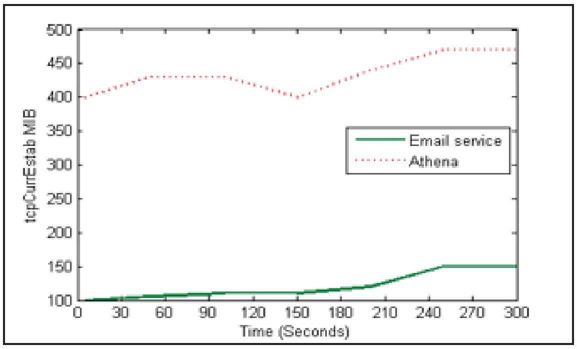


Figure 5. tcpCurrEstab MIB in Email service and Athena

graphic, that the tcpCurrEstab MIB variable undergoes considerable modifications over the course of the Athena HTTP bot propagation.

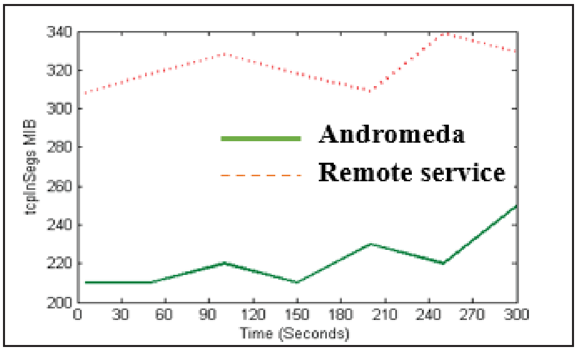


Figure 6. tcpInSegs MIB in remote service and Andromeda

A comparison of the tcpInSegs MIB variable used by Remote service and the Andromeda HTTP botnet can be seen in Figure 6. One can deduce from the figure that there is a significant shift in the value of the tcpInSegs MIB variable that occurs when the Andromeda HTTP bot is in the process of propagating.

Web botnets do not keep a connection with the command and control server, but they do routinely make web requests in order to retrieve the instructions at predetermined intervals. On the other hand, the activities of the botnet are entirely controlled by the computer and have nothing to do with the actions of the users. The observation sequence Ot indicates the summation of chosen SNMP-MIB variables count computed during the t^{th} second in the current state at the host machine while the bot propagation scenario is being played out. In this case, the observation period is thirty minutes, and the summation of the SNMP-MIB variables is computed once every thirty seconds. This is done whenever there are substantial changes in the present state of those variables. The research and experimental analysis conducted were both point to the duration of thirty seconds as the appropriate reaction time from the botmaster, as shown in Figure 6.

7 EXPERIMENTAL RESULTS

An HsMM model is trained using 70 percent of the datasets from both the normal and botnets after the important SNMP-MIB variables from both the normal and bot propagation systems have been extracted. For instance, we began by instructing the HsMM model using seventy percent of the combined web service and SpyEye dataset. The HsMM model that has been trained is next evaluated using the SpyEye dataset that has not been trained. Experiments are carried out in the same manner with several additional datasets as well. When analysing the suggested model, accuracy serves as the primary statistic that is considered. The findings of the detection accuracy are shown in Table 2. The detection accuracy is rather high, and the rate of false positives is quite low.

After getting the parameters of the HsMM, the next step is to compute the observation sequences for ALL of the regular traffic, followed by the observation sequences for ALL of the botnet traffic. It has been determined that the level of confidence interval for ALL at the 5 % level is $[-2.5, -0]$ when the system is operating normally and EVERYTHING falls inside the range of confidence that spans 5 %. When the system is talking with the C & C server, the values $[-7.5, -4.0]$ are shown.

Datasets	FPR	Detection Accuracy	Results
FTP service	0 %	100 %	Normal
Web service	0 %	100 %	Normal
SpyEye	1.67 %	98.14 %	Botnet
BlackEnergy	1.58 %	98.72 %	Botnet
Zeus	1.75 %	98.02 %	Botnet
Athena	1.29 %	98.94 %	Botnet
Andromeda	1.47 %	98.62 %	Botnet

Table 2. Performance of the proposed HsMM model

Our model was validated using a Windows computer with the settings Intel (R) CoreTM i5 3317U, 1.70 GHz, and the operating system Microsoft Windows 2000. The validation was performed using a variety of genuine botnets as well as standard data. In order to evaluate the effectiveness of the suggested model, we have made use of R, a statistical tool that is freely available online.

R serves as both a programming language and a platform for statistical analysis and visualisation. It supports a wide variety of statistical techniques, including as time-series analysis, classification, clustering, graphical approaches, linear and nonlinear modelling, conventional statistical tests, and more. In addition to that, it has a very flexible design.

7.1 Running Time

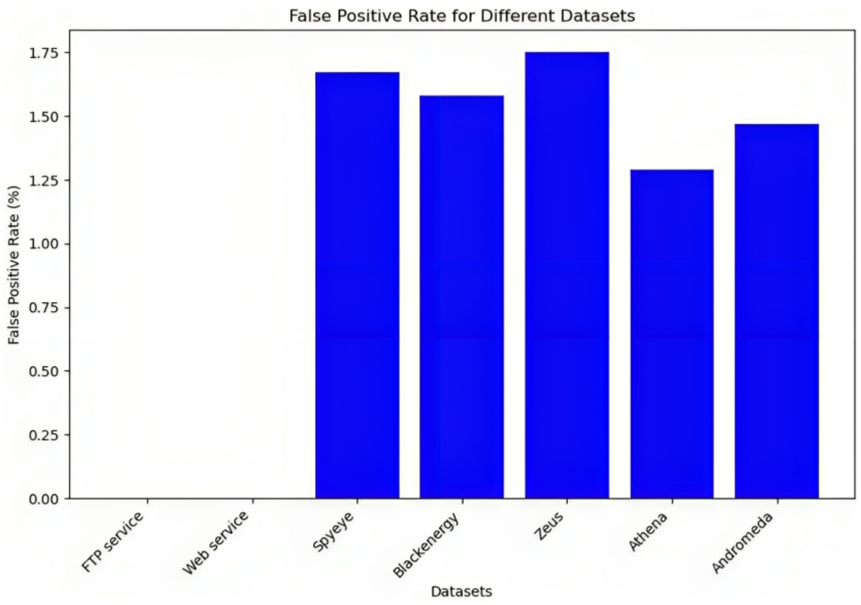
The performance analysis of the suggested model is shown in Figure 7, which presents two important metrics for various datasets: FPR and Detection Accuracy. The fluctuation of the False Positive Rate when the model comes across various datasets is the main topic of Figure 7 a). Conversely, Figure 7 b) presents the model's Detection Accuracy across several datasets. We put the HsMM model through its paces in the system with the settings described earlier so that we could evaluate how well it handled the detection time constraints shown in Figure 8.

It has been noticed that the HsMM model classifies the test instances of web service, FTP service, SpyEye, BlackEnergy, Zeus, Athena, and Andromeda MIB datasets in a span of just 2.05 seconds, 3.10 seconds, 2.51 seconds, 2.95 seconds, 2.73 seconds, 2.84 seconds, and 3.18 seconds, respectively. According to the findings, the HsMM model provides the greatest accuracy while simultaneously reducing the amount of time required for detection.

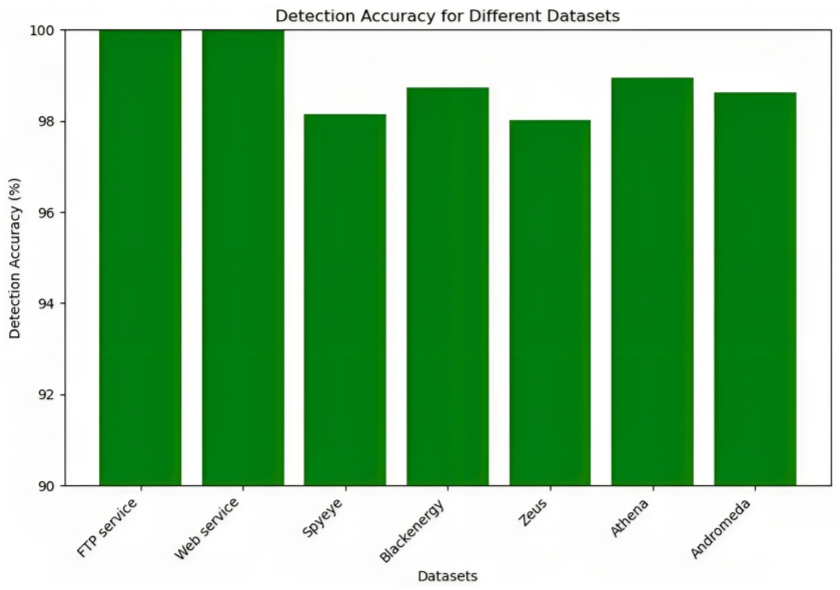
7.2 Analysis in Relation to Previous Works

The accuracy percentages of several approaches for botnet attack detection are shown in Figure 9, with each method's corresponding research publication linked. The accuracy of Naïve Bayes is 97.1 % [33]. Decision Trees have a 99.8 % accuracy rate, according to Alshamkhany et al.'s work on botnet attack detection using machine learning. Achieving a 95 % accuracy rate, Shareena et al. proposed the Deep Neural Network (DNN) technique in their work on an intrusion detection system for IoT botnet assaults using deep learning [34]. Using a deep learning-based intrusion detection system, the BotIDS approach – which was presented by Idrissi et al. in their paper – achieves a high accuracy of 99.94 % in identifying IoT botnet assaults [35]. Using machine learning and the Tree-based Algorithm, Waqas et al. show 99 % accuracy in identifying botnet assaults in Internet of Things devices via a cloud environment [14]. Last but not least, Sriram et al. research on network flow-based IoT botnet attack detection using deep learning describes the K-Nearest Neighbours (KNN) technique, which reaches an accuracy of 99.8 % [36]. These results demonstrate how several machine learning techniques, each with a distinct methodology and accuracy level, can effectively detect botnet assaults.

The M-FLMCM algorithm is an improved image segmentation technique that addresses Fuzzy Local Means' limitations by introducing a modified approach based on local standard deviation and fuzzy c-means clustering. It has shown promising results in medical image segmentation tasks but requires optimal parameter selection and input data quality. Because the M-FLMCM algorithm includes a fuzzy median method, it is one of a kind in that it has the capability to proceeds a median value from a window that contains neighbouring pixels. This enables it to achieve greater precision on usual than other algorithms. The majority of the time, the Zeus botnet were circumscribed by the area that had greater contour (because of the fact that there were not many grayscale fluctuations), as shown in Figure 9.



a) FPR with different datasets



b) Detection Accuracy with different dataset

Figure 7. Performance analysis of the proposed model

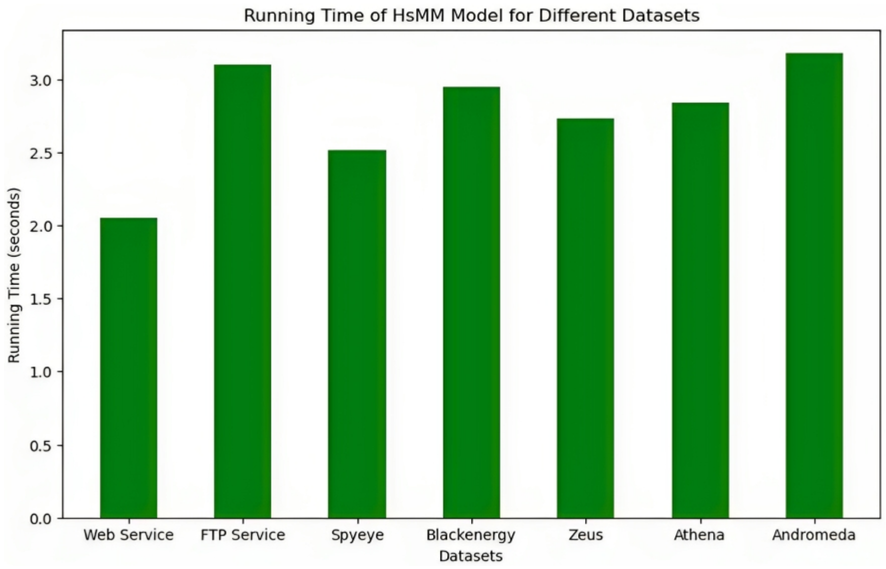


Figure 8. Detection rate

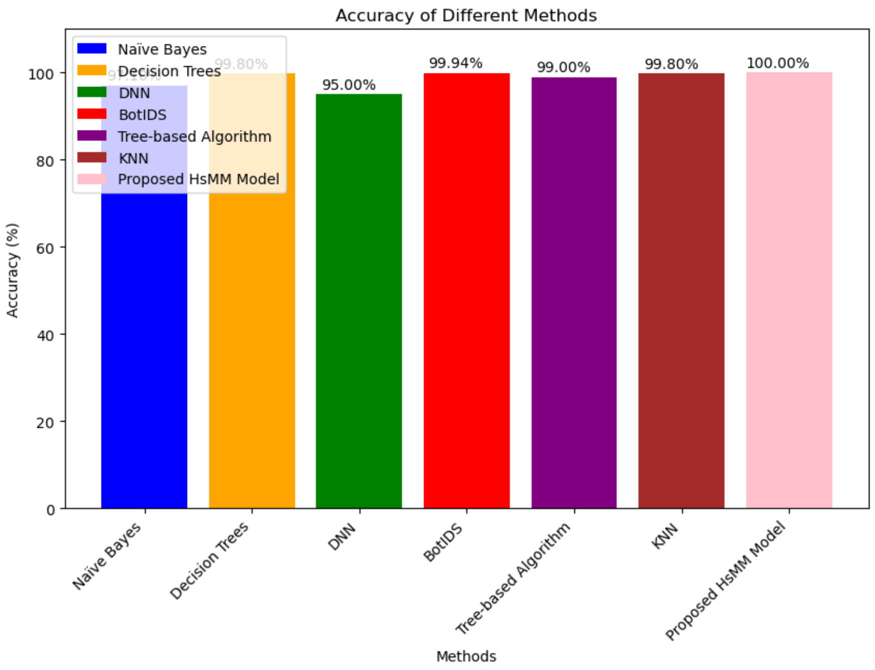


Figure 9. Accuracy comparisons with existing Zeus botnet detection techniques

Figure 10 shows the feature extracted for classifier. Nevertheless, the suggested technique was developed in such a manner that it may successfully construct clusters by identifying even minute changes in the surrounding data neighbourhood. The amount of variation, or noise, in the picture is changed in order to do research on the fundamental aspects of the algorithms. The noise variance, denoted by, provides an estimate of the noise's overall impact on the data. It does this by infecting the image's pixels in order to facilitate the classification process or the clustering of difficult data. The frequencies at which the noises, which have an effect on the data in real time, may range from high to low. In order to conduct this experiment's analysis, the high-frequency noise was collected.

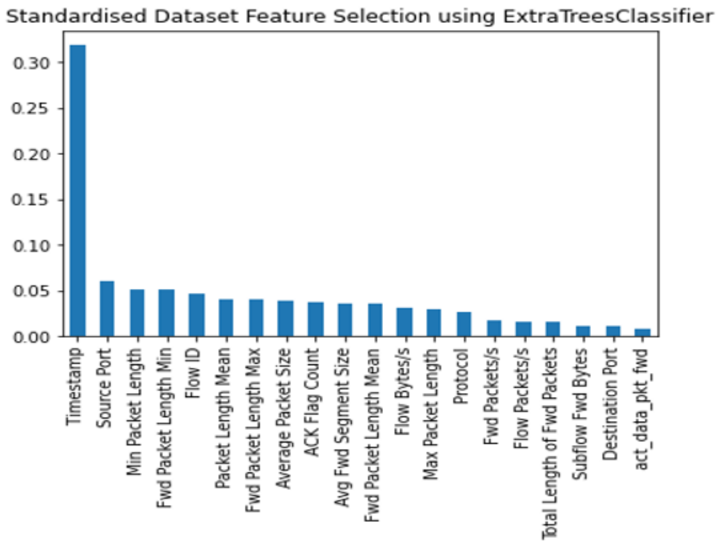


Figure 10. Feature extracted for classifier

In this experiment shown in Figure 11, the amount of noise is changed between four different levels (which can range from 0 to 1), for example, 0.2, which indicates that 20 % of the data is corrupted, 0.4, which indicates that 40 % of the data is corrupted, 0.6, which indicates that 60 % of the information is corrupted, and 0.8, which indicates that 80 % of the data are corrupted. In light of the findings of this validation, M-FLMCM may be considered to have a higher accuracy ratio than various other techniques.

The time-complexity, on the other hand, is the crucial characteristic that has to be regarded equivalent to the accuracy.

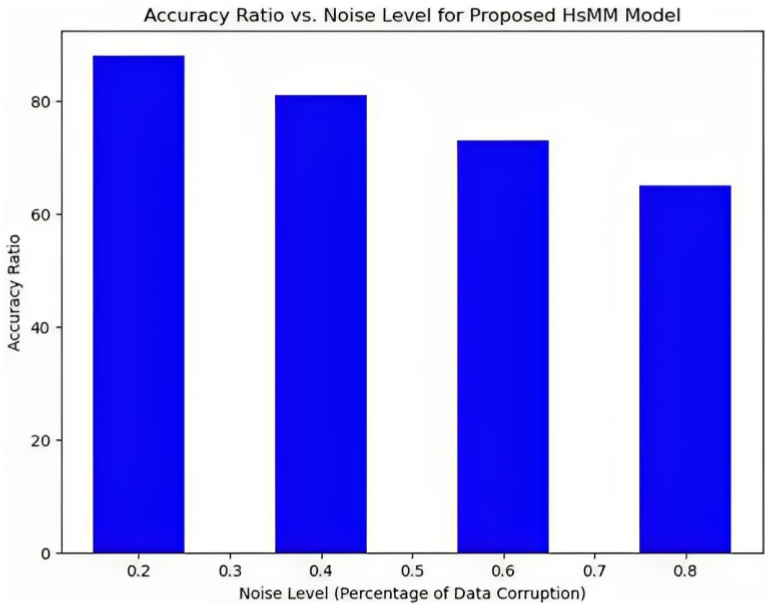


Figure 11. The amount of noise is changed between four different levels

8 CONCLUSION

A host-based botnet identification approach that makes use of TCP-based SNMP-MIB variables as characteristics is presented in this paper. This technique takes into consideration the coordination of HTTP bots within a botnet as well as the malicious behaviour of each individual bot. Utilizing a powerful forward-backward learning technique, these characteristics are plugged into the HsMM in order to get an estimation of the model parameters. The efficiency of the proposed model is shown by its high detection accuracy and low incidence of false positives. Because the suggested technique uses SNMP-MIB variables as features, it is very efficient from a computational standpoint.

The host-based techniques, on the other hand, are used to monitor specific computers to look for suspicious behaviour. This has the capability of detecting attacks while they are in progress or maybe preventing a possible attack from having any impact on the system. This method is not scalable despite the significance of host-based monitoring since it requires the provision of all computers with an extensive collection of efficient monitoring tools. On the other hand, network-based techniques get their data from the network as a whole as opposed to gathering information from each individual host. This article discusses a strategy that is built on networks. The flaws or gaps that exist in software and hardware systems are allowing cyber-attacks to become more sophisticated. The confidentiality, integrity, and credibility of the communication will all be put at risk as a result of the cyber-threats that have been covered in this article. These difficult challenges to cyber security must be adequately addressed for a variety of preventative measures and corrective steps to be performed.

REFERENCES

[1] Kumar, S.—Dalal, S.—Dixit, V.: The OSI Model: Overview on the Seven Layers of Computer Networks. International Journal of Computer Science and Information Technology Research, Vol. 2, 2014, No. 3, pp. 461–466.

[2] Suresh, P.: Survey on Seven Layered Architecture of OSI Model. International Journal of Research in Computer Applications and Robotics, Vol. 4, 2016, No. 8, pp. 1–10.

[3] Daya, B.: Network Security: History, Importance, and Future. Alpha Wireless, 2013, `http://www.alphawireless.co.za/wp-content/uploads/2013/01/Network-Security-article.pdf`.

[4] Gharibi, W.—Shaabi, M.: Cyber Threats in Social Networking Websites. CoRR, 2012, doi: 10.48550/arXiv.1202.2420.

[5] Sinha, P.—Jha, V. K.—Rai, A. K.—Bhushan, B.: Security Vulnerabilities, Attacks and Countermeasures in Wireless Sensor Networks at Various Layers of OSI Reference Model: A Survey. 2017 International Conference on Signal Processing and Communication (ICSPC), IEEE, 2017, pp. 173–182, doi: 10.1109/CSPC.2017.8305855.

[6] Abomhara, M.—Køien, G. M.: Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. Journal of Cyber Security and Mobility, Vol. 4, 2015, No. 1, pp. 65–88, doi: 10.13052/jcsm2245-1439.414.

[7] Messai, M. L.: Classification of Attacks in Wireless Sensor Networks. CoRR, 2014, doi: 10.48550/arXiv.1406.4516.

[8] Obaid, H. S.—Abeed, E. H.: DoS and DDoS Attacks at OSI Layers. International Journal of Multidisciplinary Research and Publications, Vol. 2, 2020, No. 8, pp. 1–9.

[9] Sula, E.: A Review of Network Layer and Transport Layer Attacks on Wireless Networks. International Journal of Modern Engineering Research (IJMER), Vol. 8, 2018, No. 12, pp. 23–27.

[10] Jain, K. M.—Jain, M. V.—Borade, J. L.: A Survey on Man in the Middle Attack. IJSTE – International Journal of Science, Technology and Engineering, Vol. 2, 2016, No. 9, pp. 277–280.

[11] Hasan, T.—Malik, J.—Bibi, I.—Khan, W. U.—Al-Wesabi, F. N.—Dev, K.—Huang, G.: Securing Industrial Internet of Things Against Botnet Attacks Using Hybrid Deep Learning Approach. IEEE Transactions on Network Science and Engineering, Vol. 10, 2023, No. 5, pp. 2952–2963, doi: 10.1109/TNSE.2022.3168533.

[12] Abu Al-Haija, Q.—Al-Dala'ien, M.: ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks. Journal of Sensor and Actuator Networks, Vol. 11, 2022, No. 1, Art. No. 18, doi: 10.3390/jsan11010018.

[13] Maurya, S.—Kumar, S.—Garg, U.—Kumar, M.: An Efficient Framework for Detection and Classification of IoT Botnet Traffic. ECS Sensors Plus, Vol. 1, 2022, No. 2, Art. No. 026401, doi: 10.1149/2754-2726/ac7abc.

[14] Waqas, M.—Kumar, K.—Laghari, A. A.—Saeed, U.—Rind, M. M.—Shaikh, A. A.—Hussain, F.—Rai, A.—Qazi, A. Q.: Botnet Attack Detection in Internet of Things Devices over Cloud Environment via Machine Learning. Concurrency and Computation: Practice and Experience, Vol. 34, 2022, No. 4, Art. No. e6662, doi: 10.1002/cpe.6662.

[15] Alazab, M.: A Discrete Time-Varying Greywolf IoT Botnet Detection System. Computer Communications, Vol. 192, 2022, pp. 405–416, doi: 10.1016/j.comcom.2022.06.016.

[16] Syamsuddin, I.—Barukab, O. M.: SUKRY: Suricata IDS with Enhanced kNN Algorithm on Raspberry Pi for Classifying IoT Botnet Attacks. Electronics, Vol. 11, 2022, No. 5, Art. No. 737, doi: 10.3390/electronics11050737.

[17] Almseidin, M.—Alkasassbeh, M.: An Accurate Detection Approach for IoT Botnet Attacks Using Interpolation Reasoning Method. Information, Vol. 13, 2022, No. 6, Art. No. 300, doi: 10.3390/info13060300.

[18] Pan, X.—Yamaguchi, S.—Kageyama, T.—Kamilin, M. H. B.: Machine-Learning-Based White-Hat Worm Launcher in Botnet Defense System. International Journal of Software Science and Computational Intelligence (IJSSCI), Vol. 14, 2022, No. 1, pp. 1–14, doi: 10.4018/IJSSCI.291713.

[19] Ahmed, A. A.—Jabbar, W. A.—Sadiq, A. S.—Patel, H.: Deep Learning-Based Classification Model for Botnet Attack Detection. Journal of Ambient Intelligence and Humanized Computing, Vol. 13, 2022, pp. 3457–3466, doi: 10.1007/s12652-020-01848-9.

[20] Tuan, T. A.—Long, H. V.—Son, L. H.—Kumar, R.—Priyadarshini, I.—Son, N. T. K.: Performance Evaluation of Botnet DDoS Attack Detection Using Machine Learning. Evolutionary Intelligence, Vol. 13, 2020, No. 2, pp. 283–294, doi: 10.1007/s12065-019-00310-w.

[21] Gupta, B.—Agrawal, D. P.—Yamaguchi, S.: Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security. IGI Global, 2016.

[22] Moradi, M.—Zulkernine, M.: A Neural Network Based System for Intrusion Detection and Classification of Attacks. Proceedings of the IEEE International Conference on Advances in Intelligent Systems – Theory and Applications, 2004, pp. 15–18.

[23] Shahin, M.—Chen, F. F.—Bouzary, H.—Hosseinzadeh, A.—Rashidifar, R.: A Novel Fully Convolutional Neural Network Approach for Detection and Classification of Attacks on Industrial IoT Devices in Smart Manufacturing Systems. The International Journal of Advanced Manufacturing Technology, Vol. 123, 2022, No. 5-6, pp. 2017–2029, doi: 10.1007/s00170-022-10259-3.

[24] Liu, D.—Hu, W.: Imperceptible Transfer Attack and Defense on 3D Point Cloud Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, 2022, No. 4, pp. 4727–4746, doi: 10.1109/TPAMI.2022.3193449.

[25] Kwon, H.—Lee, S.: Ensemble Transfer Attack Targeting Text Classification Systems. Computers and Security, Vol. 117, 2022, Art. No. 102695, doi:

10.1016/j.cose.2022.102695.

[26] Hegazy, H. I.—Tag Eldien, A. S.—Tantawy, M. M.—Fouda, M. M.—TagElDien, H. A.: Real-Time Locational Detection of Stealthy False Data Injection Attack in Smart Grid: Using Multivariate-Based Multi-Label Classification Approach. Energies, Vol. 15, 2022, No. 14, Art. No. 5312, doi: 10.3390/en15145312.

[27] Mukherjee, D.—Chakraborty, S.—Ghosh, S.: Deep Learning-Based Multilabel Classification for Locational Detection of False Data Injection Attack in Smart Grids. Electrical Engineering, Vol. 104, 2022, No. 1, pp. 259–282, doi: 10.1007/s00202-021-01278-6.

[28] Hu, H.—Salcic, Z.—Sun, L.—Dobbie, G.—Yu, P. S.—Zhang, X.: Membership Inference Attacks on Machine Learning: A Survey. ACM Computing Surveys (CSUR), Vol. 54, 2022, No. 11s, Art. No. 235, doi: 10.1145/3523273.

[29] Islam, U.—Muhammad, A.—Mansoor, R.—Hossain, M. S.—Ahmad, I.—Eldin, E. T.—Khan, J. A.—Rehman, A. U.—Shafiq, M.: Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. Sustainability, Vol. 14, 2022, No. 14, Art. No. 8374, doi: 10.3390/su14148374.

[30] Rafiq, H.—Aslam, N.—Ahmed, U.—Lin, J. C. W.: Mitigating Malicious Adversaries Evasion Attacks in Industrial Internet of Things. IEEE Transactions on Industrial Informatics, Vol. 19, 2022, No. 1, pp. 960–968, doi: 10.1109/TII.2022.3189046.

[31] Punitha, P.—Shanthini, J.—Karthik, S.: A Review on Cluster Based Group Adaptive Hybrid Routing Protocol for Mobility in MANET. 2018 International Conference on Soft-Computing and Network Security (ICSNS), 2018, pp. 1–6, doi: 10.1109/ICSNS.2018.8573627.

[32] Kumar, R. L.—Bhavadharini, R. M.—Karthick, S.—Punitha, P.: A Survey on Detecting Packet Dropping Attacks in MANET. International Journal of Applied Engineering Research, Vol. 10, 2015, No. 85, pp. 132–139.

[33] Alshamkhany, M.—Alshamkhany, W.—Mansour, M.—Khan, M.—Dhou, S.—Aloul, F.: Botnet Attack Detection Using Machine Learning. 2020 14th International Conference on Innovations in Information Technology (IIT), IEEE, 2020, pp. 203–208, doi: 10.1109/IIT50501.2020.9299061.

[34] Jithu, P.—Shareena, J.—Ramdas, A.—Haripriya, A. P.: Intrusion Detection System for IOT Botnet Attacks Using Deep Learning. SN Computer Science, Vol. 2, 2021, No. 3, Art. No. 205, doi: 10.1007/s42979-021-00516-9.

[35] Idrissi, I.—Boukabous, M.—Azizi, M.—Moussaoui, O.—El Fadili, H.: Toward a Deep Learning-Based Intrusion Detection System for IoT Against Botnet Attacks. IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 10, 2021, No. 1, pp. 110–120, doi: 10.11591/ijai.v10.i1.pp110-120.

[36] Sriram, S.—Vinayakumar, R.—Alazab, M.—Soman, K. P.: Network Flow Based IoT Botnet Attack Detection Using Deep Learning. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 189–194, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

Rami Mohammed Baazeem is the Head of the Management Information Systems (MIS) Department, College of Business, University of Jeddah, KSA. He is a member of many national and international committees. Also, he is the Head of the Quality Assurance unit, College of Business, University of Jeddah. He has 15 years of experience in information systems and management. He published many research papers related to e-commerce and data management. He attended many training courses in his specialization. He obtained his Ph.D. from Kingston University, UK and his M.Sc. from Griffith University, Brisbane, Australia. His research interests include data management, IT governance, e-commerce, and business intelligence.