International Journal of Computer

(IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Forensics Based Sdn in Data Centers

Oribe Zakareya Mohammed*

Faculty of Computing & Information Technology, Information System, King Abdul-Aziz University, Jeddah

Email: warmheart1166@hotmail.com

Abstract

Recently, most data centers have adopted for Software-Defined Network (SDN) architecture to meet the demands for scalability and cost-efficient computer networks. SDN controller separates the data plane and control plane and implements instructions instead of protocols, which improves the Quality of Services (QoS) , enhances energy efficiency and protection mechanisms . However, such centralizations present an opportunity for attackers to utilize the controller of the network and master the entire network devices, which makes it vulnerable. Recent studies efforts have attempted to address the security issue with minimal consideration to the forensics aspects. Based on this, the research will focus on the forensic issue on the SDN network of data center environments. There are diverse approaches to accurately identify the various possible threats to protect the network. For this reason, deep learning approach will used to detect DDoS attacks, which is regarded as the most proper approach for detection of threat. Therefore, the proposed network consists of mobile nodes, head controller, detection engine, domain controller, source controller, Gateway and cloud center. The first stage of the attack is analyzed as serious, where the process includes recording the traffic as criminal evidence to track the criminal, add the IP source of the packet to blacklist and block all packets from this source and eliminate all packets. The second stage not-serious, which includes blocking all packets from the source node for this session, or the non-malicious packets are transmitted using the proposed protocol. This study is evaluated in OMNET ++ environment as a simulation and showed successful results than the existing approaches.

Keywords: SDN; DDoS Attack; Deep Learning; Protocol; Packet ; Cloud Center ; Evidence ; Detection ; OMNET++ ; Forensics .

---

* Corresponding author

1. Introduction

Today, various networks experience attacks on daily bases, especially the largest systems due to variety and tremendous forms of attacks, which attackers use to target the integrated users and devices. Thus, efforts and perspectives need to be constantly in place for developing and innovating novel protection mechanisms [19]. Such approaches are helpful as they provide optimal defense mechanism once the network experiences attacks; however, there exists a central weakness in such methods, since they do not conduct root and origin analysis of the attack. Research indicates that approaches based on forensics in-network contributes in identifying and investigating the origins of attacks by tracking its origins [4]. Network forensics is about tracking, collecting evidence against attacks to obtain information on attacks which involves identification, properties, and nature of attacks. The exploitation of such processing previously experienced faults and attacks in networks investigations of various processes [13].

In terms of data centers, the work provenance is commonly used for traceback network traffic to identify causes of actions and activities registered through traffic, regardless of confiding investigation of the unauthorized generation source, including the unauthenticated IP headers [21]. Nevertheless, the attackers' intelligence has the potential to outperform the provenance systems by avoiding uncompromised nodes in the network, passing into a network without any sensing or tracing from forensic provenance systems' correctness node. This can lead to possibility of data leakage from datacenter, which could critically affect the sensitivity of data canter. This aspect questions the trust in the provenance of such events regarding the accurate maintenance of satisfying security system resilience requirement [5]. The most commonly experienced attacks in the data center are the DDoS attacks, where 20% of service providers reported attacks over 50 Gbps in the last year, while it constituted as one-third of attacks experienced by customers [18]. The Arista report has demonstrated that the security challenges of data centers consist of undetected attacks, including vulnerabilities in data centers, which are: intrusions, Distributed Denial of Service (DDoS) attacks, Advanced Persistent Threats (APTs), undetectable back-door break-ins, sophisticated multi-phase targeted attacks [2]. Further, [1] summarized the attackers in the data center environments, especially during communication over the network, which are the Denial of service (DoS) attacks, which isolates the network items from its original network and makes it inaccessible, where physical attacks target the hardware mechanisms and privacy attacks transmit information without evaluation.

Therefore, this study focuses on experimenting the proposed forensics using the SDN as state-of-the-art mechanism to collect substantial evidence to criminalize and track the offenders in data centers. The centers are chosen due to the conveyed comprehensive information, including sensitive data, in addition to data center exposure to various cyber-crime and attacks.

Concepts such as SDN have received significant attention and growth in recent times. This can be attributed to the evolution in the field of networking. The behavior of different components in the network can be effectively handled with centralized control. At present, the world digitizes immense information, and it is the reason why data transmission is being regarded as one of the most critical activities. However, some users steal of the prevalent information and use it for illegal activities to achieve personal gains. The particular activity has a significant impact on communication privacy.

For instance, the popular DDoS is a type of malicious attack that has a direct effect on the normal functioning of the network The respective threat floods the network traffic, and the victims take a very long time with an aim to recover from the respective attack. However, this issue can be mitigated by checking the packets transmitted every time before sending the same to the source. Further, deep learning is being used to perform the verification of the packets from the source node. Deep learning is the machine learning method, and it possesses the ability to handle the complex data in addition to ensuring support to the network scalability.

With this as the background, it is evident that, the continuous series of cyber-attacks and malware across aggregate networks is a critical threatening issue which needs to be mitigated and controlled, either through reactive solutions or prevention solutions. The security measurements and scales have to be implemented to protect the network and to track the real criminals. However, to bring the criminal to justice, abundant evidence collection and preparation to criminalize the offenders is essential.

Rest of the paper is organized as follows, Section I contains the introduction of DDoS attack and Forensic based SDN, Section II contain the related work of forensics based SDN in data center. In this section, different literature will be reviewed. Section III contain the Methodology; Section IV contain communication protocol that is proposed. Section V possesses information about proposed forensic system flowchart. Section VI describes simulation environment. Section VII will have key metrics and recommendation. Section VIII simulation analysis, Section IX possess result and analysis along with key findings. Section X will be conclusion, Section XI will be future work and Section XII will be recommendation.

2. Related Work

In [14] authors have proposed to introduce a new fingerprinting attack in SDN. The network setup consists of Openflow switch, a controller and two host devices for network communication. Each client in the network observe different response time as the flow setup time adds as there is no flow rule for handling data packets. Author formalized the response time at client side. There exists a SDN SCANNER, which contains the time period in which each host devices communicate with the head field. When the attacker runs SDN SCANNER he will find the time slots of the target and simply send packets to consume resource information from the target in the network. They performed experimental analysis with bandwidth and time to obtain the resources of data and control plane. In [16] authors have proposed a deep neural network model for intrusion detection in SDN. It is widely preferred for its flexibility to extract the network information. They have confirmed that using deep learning approach has strong potential in case of fault identification. The deep learning module help obtain and implement basic information like network using deep neural network. They performed experimental analysis and obtained observations which proved that this approach outperformed than Naïve Bayes (NB), Random Forest, Random Tree, SVM and Multi-layer Perceptron in terms of accuracy. In [16] authors have proposed the effects of two attacks that are specific to software defined networking as DoS (Denial of Service) and DDoS (Distributed Denial of Service) and analyzed their impact in the network. They have put forth the recovery methodologies for the aforementioned attacks. In SDN, the data plane (which transfers the packet) and control plane (which controls the movement of data packets) approaches separate and centralize control to handle traffic and pack flow efficiently. They performed experimental evaluation on the basis of timeout value and bandwidth of control plane with their impact in switch's capability on a Mininet framework and provided their analysis results about these two attacks. In [11] authors have proposed a multi-vector based Distributed Denial of Service attack detection

system. Software Defined network is highly flexible with different objectives and reduces the need for a third party hardware specific vendor. They implied deep learning approach for detection of Distributed Denial of Service attack and placed the system on top of SDN controller to improve efficiency. They performed experimental analysis on different metrics in different traffic traces collected from diverse environments and acquired high accuracy for attack detection. In [12] authors have proposed an enhanced solution for mitigating Distributed Denial of Service attack in SDN-based Cloud Environment using a hybrid machine learning model with support vector machine and self-organizing map algorithms. They have also proposed an enriched history-based IP filtering scheme to improve attack detection rate. And they developed a novel mechanism in order to combine the two approaches to protect SDN-based Cloud from DDoS attack. The experimental results showed that this approach outperformed the existing mechanisms as SVMs-SOM, SVMs and SOMs for classification and detection of DDoS attack in terms of attack arrival rate, weights of boundary, observation time, attacks time, number of SVM classifiers and number of SOM neurons. In [22] authors have proposed a cross domain attack detection to improve the existing detection mechanisms. They tried to overcome the real-time abnormal traffic issues in multiple network domains. There are chances for leakage of information, as each SDN needs to provide traffic data. Existing privacy protection schemes achieve confidentiality at the cost of accuracy and time. Achieving accuracy at minimal time period is a challenging task, which the proposed system addresses. Their approach, combination of perturbation and data encryption to protect privacy and an improved k-NN (k-Nearest Neighbors) algorithm the detection. k- NN algorithm is not an iterative approach and suffers from linear analysis attack and is not very clear about the attribute and type of distance to use. Experimental evaluations showed that this approach obtains accurate attack detection with privacy of sensitive information. In [3] authors have proposed Game Theoretic based approach which is an autonomous system to detect, identify and mitigate the impact caused by DoS and DDoS. They also performed Fuzzy-GADS (Genetic Algorithm Detection System) instead of GT-HWDS (Game Theoretical system using Holt-Winters) to compare the performances under each. They have compared the two approaches in terms of dropped malicious flows and dropped legitimate flows as drop percentage. They have concluded GT-HWDS is efficient for SDN against malicious threats and to avoid congestion near the controller by appropriate mitigate actions. GT based decision making guarantee proper functioning of SDN, even though it impacts depend on the attack intensity. In [23] authors have designed and implemented a Software defined Intrusion Detection System to mitigate the adverse effects of Distributed Denial of Service (DDoS) attack and provide security to the network. S-IDS monitors& detects the attack in the network and then notifies the SDN controller to protect the network at early stage. Their approach based on SDN at the client side, mitigates the DDoS attack while maintaining the normal network operation flow without pause. The use of intrusion detection system for monitoring the network has greatly reduced the workload of SDN controller which takes too much time to discover and mitigate a DDoS attack. The experimental results showed that their approach detects several types of cyber-attacks based on DDoS and reduces their impact by ensuring the data delivery efficiency. In [10] authors have proposed the DDoS attacks that are persisting in the SDN environment. They have proposed two different views of issues as (i) SDN network prevents its members from DDoS attack (ii) SDN network itself becomes the victim of DDoS attack. They have proposed existing techniques that suits for mitigation of first issue as machine learning based approach, statistical based approach and application specific approach. For the second case of issue they have specified protecting SDN at various levels of network as protection for data plane and control plane separately with integrated defense by collaborative intelligence in switches. The experimental results show that the SDN is not completely secure and needs more security. In [15] authors have proposed a detection method of DDoS attack Advanced Support Vector Machine (ASVM) technique

which is a multiclass classification method used with linear kernel and three different classes in this paper. They detect two flooding-based DDoS attacks (UDP flooding attack and SYN flooding attack) with three controllers as if any one controller fails the other two controller can be use. The SDN traffic creates the data set from Openflow Switches. The experimental results perform considering classification error, gamma value and decision function shape which measures in terms of a false alarm rate, detection rate and accuracy using Mininet and cross validation method for training and testing and showed great improvement in the all the performance metrics. In [15] authors have proposed an AIS (accounting information system) process based SDN to detect and mitigate DDoS attacks from the network. They proposed a collaborative traffic monitoring with OpenFlow v1.3 based network application. The progress in detecting DDoS attacks assuming SOM module does a detection signature. They also analyzed the toxicity level of DDoS attack in the network considered. The experimental results performed by considering network traffic, cumulative packets detected, packet tox and byte tox in NS-3 network simulator. This shows that the proposed approach is way better than WILDCARD profile which is another existing scheme. In [15] the roles of handoff process of the wireless node in wireless network with the access points having service guarantee is elaborated. The wireless node usually initiate the handoff process and handoff occurs smoothly within the fraction of second. But in case of the application such as multimedia, real time video accessing, and voice over the Internet protocol service by the wireless node the hand-off process becomes inadequate and after few seconds the loss of the packets are caused which further results the loss of the information for the wireless node in the wireless local area network environment. But, with the SDN based WLAN the controller of SDN determines when to execute hand-off process and selection of the access point to be connected after handing off the previous access point and enhances the security features associated with handoff process from hacking the information during handoff delay. The SDN based WLAN handoff is implemented under a testbed scenario and the efficient result of decreased delay in the process of handoff of wireless node is produced. Therefore, stability of the handoff process is ensured by SDN. In [8] under the SWAN (Software Wireless Network) a SDN having the architecture of Software Access Point (SAP) is taken as key technology. The migration of wireless node from current access point to another access point is taken by wireless node without the breakup of the connection. This is due to fact that the access point is identified by BSSID and SWAN. The logical isolation is provided by SAP so that consistency of SAP is maintained instead of physical access point from which the wireless node connects. This technique of SAP and SWAN can be used to develop the effective and efficient handover mechanism for the WLAN. In their work the authors, proposed secured environment for wireless nodes to receive all the agents which are listenable and a threshold is used to detect the mobility. The value once obtained is greater than that of the defined threshold value then it is considered that the corresponding access point is not going to be connected with the wireless node so that intruder cannot be able to connect during the handover process as the threshold is automated and non-predictable due to SDN. This states that wireless node moved and the information is sent to the access point once the value is lower or equal to the given and defined threshold. The handover handler is only evoked in the case of meeting the criteria of the thresholding. SAP effectivity does this so that SDN is more preferable option for the handover mechanism for WLAN.

3. METHODOLOGY

3.1 Introduction

This chapter represents the methodology of the research. An overview of the theoretical approach is presented, followed by a methodology flow chart and flow chart of the overall system. In addition, the flow chart of the

proposed forensic system is provided, with a description of the simulation environment. An overview of the communication protocol and deep learning approach is discussed. The scenarios of packet processing with forensics-based SDN server is also illustrated in this chapter.

3.2 Overview

SDN has grown tremendously with the increasing technological advancements and demand in networking. With the help of a centralized control, behavior of all the components in the network can be handled efficiently. Today's world is digitized with increased waves of information, which makes transmission of data a crucial part in a network. There are certain users who steal this information for illegal purposes and may modify the data and transmit to their own destination. Such practice disturbs the agreed communication privacy.

DDoS attack is a malicious threat which disrupts the normal working of network by causing a flood of network traffic. When this attack affects the network, it takes a long time to realize and recover from the attack. In order to overcome the discomfort, the current study proposed to check the packet every time before being transmitted to the source. The packets are verified from the source node using deep learning approaches to identify the presence of DDoS attack. Deep learning is a method of machine learning and it is adopted for its ability to handle complex data and support scalability of network.

The main advantages of SDN lists as follows:

1. Centralized networking approach
2. Guaranteed packet delivery efficiency
3. High network security

Traffic control cloud data centers handle different data centers. They can process a large volume of data. Forensic in SDN is the technique to investigate unlawful activities in the network caused by attackers (illegal users) who wish to steal the information or pass unwanted information (virus, malware, worm) to the target victim which can have serious impacts.

Many researchers have proposed various algorithms and approaches to protect the network from threats such as DoS, DDoS, Port Scan, Flash Crowd etc. Some researchers identified the possibility of new attacks in SDN [14]. These attacks and the diverse approaches that attackers can use shows the impact they cause in the network. Authors have been implementing optimization algorithm (genetic algorithm, fuzzy), machine learning algorithm (deep neural network, deep learning) and others to accurately pinpoint the attack and its plausible impacts in the network. Researchers such as [3] concluded that, machine learning algorithms provide effective results than optimization algorithms for identifying the attacks accurately. It shows that deep learning approaches are most suitable to mitigate the threats of attacks.

In this project, an efficient method for routing only the safe data among the nodes in the network has been proposed. Several components, approaches and algorithms to obtain accurate results than the other existing network have been used. The objective is to identify the attacker who is causing threat in the network and revoke their access. The nature of data packets from the mobile node every time to prevent malicious data being transmitted in the network which causes a serious threat to the network is subjected to verification. The

transmission of normal data packets from unaffected nodes in an efficient way through the implication of OpenFlow switch is allowed.

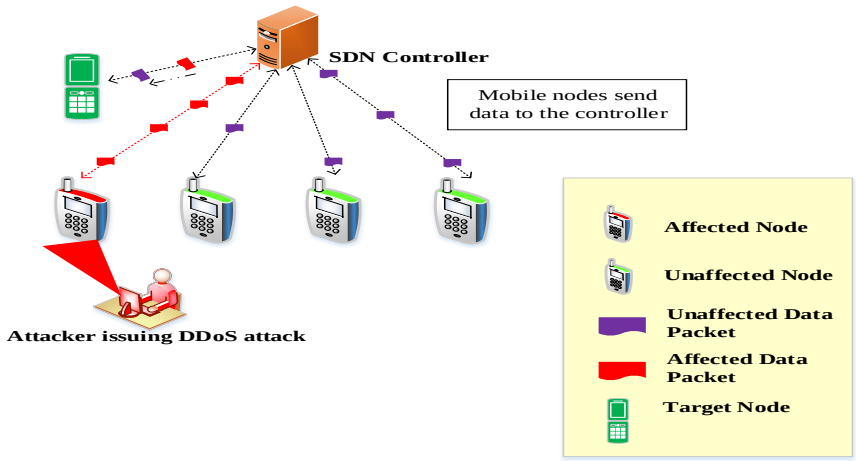

Figure 1: A simple DDoS attack in SDN

Figure 6 represents the DDoS attack performed by an attacker in the SDN environment. The attacker will release malicious packets into the network that will affect the controller and the target, disrupting the normal network performance.

3.3 Methodology Flow Chart and Overall System Description

In this project, a novel approach has been presented to identify the illegal mobile (user) from the network and block their transaction completely as shown in figure 2. The project involves three phases:

- Mobile IP verification
- Threat verification and corresponding action
- Transmission of packets

Initially a network is set up with 20 mobile nodes, OpenFlow switch, 1 head controller, 1 detection engine, 1 domain controller, 1 source controller and cloud server. In the first phase, a consideration is given to mobile nodes, which are trying to send the packets to the destination. It first sends the data to the head controller where its IP checks with the blacklist maintained by the source controller. If present then data packet from that particular mobile node drops. Otherwise, the data packet forwards to detection engine for further proceeding.

In the second phase, detection engine performs the role of detecting security attacks and reporting to the head controller for attack mitigation. Detection engine is the part of forensics that identifies the DDoS attack efficiently. The head controller communicates with the source controller to take necessary actions. If the source controller does not take any action, then the head controller communicates with the domain controller to perform the corresponding actions. The domain controller is in the detection engine and monitors the data packet arrival port. It implements the deep learning approach to verify the packet and report the status to head controller. On the basis of the status from the detection engine, head controller decides the actions as whether to transmit the data packet from the mobile or drop the particular data packet.

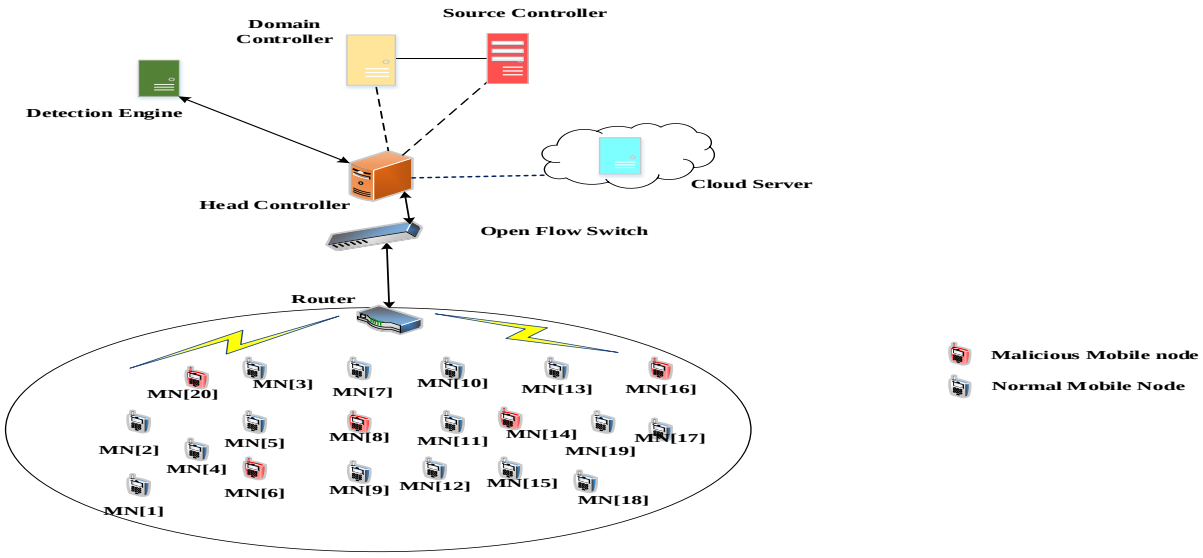

Figure 2: Simulation setup

Figure 7 represents the overall network setup of the simulation environment. In the current study 20 nodes, 1 head controller, 1 detection engine, 1 domain controller, 1 source controller and cloud server have been deployed. There are some malicious nodes which efficiently identify and block data packets in the network.

DDoS is a malicious attack that takes place mostly in SDN, and it very hard to realize the attack. Nowadays, illegal users attack the entire network in order to bring down one target which is a serious dispute. A deep learning approach has been proposed to solve this challenge by detecting the attack during every packet transmission. Deep learning is a part of machine learning which has proved to be the best approach for verifications of data packets in the network. The domain controller then communicates with the head controller and transmits the result back to it. Based on the beacon time out and Duplicate Address Detection (DAD), the data packets are detected whether they arriving from the attackers or not. The checking takes place again if the data gets affected by the attacker, to identify whether it is in a serious or normal stage. If the attack is not in a serious stage, then all packets from the source node for that particular session will be blocked. If the attack proves to be in a serious stage, then the traffic as evidence to follow-up the trace of the criminal can be recorded, and the IP source of that packet to the blacklist can be included and blocked from this source and eliminated.

In the third phase, head controller transmits the data to the destination through an OpenFlow switch, ‘if and only if’ the packet is sent from a mobile node without any record of any attack and blacklist encounters. OpenFlow switch is a hardware device that forwards data packets in SDN network and can record with computing resources in an easy manner. OpenFlow switch supports SDN for the purpose of routing bulk traffic in the network. Using this OpenFlow switch, unaffected data from valid users are allowed in the network. It also has the potential to arrest the activities of illegal users from the network completely.

The proposed approach is faster than the routing processes as it also maintains the attack record in the source controller as evidence in case of a forensic examination. With the proposed approach the malicious user can be identified by following the traffic in the network, which is of great use in forensic department. The number of

nodes in the network can be increased as the proposed approach supports network scalability. This approach is highly reliable for distinguishing the malicious nodes effectively and accurately. The experimental results show the originality of the approach and its significant actions in categorizing any threat.

3.4 Flow Chart of the Proposed Forensic System

The following figure 8 illustrates the flowchart of the proposed forensic system, the detection engine is in the ideal state until it receives a new packet. The verified packets is allowed only if it has an authentic IP address. For unauthorized packet, the detection engine algorithms will be run; first, the algorithm for DDoS detection will be run, if the algorithm detects DDoS, then it will ask head controller for an action the head controller will coordinate with the source controller and domain controller, as explained in the section 3.2 to take a the appropriate action. If the algorithm does not detect any of DDoS, then the detection engine will run the network intrusion algorithm [3]. If the test does not identify any network intrusion in the forensic system, it then allows the packet and adds the IP source to whitelist.

If forensic system detects DDoS or any network intrusion attack, then head controller with source and domain controller coordinates to take actions. The first action is determining if the attack is serious or not, and neglect some of the les serious attacks. If the attack is not serious, then all packets from the source for this session are blocked. However, if the attack is serious, then the traffic is recorded as a criminal evidence to trace the criminals and bring them to justice. The IP source of the packet is added to blacklist, all packets from this source are blocked and eliminates.

ALGORITHM FOR DETECTION ENGINE AND DOMAIN CONTROLLER

Input: Packets from mobile node, Blacklist

Step 1: Begin

Step 2: Mobile node give packets to head controller and head controller verifies with source controller

Step 3: Head controller transfer packet to detection engine

for each packet do

if (packet == affected by DDoS)

// The domain controller monitoring the packet arrival ports to verify the packets, Next, based on the verification, detect the DDOS attack node based on the deep learning approach, and add into block list and stored in the source controller.

{Inform head controller}

else

{Inform head controller}

// In this else statement, have information without attacker.

Step 4: head controller now send packet to domain controller

if (attack ≠ serious)

// For measure the serious, consider the blocklist.txt file only, like the IP occurred in the file[list] or not, because a simulation based process only.

{add the mobile IP to the blacklist}

// Maintain the blacklist only, which IP does not occur in the blacklist, it is consider as a normal node.

else

{add to the blacklist and trace to find the identity of the attacker}

Step 5: send the unaffected packet for transmission to destination

Step 6: loop Step 3 till 5 until all the mobile nodes answers 4

step 7: End

Output: All attack packets are detected

3.5 Description of the Simulation Environment

The proposed solution evaluates in OMNeT++ 4.6, an event simulator for modeling communication networks, to produce accurate results. OMNeT++ suites for SDN as it eliminates the complexity, and supports to run network model in cloud. OMNeT++ is a C++-based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems. OMNeT++ attempts to fill the gap between open-source, research-oriented simulation software such as NS-2 and expensive commercial alternatives like OPNET. The simulation software facilitates visualizing and debugging of simulation models in order to reduce debugging time, which traditionally takes up a large percentage of simulation projects. This simulation evaluation performance shows the accuracy of the proposed project.

3.6 Communication Protocol

In standard SDN, the communications between the controller and the switches is with theOpenFlow protocol which is not secure and sensitive for DoS and DDoS attacks [6]. This research proposed a new communication protocol which is secure. The proposed protocol with a packet header contains the necessary information or data to allow the forensic network system to accomplish its tasks, each controller has its ID, and each method within the controller has its signature and scope to reject unauthorized access. The header contains Time-To-Live (TTL) field and starts with the highest value of ‘live’ which is determined by the administrator of domain controller level, and for each hop (switch, controller, etc..) that receives this packet, the TTL decreases by 1. Thus, when

TTL reaches 0 at a specific hop, this hop will eliminate the packet. Domain controller level for a period time according the enterprise policy will re-generate the tokenizer that uses the proposed protocol to encrypt the content of communication packets.

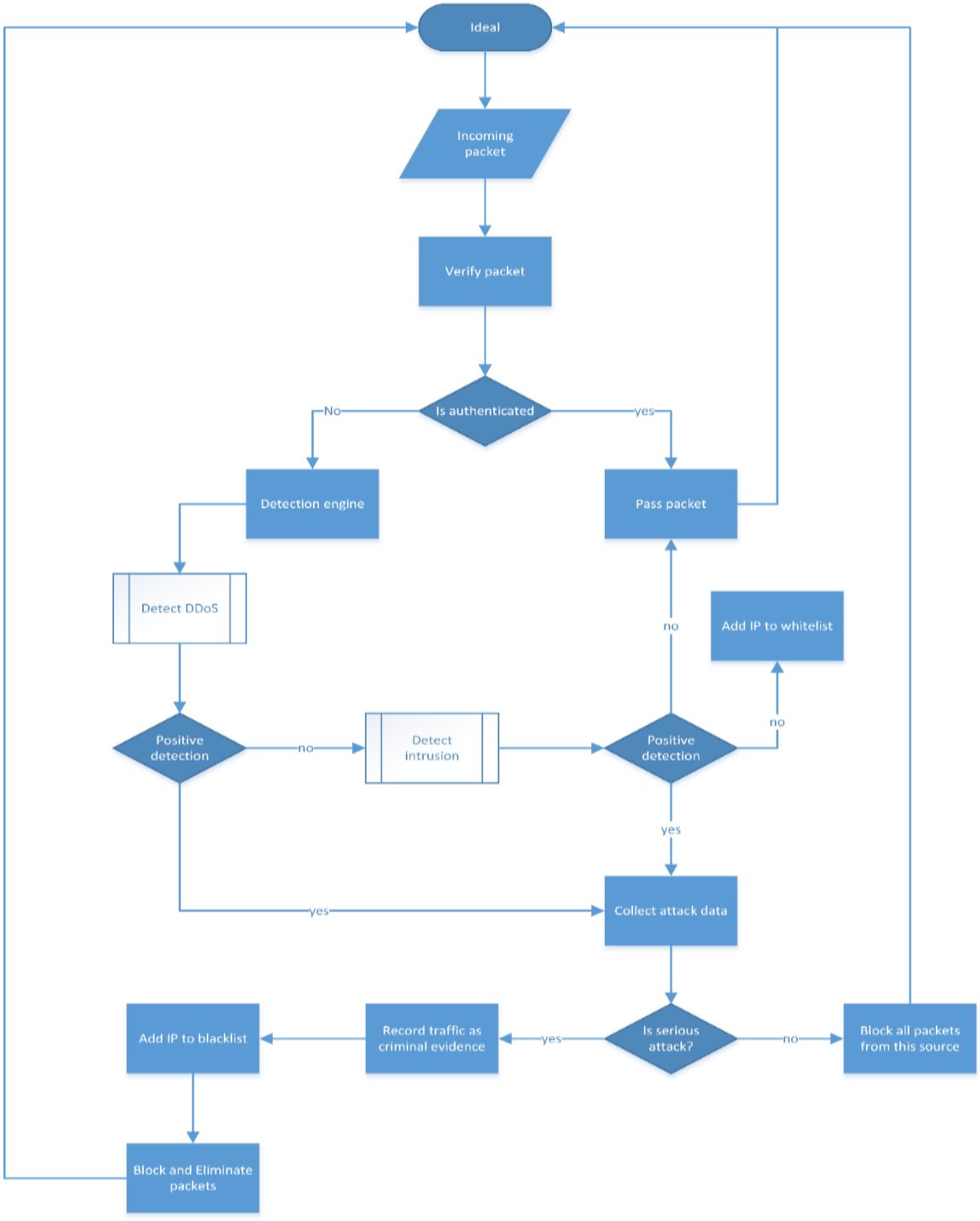

Figure 3: Proposed forensic system flowchart

Each controller or switch that wants to communicate with others need this key; moreover, every method signature will encrypt with this key. The proposed communication protocol packet caries the IDs for the destination and the source, each ID stores at routing table for each controller and secures it to keep tracking of paths. The protocol helps the forensic network system to reject unauthorized access to the data centers.

3.7 Deep Learning Approach

In classical machine learning, important features of a design are inputted manually, and the system automatically learns to map the features to an output. In deep learning, there are multiple levels of features. These features are automatically discovered and composed together in various levels to produce outputs. Each level represents abstract features that are embraced from the features presented in the previous level on a continuous basis [11].

3.8 Scenarios – Packet Processing with Forensics Based SDN to the Cloud Server

3.8.1 Scenario 1

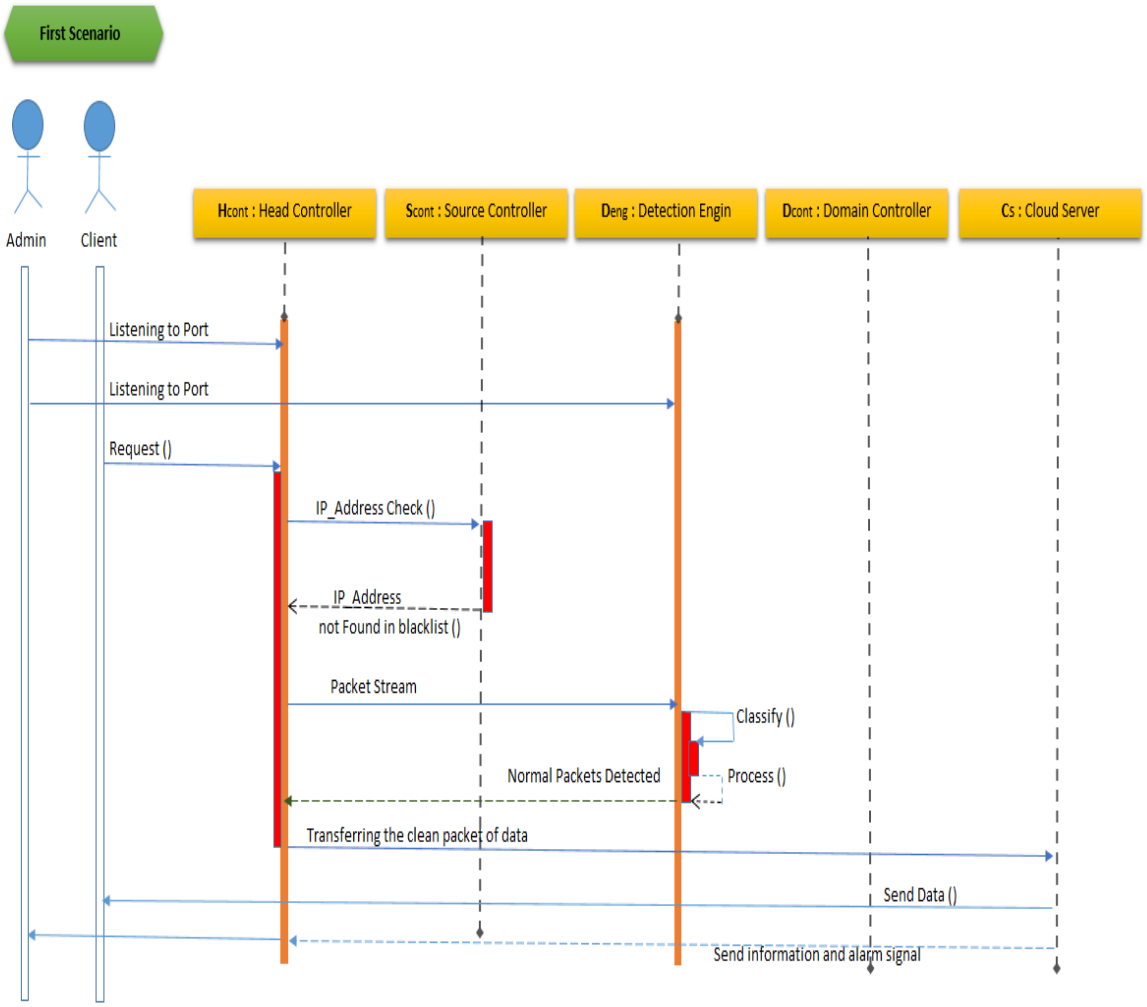

Figure 4: Normal nodes transfer packets to the cloud server

3.8.2 Scenario 2

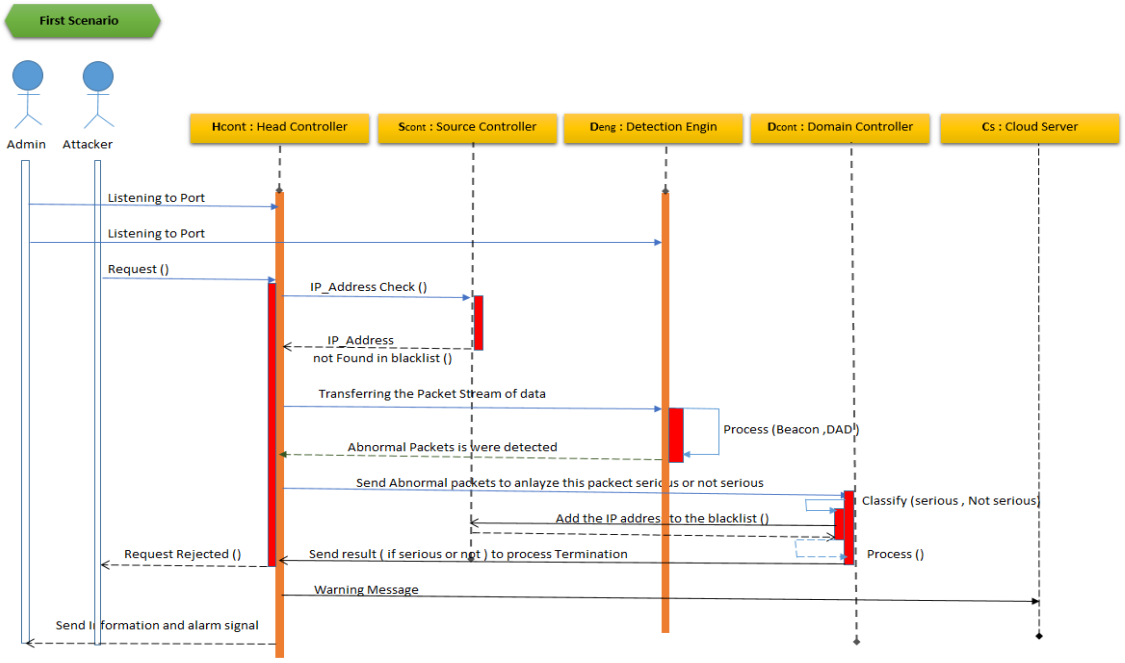

Figure 5: Malicious nodes/attackers attempt to transfer packets to the cloud server abnormal packets are stored into the blacklist)

3.8.3 Scenario 3

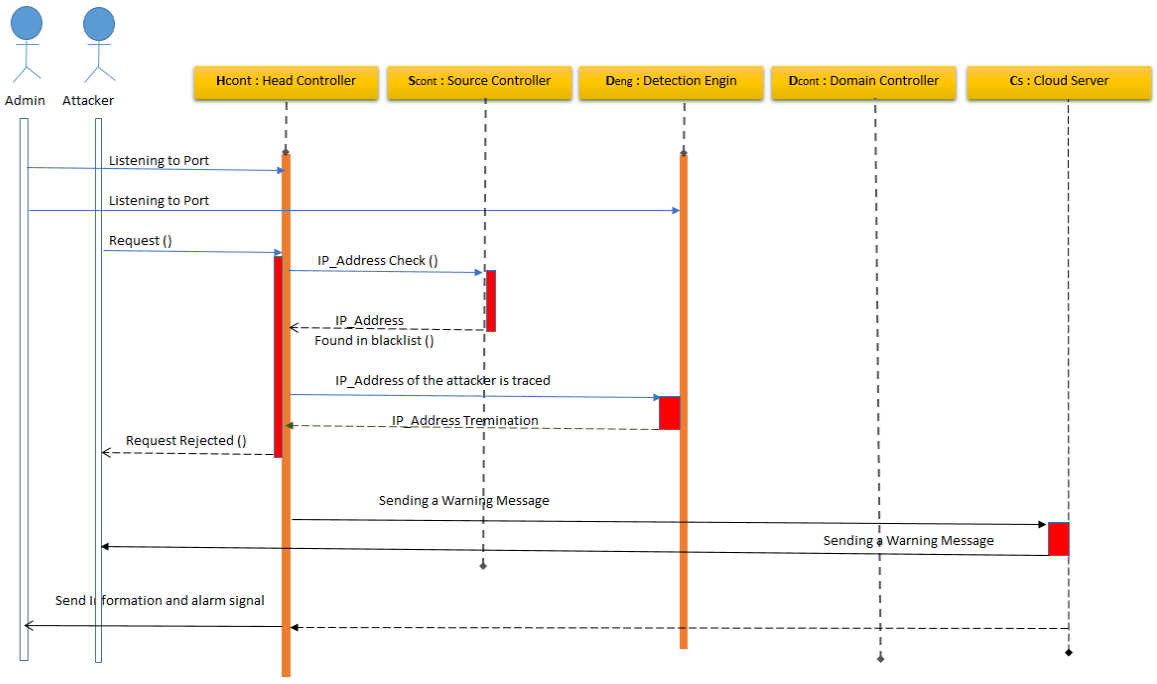

Figure 6: Malicious nodes/attackers attempt to transfer packets to the cloud server (abnormal packets are already in the blacklist)

3.9 A Description of the Packet Features

Table 1: Packet Features

Packet Feature Type	Packet Features Name	DDoS affected node	Normal node
Packet Header Length	Packet header data size	Huge amount of data	Normal data
Protocol used	Type of protocol used e.g. TCP or UDP	TCP	TCP
Packet count	Amount of bytes in each flow	high	normal
Duration (sec)	Switch alive time	high	normal
RTR	Reply to response	---	Late reply
Flags	Used to inform about connection status	periodical	continuous

The http is a service type packet feature. The packet header length has a large amount of data on DDoS affected node and Normal data on the normal node [20]. The packet count is high for a DDoS affected node and normal for normal node [9]. The switch alive time is high for a DDoS affected node and normal for normal node [17]. A late reply to response obtained for normal node. The periodical flags **use** DDoS affected node and the normal node uses continuous flags to inform about the connection status [7].

- Service Type

The node will send the packet to the head controller irrespective of the status as affected or not affected. The affected node will request to send data [beacon] packets and the normal node will request to send the data [beacon] packets to the head controller. Both of them use the same http protocol to send request to the head controller.

- Packet Header Length

The normal will send only the necessary data packet that has to be transmitted through the head controller. But, the affected node will send packets and flood the network so that the controller cannot respond to other normal nodes in the network. The length of the packet header is determined by its size.

- Protocol used

The node will send the packet to the head controller irrespective of the status as affected or not affected. The affected node will send the packets and the normal node will send the data packets to the head controller. Both of them use the same TCP protocol to transfer the packets.

- Packet count

The normal nodes will only the necessary number of packets, but the affected node will send lots of packets to make the network get congested. The packet count is determined by the bytes sent in each flow (from mobile node to head controller)

- Duration (sec)

In a network, each node will have a particular time slot to access the switch. If it is a normal node, the switch alive time will be normal, because it will get immediate response from the head controller. If it is an affected node, then taken to utilize the switch increases, and hence high for DDoS affected nodes.

- RTR

The network is good only when the nodes get immediate response. If the DDoS affected node has corrupted the network then the normal nodes will get late reply from the destination.

- Flags

In general each flag has three bits as synchronization (SYN), acknowledgement (ACK) and finish (FIN). If is a normal node in the network, then it will have continuous connection with the head controller and if it is a DDoS affected node then the connection will be irregular or periodical (improper connection).

4. RESULTS AND DISCUSSION

4.1 Introduction

In this chapter the results of the study are presented and discussed. The evaluation of the proposed model with state-of-the-art is conducted from three perspectives. First, the provision of an easy-to-implement forensic model will be discussed. Second, a discussion will be conducted regarding the applicability of the SDN model for forensics purposes to overcome the issues described in the introduction chapter. Third, the results of an investigation pertinent to the SDN based forensics systems capturing shreds of evidence capability against attackers in data centers environment will be presented.

Requirement of Experiment (Simulation/Emulation)

The basic requirements of the proposed project can be categorized into hardware requirements and software requirments

- Hardware Requirements
  1) Operating System: Windows 7 Ultimate [32 bits]
  2) Processor: Intel® Pentium® CPU G2030 @ 3.00GHz
  3) RAM: 2 GB
- Software Requirements

  OMNeT++ 4.6

4.2 Experimental Set Up and Procedure

In this study, topology of a SDN, consisted of 20 - mobile Nodes [users], OpenFlow switch ,1 Detection engine , 1 Domain controller , 1 Source controller, 1 Head controller and cloud server. The simulation parameters which were implemented in the network is presented in Table 2.

Table 2: Presentation of the simulation parameters considered in the study for performing the simulation.

Simulation Area	1000m x 1000m
Number of Controllers	3
Number of Detection Engine	1
Number of Nodes	20
Bandwidth	50-100 Mbps
Traffic Type	TCP
Transmission Rate	2Mbps
Moving Pattern	Rectangle Mobility
SDN Controller	POX
Packet Size	56 Bytes
Packet Interval	0.1sec
Simulation Time	10sec

The experimental steps performed in the present study are described below:

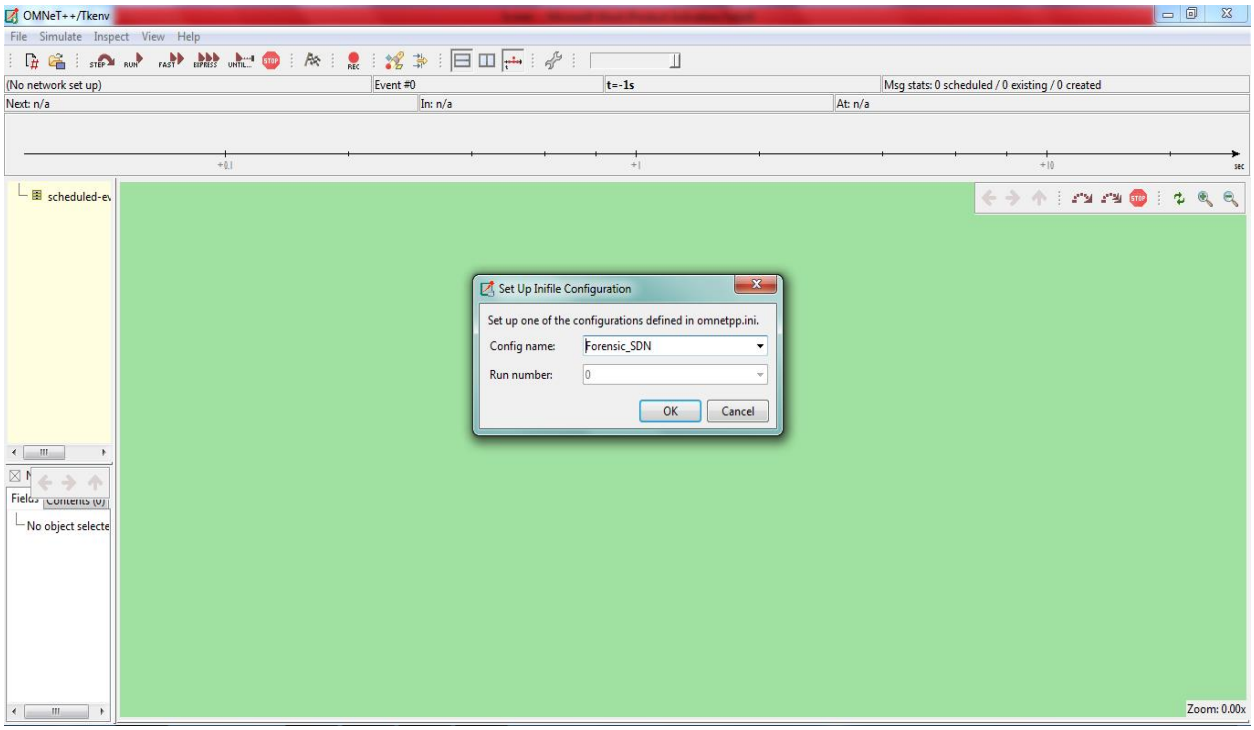

Figure 7: Simulation environment

Figure 15 represents the initial configurations in the simulation environment. There is an initial configuration setup of the network followed by the movement of mobile nodes simulated environment.

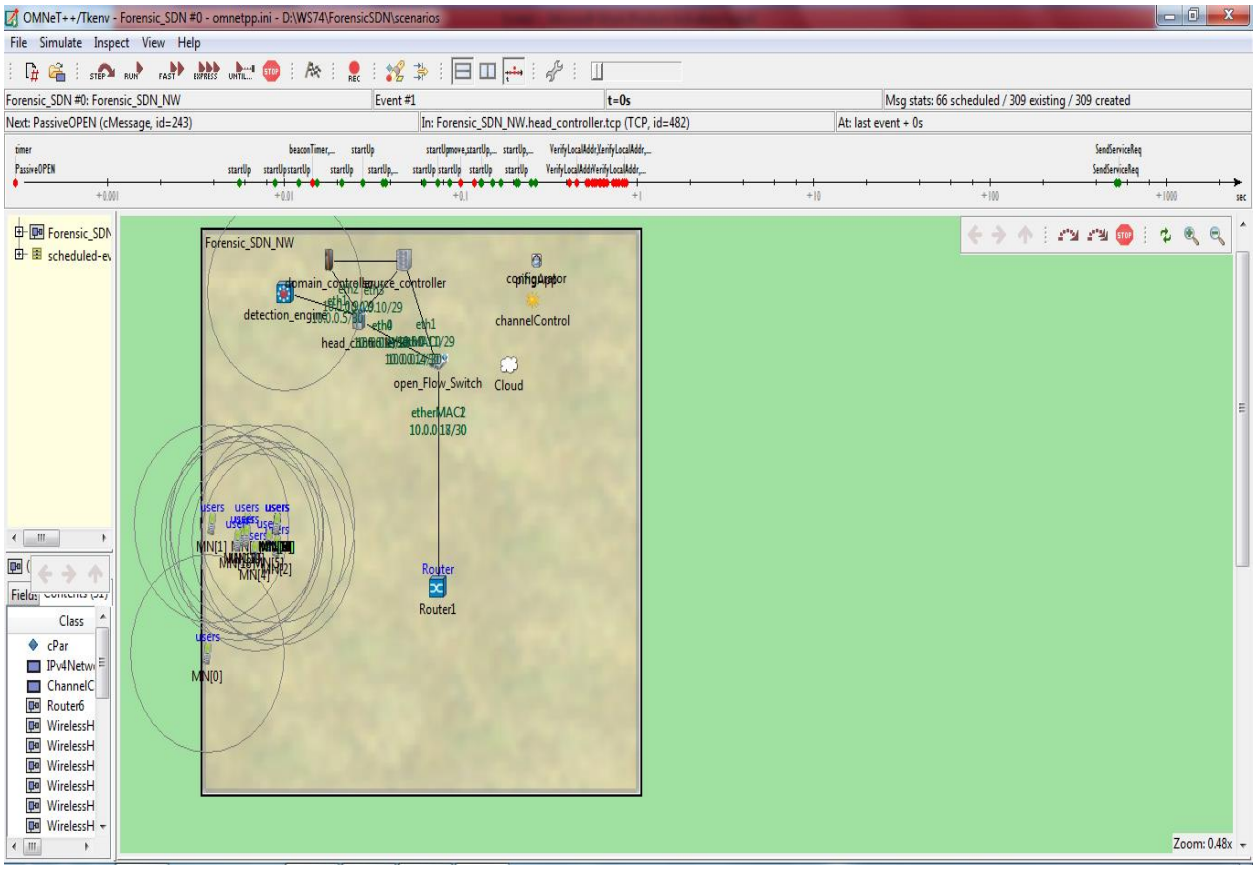

Figure 8: Overview of simulation environment

Figure 16 presents the created simulation environment that consists of 20 mobile nodes (users), an OpenFlow switch, a detection engine, a domain controller, a source controller, a head controller and a cloud server.

4.4 Discussion of Key Metrics

The metrics considered are:

1) Execution Time: Execution time is the total time required by the process to complete the communication across the mobile nodes. Each node operates with different time and the total time required for all the nodes in the network to complete the performance is the execution time of the network. There is only a slight improvement in the execution time with increase in number of nodes.

$$\text{Execution Time} = \sum_{n=0}^{N} t \text{ (per\_packet\_transmission)} \quad (1)$$

Where N denotes the number of nodes in the network, t denotes the time required by the nodes to complete the transmission of data. Thus, the total time consumed for execution by the network for the given N number of nodes can be obtained

2) Throughput: Throughput in the network is the amount of information being transmitted in the packets. If throughput of a network is large it means more data is being transmitted.

$$\text{Throughput} = \frac{\sum_{n=0}^{N} P*size}{s} \quad (2)$$

Where *N* denotes the total number of nodes in the network, *P* denotes the number of packets transferred in the network, size denotes the average size of the packet, *s* denotes the number of seconds. Thus, throughput consumed by the network for the given *N* number of nodes can be obtained

Network throughput is the rate (in bits per sec (bps) or packets per second (pps)) at which packets or bits are delivered over a network channel. Thus, packets received by all nodes can be summed up to calculate the value for a small network or network segment.

3) Packet Loss: Packet loss is the number of data packets being dropped in the network. In the current experiment, 20 nodes were implemented and the amount of packet loss was negligible. If a situation of adding 100 nodes in addition to the existing nodes in the network is hypothesized, then the packet loss will increase with respect to several factors such as bandwidth constraints in the network.

$$\text{Packet loss} = \frac{\sum_{n=0}^{N}(\#\ of\ packets)}{s} \quad (3)$$

Where *N* denotes the total number of nodes in the network, *P* denotes the number of packets dropped by the network due to congestion and *s* denotes the number of seconds. Thus, the total number of packets dropped by the network for the given *N* number of nodes can be obtained

4) End-To-End Delay: End-to-End delay is the time taken for the transmission of data packet across the network. Since the nodes are dynamic, the distance from source to destination alters frequently which increases the time taken for data packet transmission.

$$\text{End to end delay} = \frac{\sum_{n=0}^{N} R}{s} \quad (4)$$

Where $R = (t_S - t_R)$ denotes the difference in time required to send, (t_S) is the request and time required to receive the reply (t_R), *N* denotes the total number of nodes in the network, *s* denotes the number of seconds. Thus, the total end-to-end delay by the network for the given *N* number of nodes can be obtained

5) Energy Consumption: Energy consumption is the amount of energy required for performing communication between the source and the destination measures in terms of energy consumption

$$\text{Energy Consumption} = \frac{\sum_{n=0}^{N}(IE-RE)per\_packet\_transmission}{s} \quad (5)$$

Where *N* denotes the total number of nodes in the network, *IE* denotes the initial energy and *RE* denotes the residual energy, *s* denotes the number of seconds. Thus, the total energy consumed by the network for the given *N* number of nodes can be obtained

4.5 Results of Each Metric in The Study and Comparison with the SDN standard

1) Execution Time

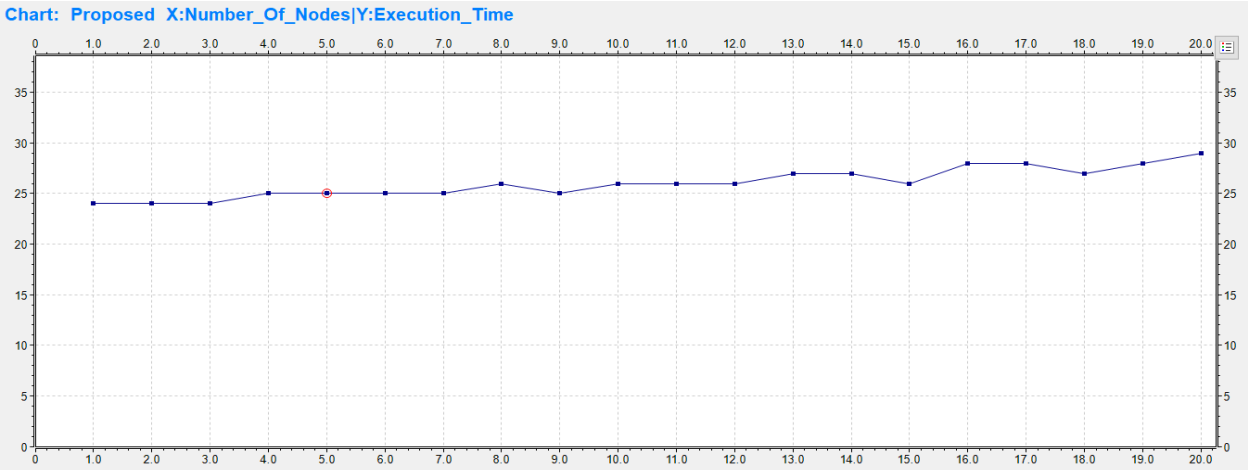

Figure 9: Graph of execution time against number of nodes in the network

Figure 23 represents the graph plotted against execution time and number of nodes in the network. Transmission stage calculates the execution time, with the addition of new nodes into the network. As the nodes increases, the execution time also increases gradually in the network, as the time taken by each node to complete their tasks varies with respect to several factors.

Table 3: Execution time

Number of Nodes	Execution Time (ms)
5	21
10	22
15	24
20	25

2) Throughput

The rate at which the data gets transmitted in the network is the throughput.

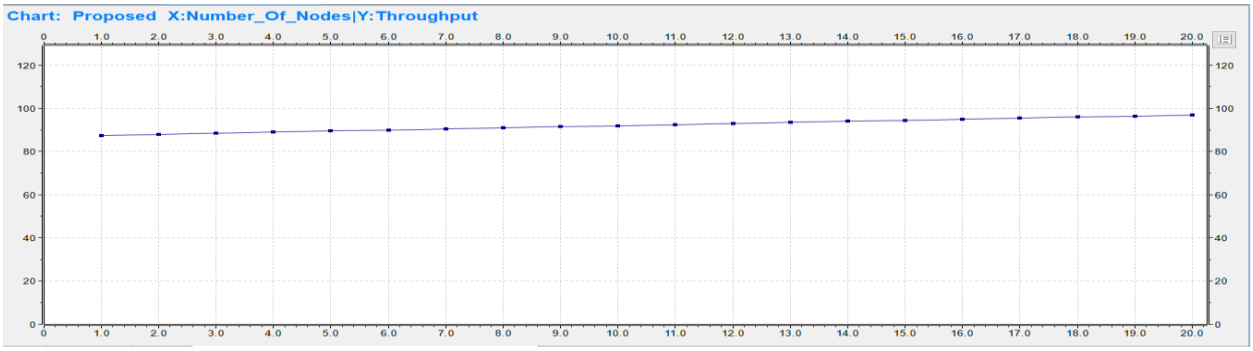

Figure 10: Graph of throughput against number of nodes in the network

Figure 24 represents the graph plotted against throughput and number of nodes in the network. During data transmission in the network, the amount of information transmitted is larger and efficient. A graph for throughput against number of nodes in the network has been plotted and results indicate that the throughput showed great increase.

Table 4: Throughput

Number of Nodes	Throughput (pps)
5	90
10	91
15	93
20	97

3) Packet Loss

A graph plot can identify the packet loss in the network against the number of nodes.

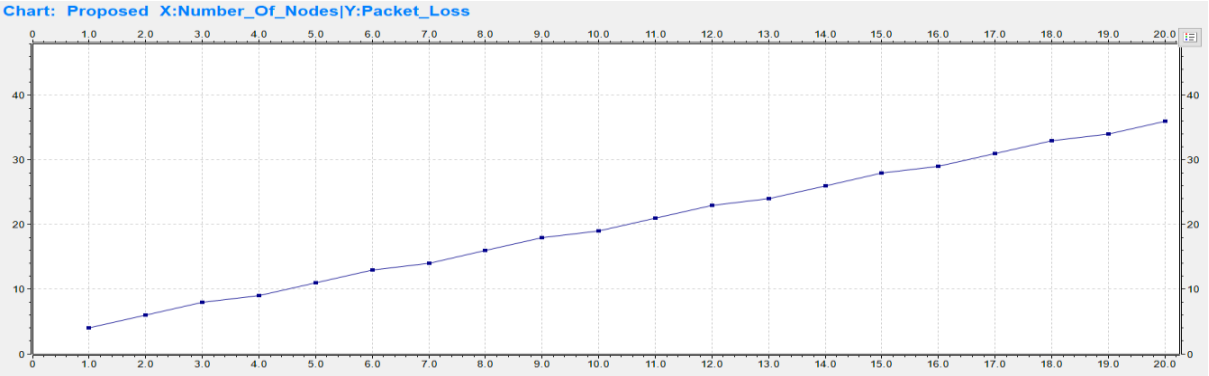

Figure 11: Graph of packet loss against number of nodes in the network

Figure 25 represents the graph plotted against packet loss and the number of nodes in the network. It is visible that as the number of nodes in the network increases packet loss also increases. Elaborating on the same: in the current network there are 20 nodes and each node transmits 10 packets (200 packets). There is a possibility that then there some packets can be dropped in the network due to various reasons like congestion. In the current simulation, only a slight packet loss was evident.

Table 5: Packet loss

Number of Nodes	Packet Loss (data packets)
5	11
10	19
15	26
20	34

4) End-to-End delay

A graph plot can determine the end-to-end delay against the number of nodes.

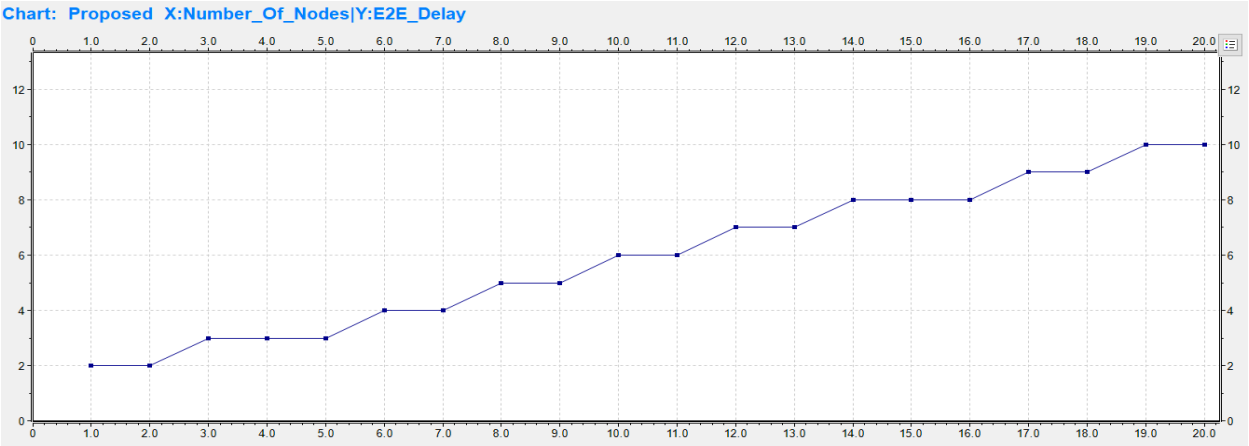

Figure 12: Graph of end-to-end delay against number of nodes in the network

Figure 26 represents the graph plotted against end-to-end delay and the number of nodes in the network. The time lapse between data transmission from source to destination is end-to-end delay.

Table 6: End-to-End delay

Number of Nodes	End to End Delay(seconds)
5	3
10	5
15	8
20	10

5) Energy Consumption

Energy consumption increases with the increase in number of nodes. In order to perform, communication nodes require energy. When the network is extensive, the amount of energy required for communication increases gradually without causing adverse effects.

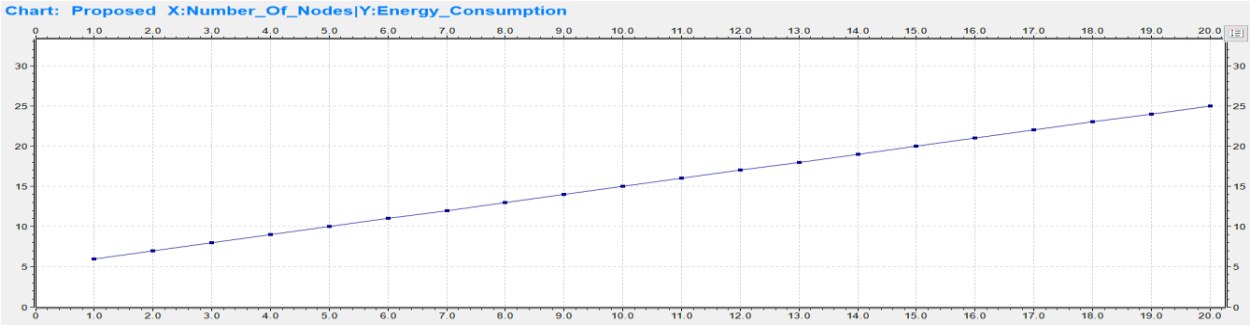

Figure 13: Graph of Energy Consumption against Number of Nodes in the Network

Figure 27 represents the graph plotted against the energy consumed by the network as the node increases. Since dynamic nodes were deployed, each node uses different energy for performing data transmission. Thus, the overall energy consumed by the network increases with the addition of new nodes.

Table 7: Energy Consumption

Number of Nodes	Energy Consumption (mJ)
5	10
10	15
15	20
20	25

5. Conclusion

The main contribution of this research is detecting attacks, following and recognizing the attackers and bring them to justice. Also, the research introduces a new proposed secure protocol that maintains communications between controllers and others, and ensures the security between controllers and switches, and helps in correcting the forwarding packets or eliminating them. This project has demonstrated an efficient deep learning approach in order to detect the DDoS attack in SDN. The required components of the network were positioned in the simulation environment and proposed approach was performed. The mechanism to stop the malicious nodes from further transmissions in the network was also performed. In this approach, the head-controller assumed the responsibility of verifying IP addresses of each incoming mobile node, that aimed to transmit data to the destination. The source controller maintained a blacklist which contained the guilty IP address of mobile nodes. Detection engine, used forensic purposes, checked the packets from head controller to identify any possible threat. The domain controller performed deep learning algorithm analysis with detection engine to accurately identify the nature of the data packets as either attacked or un-attacked. Only the unaffected data packets from the normal mobile node were transmitted, which protected the network from illegal users. A record of attacks in the network were also managed .

5.1 Constraints and Limitations

A simulation environment was used to evaluate the effectiveness of the proposed forensic-based approach. Use of simulation environment is associated with multiple constraints which has impacted the study. Firstly, the simulation can predict the performance of the suggested approach; however, it fails provide accurate details on its performance in real-life. The lack of opportunity to know the effectiveness of the new system in actual data centers is a constraint in this case. Secondly, the difficulties in validating the simulation findings are another constraint in this case. The simulation provides the researchers with flexibility to set the environment. In this research, use of simulation was beneficial to enhance scalability of the network to a limited extent. However, in real scenario, the security of data centers is affected due to various factors such as the behavior of users. Therefore, the lack of real-life dynamics was another constraint in the research. Further, there are several limitations associated with the research. Firstly, the research is done using limited number of nodes. Practically, the data centers have ‘n’ numerous nodes. Thus, the current research does not provide any idea on how the proposed approach will perform in large data centers. The current research does not provide any idea on how the proposed approach will perform

in the presence of other technologies. Finally, the current research does not provide detailed idea on how the attackers will be tracked and categorized. Tracking and attacking the attackers is beneficial to prevent the future attacks. The inability of obtaining complete information on the attacker is another limitation in this research.

5.2 Future Works

The results of the study can be extended to plausible areas, which have been identified below:

- Scalability

Simulation tests will be performed with increased number of nodes in the network to test the scalability of the new technology. The aim is to ensure low energy consumption. We will also try to keep the energy consumption as low as possible.

- Security Analysis

DDoS attack is a malicious threat which disrupts the normal working of network by causing flood of network traffic. When this attack affects the network, it takes a long time to recover for realization and recovery from the attack. A plan exists to extend the detection of attacks on a large scale. In this document, analysis is conducted with DDoS attack as a main issue. In this line, in future tests, the goal will be to detect the unobservable attacks such as data exfiltration and collusion between compromised nodes.

- Tracking of User

Future works will extend the forensic assistance to the track various kinds of attacks under the three specific categories such as legitimate attacks, compromised attacks and malicious attacks.

- Security

Further research studies can help protect the encountered smart power in case of a cyber-attack.

- Interpolability

Further work will develop a system that will be interpolable at all the required levels.

- Side channel pacing

Covert channels are numerous in networked networks and even individual hosts, and it can be difficult to detect and delete them. Further works can focus on side-channel pacing to prevent side-channel attacks

- Automation in forensics

The need for application instrumentation is a common barrier to the implementation of a forensic device, which future studies and simulations can overcome

5.3 Recommendation

SDN controls the traffic and other network activities from a centralized location. SDN which is managed from central location and configured programmatically, separates from forwarding phase and is thus being preferred. The approach of using forensic-based SDN is as an advantage to the cyber security department as it can track the malicious user along with blocking them. SDN is vulnerable to attacks such as DoS, DDoS, Port Scan, Flash Crowd etc. Through the novel approach the users can be traced to their root and the unlawful users can be brought the justice.

6. References

[1] Alshammari, F. (2017). An Efficient Approach for the Security Threats on Data Centers in IOT Environment. *International Journal of Advanced Computer Science and Applications (IJACSA),* 8(4), pp.73-80.

[2] Arista. (2016). *Security for the Cloud Data Center, white paper*. [online]. Available at: https://www.arista.com/assets/data/pdf/Whitepapers/ARISTA_SecuritySolutionWP.pdf

[3] De Assis, M., Hamamoto, A., Abrao, T. and Proenca, M. (2017). A Game Theoretical Based System Using Holt-Winters and Genetic Algorithm With Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks. *IEEE Access*, 5, pp.9485-9496.

[4] Francois, J. and Festor, O. (2014). Anomaly Traceback using Software Defined Networking, *WIFS 2014 IEEE workshop on information Forensics and security, Atlanta, Georgia*, 2014, pp.203-208.

[5] Jacobi, I. (2010). Data Provenance in Distributed Propagator Networks, *IPAW 2010: Provenance and Annotation of Data and Processes*, Troy, New York, 2010, pp 260-264

[6] Kandoi, R. and Antikainen, M. (2015). Denial-of-service attacks in OpenFlow SDN networks. *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM),* Ottawa, Canada, pp. 1322-1326.

[7] Khan, S., Gani, A., Wahab, A., Abdelaziz, A., Ko, K., Khan, M. and Guizani, M. (2016) Software-Defined Network Forensics: Motivation, Potential Locations, Requirements, and Challenges. *IEEE Network*, 30(6), pp. 6-13.

[8] Lei, T., Lu, Z., Wen, X., Zhao, X., & Wang, L. (2014). SWAN: An SDN based campus WLAN framework. *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE*), Aalborg, Denmark, pp. 1-5.

[9] Martins, J. and Campos, M. (2016). A security architecture proposal for detection and response to threats in SDN networks. *2016 IEEE ANDESCON*, Arequipa, Peru, pp. 1-4.

[10] Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T. and Vasupongayya, S. (2019) Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking (SDN). *Journal of Computer Networks and Communications*, 2019, pp. 1-12.

[11] Niyaz, Q., Sun, W. and Javaid, A. (2017) A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN). *ICST Transactions on Security and Safety*, 4(12), p. 153515.

[12] Phan, T. and Park, M. (2019) Efficient Distributed Denial-of-Service Attack Defense in SDN-Based Cloud. *IEEE Access*, 7, pp. 18701-18714.

[13] Pimenta Rodrigues, G., de Oliveira Albuquerque, R., Gomes de Deus, F., de Sousa Jr., R., de Oliveira Júnior, G., García Villalba, L. and Kim, T. (2017). Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection. *Applied Sciences*, 7(10), p.1082.

[14] Seungwon, S. and Gu, G. (2013). Attacking Software-Defined Networks: A First Feasibility Study. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, Hong Kong, China, pp. 165-166.

[15] Swami, R., Dave, M. and Ranga, V. (2019) Software-defined Networking-based DDoS Defense Mechanisms. *ACM Computing Surveys*, 52(2), pp. 1-36.

[16] Tuan, T. A., Mhamdi, L., McLernon, S., Zaidi, S. A. R. and Ghogho, M. (2016). Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. *IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, pp. 258-263.

[17] Wang, Y., Uehara, T., & Sasaki, R. (2015). Fog Computing: Issues and Challenges in Security and Forensics. *2015 IEEE 39th Annual Computer Software and Applications Conference*, Taichung, Taiwan, pp. 53-59.

[18] Wei, R. (2004). On A Network Forensics Model for Information Security. *Information Systems Technology and its Applications. 3rd International Conference ISTA'2004*, Salt Lake City, Utah, USA, 2004, pp. 229-234

[19] Yu, Y. (2012). A Survey of Anomaly Intrusion Detection Techniques. *Journal of Circuits, Systems and Computers JCSC*, 28(1), pp.9-17.

[20] Zha, Z., Wang, A., Guo, Y., Montgomery, D., & Chen, S. (2019). BotSifter: An SDN-based Online Bot Detection Framework in Data Centers. *2019 IEEE Conference on Communications and Network Security (CNS)*, Washington D.C., pp. 142-150.

[21] Zhou, W., Cronin, E. and Loo, B. (2008). Provenance-aware Secure Networks, *IEEE 24th International Conference on Data Engineering Workshop*, Cancun, Mexico, 2008, pp. 188-193.

[22] Zhu, L., Tang, X., Shen, M., Du, X. and Guizani, M. (2018) Privacy-Preserving DDoS Attack Detection Using Cross-Domain Traffic in Software Defined Networks. *IEEE Journal on Selected Areas in Communications*, 36(3), pp. 628-643.

[24] Manso, P., Moura, J. and Serrão, C. (2019). SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks. *Information*, 10(3), p.106.