Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.028077

ARTICLE

Threshold-Based Software-Defined Networking (SDN) Solution for Healthcare Systems against Intrusion Attacks

Laila M. Halman and Mohammed J. F. Alenazi*

Department of Computer Engineering, College of Computer Science, King Saud University, Riyadh, 11451, Saudi Arabia

*Corresponding Author: Mohammed J. F. Alenazi. Email: mjalenazi@ksu.edu.sa

Received: 29 November 2022 Accepted: 09 June 2023 Published: 17 November 2023

ABSTRACT

The healthcare sector holds valuable and sensitive data. The amount of this data and the need to handle, exchange, and protect it, has been increasing at a fast pace. Due to their nature, software-defined networks (SDNs) are widely used in healthcare systems, as they ensure effective resource utilization, safety, great network management, and monitoring. In this sector, due to the value of the data, SDNs face a major challenge posed by a wide range of attacks, such as distributed denial of service (DDoS) and probe attacks. These attacks reduce network performance, causing the degradation of different key performance indicators (KPIs) or, in the worst cases, a network failure which can threaten human lives. This can be significant, especially with the current expansion of portable healthcare that supports mobile and wireless devices for what is called mobile health, or m-health. In this study, we examine the effectiveness of using SDNs for defense against DDoS, as well as their effects on different network KPIs under various scenarios. We propose a threshold-based DDoS classifier (TBDC) technique to classify DDoS attacks in healthcare SDNs, aiming to block traffic considered a hazard in the form of a DDoS attack. We then evaluate the accuracy and performance of the proposed TBDC approach. Our technique shows outstanding performance, increasing the mean throughput by 190.3%, reducing the mean delay by 95%, and reducing packet loss by 99.7% relative to normal, with DDoS attack traffic.

KEYWORDS

Network resilience; network management; attack prediction; software defined networking (SDN); distributed denial of service (DDoS); healthcare

1 Introduction

Software-defined networking (SDN) has been widely used in recent years, mainly thanks to its nature as reliable network technology that enables the control and management of the network by disaggregating or dividing the control and data planes. In other words, it enables data forwarding by disaggregating it from data-forwarding devices such as routers, then handling it by the SDN controller, which is a centralized, external, and programmable network device [1]. In contrast, in traditional networks, devices are configured and managed through a specific method, namely the vendor specific command (VSC), which clearly leads to more complex network management requirements [2]. As depicted in Fig. 1, the core architecture of SDN basically consists of three main layers; (i) application, (ii) control, and (iii) infrastructure layers.

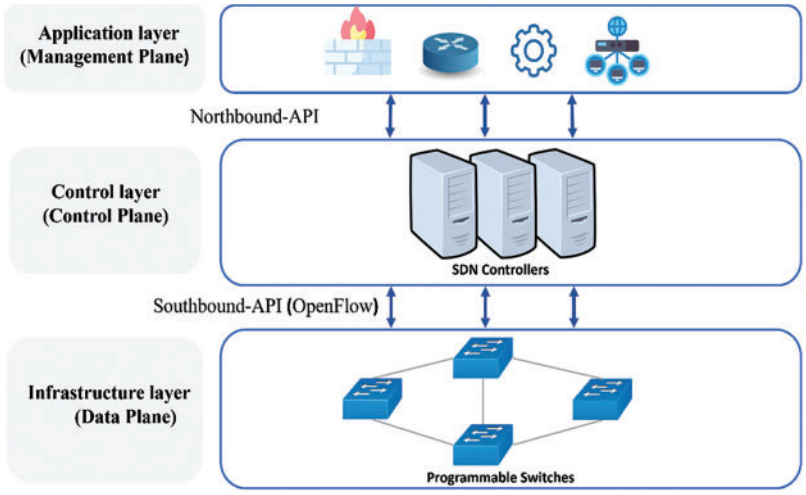

Figure 1: Software-defined networking 3 layer structures with interfaces

The ability to simplify systems while maintaining effective communication across them is a cornerstone in healthcare systems. In general, traditional network consolidation is challenging since each network device may include hundreds of configurations that must be adjusted. SDN piques a lot of interest in sectors such as healthcare, as illustrated in Fig. 2, since it allows healthcare organizations to smoothly access cloud services; large healthcare institutions need to have comprehensive visibility of their Wide Area Networks (WANs), particularly when it comes to mobile, Internet of Things (IoT), and cloud computing environments. Nowadays, electronic healthcare monitoring not only provides in-hospital service management, but further assists healthcare service providers to serve users outside health organizations, duly tracing patients' health outcomes, continuing to offer high-quality services, and detecting at-risk individuals.

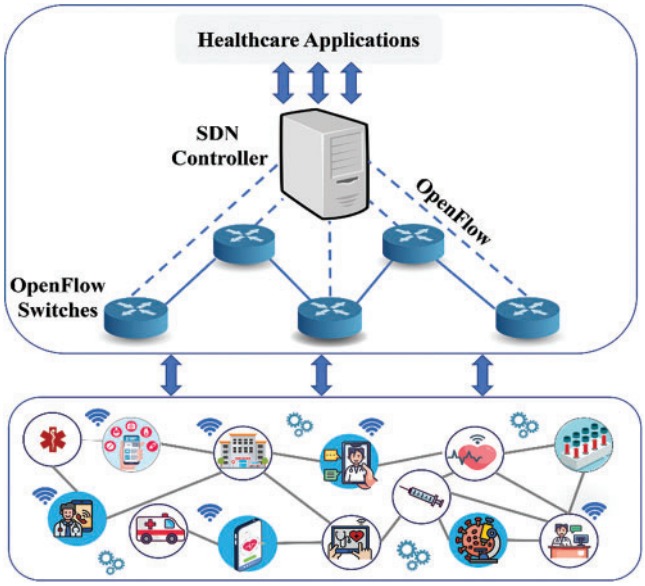

Figure 2: Architecture of SDN based healthcare systems

Furthermore, it enables patients to keep online contact with healthcare providers in real time with fast response, to be dutiful with medication schedules, and to enhance their wellness status [2,3]. Load balancing optimization can be achieved using docker swarm mode on big data applications, due to the massive data generated by healthcare edge devices [4].

However, healthcare organizations still suffer from security threats against sensitive information, which may be either clinical or financial. Researchers have implemented data sharing strategies using blockchain to share the data between healthcare edge devices without threatening patient privacy [5–7]. The most sensitive concerns are ensuring information availability, integrity, and confidentiality. A key threat to this comes from the Distributed Denial of Service Attack (DDoS), which restricts patients' data availability, since the main goal of a DDoS attack is not only to breach information and services, but also to prevent a legitimate user from accessing information whenever they need it (i.e., it affects information availability).

As reported by the Cybersecurity and Infrastructure Security Agency (CISA), identifying DDoS attacks is problematic, as this type of attack might be carried out virtually. Sensitive data can be modified by malicious intruders, and false information may be fed into different data streams by a false node [8].

Medical records include financial transactions that are linked to sensitive information, such as credit card information. Therefore, efficient detection and prevention mechanisms must be put in place to mitigate and tackle the impact of DDoS attacks. Moreover, other concerns should be tackled to ensure a high level of security and privacy, while allowing easy accessibility for electronic health applications, including high-bandwidth mobile networks, low-cost cellular network links, high availability of internet connections, and the heterogeneous platforms supported by different mobile devices [9]. Cyberattacks have increased during the COVID-19 pandemic, resulting in data violations for 90% of healthcare providers [10]. Consequently, there is a need for effective solutions to address these security risks and reduce insider threats. Researchers are exploring the use of SDNs in healthcare institutions as a means of protecting medical networks from various attacks, such as Distributed Denial of Service (DDoS) and probe attacks [11].

Intrusion refers to any unauthorized actions that harm an information system and potentially threaten its confidentiality, integrity, or availability [12]. An IDS is a software or hardware solution that detects malicious actions on computer systems to maintain system security. It identifies various types of malicious network traffic and computer usage that cannot be detected by a traditional firewall [13]. The purpose of an IDS is to provide high protection against actions that compromise the availability, integrity, or confidentiality of computer systems. IDS classification can be based on the data sources used to identify abnormal activities. There are two main types of IDS technologies [12]: Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS examines data that originates from the host system and audit sources, such as operating system logs, firewall logs, application system audits, or database logs. HIDS is capable of detecting insider attacks that don't involve network traffic. NIDS monitors network traffic extracted from a network through methods such as packet capture and NetFlow. NIDS can be used to monitor many computers connected to a network, and it can detect external malicious activities at an early stage before they spread to other systems. However, NIDS has limited ability to inspect all data in high bandwidth networks due to the large amount of data that passes through these networks. When NIDS is deployed at multiple positions in a network topology, along with HIDS and firewalls, it provides robust, multi-tier protection against both external and insider attacks [13,14].

Therefore, in this article, the main goal is to propose an efficient, low complexity system of healthcare, based on a threshold-based DDoS classifier (TBDC). The contributions of this article are summarized as follows:

- Proposing a healthcare virtualized network based on SDN to generate new dataset
- Generating a dataset that contains normal and DDoS traffic
- Proposing an efficient, low-complexity classifier using TBDC data analysis to detect and mitigate possible DDoS attacks
- Deploying TBDC to show its impact on optimizing different network KPIs

The remaining sections are arranged as follows. In Section 2, we introduce the specific background of the study and previous works. In Section 3, we introduce the main features of the evaluation of our proposed approach, including network topologies, emulation environment, and performance metrics. Section 4 presents the emulation results and a comprehensive evaluation. Finally, Section 5 offers conclusions and suggests future work.

2 Background and Related Works

In this section, we present a brief review of related works on SDN, methods related to managing attacks, and distributed denial-of-service (DDoS) threats.

2.1 SDNs

As mentioned earlier, SDN is a method of designing the network infrastructure to provide dynamically efficient control and management for network devices by administrators, engineers, and developers, through programmable and open interfaces via the REST API. For current needs, the core requirement for SDN development is to provide the greatest possible simplicity and flexibility in different parts of the network [15].

In SDN, the control plane controls and makes decisions where the traffic goes, while the data path handles and sends the packages [16]. The basic idea behind SDN is not really new. The concept of server virtualization is about establishing a layer between the physical server and operating systems executing operations in it. A similar idea exists in relation to storage virtualization; it has now simply become the network's turn to be virtualized. This notion, too, is not particularly new; operators have already used technologies such as multiprotocol label switching (MPLS) to be able to place customers' networks on top of the operator's own physical network [17]. Development has been rapid, particularly in server virtualization. A new virtual server can be created, using a ready-made template, within a few seconds. However, if (and this is usually the case) this server is to operate on a network, a subnet must be created and firewall rules set on how this subnet may communicate with other networks. Traditionally, this is done by someone in the network team, and automation has not been as fast. Moreover, the benefit of being able to quickly create a virtual server is then lost to some extent.

Another factor is scalability. Different subnets are now logically separated with the help of virtual local area networks (VLANs). There are, according to the standard, 4096 VLANs, and this is probably enough for a typical company, but not for a cloud provider of platforms.

The next problem is the Layer 2 protocol, Ethernet. The reason why Ethernet "won" the battle for the local network against other technologies a few years ago is largely because it is a fairly simple technology, and manufacturing the hardware needed is relatively cheap. However, this protocol is, in

fact, poorly equipped for virtual environments where IP addresses move around between different platforms and, perhaps, between different data centers.

The utilization of SDN in the healthcare sector can bring several key advantages. First, it offers enhanced network flexibility, allowing for centralized management and the ability to quickly adapt to changing needs. Additionally, SDN enhances network security by providing greater visibility and facilitating the enforcement of security policies, protecting sensitive patient information. Furthermore, it can improve network performance by optimizing resource utilization and reducing congestion. SDN also has the potential to streamline clinical workflows and improve patient care by integrating medical devices and systems. SDN may be employed to present patients with data security and speed in the transfer of information from endpoint to endpoint, since the SDN controller identifies the connectivity of the patient monitoring endpoint to the network.

2.2 DDoS Attacks

There are different types of DDoS attacks, from Smurfs and Teardrops to Pings of Death. Some of the most common DDoS attacks are as follows:

- ICMP (ping) flood
- SYN flood
- Ping of Death
- Slowloris
- NTP amplification
- HTTP flood
- Zero-day DDoS attacks
- Volume-based attacks. Imperva counters these attacks by absorbing them with a global network of scrubbing centers that scale, on demand, to counter multi-gigabyte DDoS attacks.

In the past, most DDoS attacks were aimed at harming the affected company or organization by making one or more web pages inaccessible. Nowadays, it is more common for the attacker to demand a ransom to interrupt the attack. Another common approach is to use DDoS attacks as a pure distraction, whereby IT departments are kept busy while the attackers step in with ransomware (hostage programs) or try to steal data.

Considering reports from the media, authorities, and the security industry itself, it can sometimes be the case that companies focus on one type of cyber threat at a time. Unfortunately, reality is more complex than that: companies have to deal with many parallel threats. “It is never possible to settle down, and old approaches can easily return in a partly new suit,” comments Peter Gustafsson, responsible for Barracuda Networks in the Nordic region [18].

A new type of attack does not require a large botnet. Some of the companies that have recently been affected by DDoS attacks are Bandwidth, VoIP.ms, VoIP Unlimited, and Voipfone. So-called Black Storm attacks are particularly dangerous for service providers in communications. Such attacks do not require the attacker to use a large botnet, and are therefore relatively easy to carry out. In a “Black Storm” attack, the attacker sends the User Datagram Protocol (UDP) request to many devices and servers on a network. The request is “spoofed,” i.e., disguised, in this case to make it look like it is coming from other devices in the same network.

The approach then triggers a kind of snowball effect that can quickly knock out a service provider (CSP) with a storm of internal data traffic. Although the method has, so far, only been

described in tests, companies should ideally be prepared to handle such attacks. With the onset of the COVID-19 pandemic and as more people started working from home, healthcare became a target. The combination of different online services for booking and responding to tests, and the widespread use of insufficiently protected IoT devices, have contributed to a large number of healthcare activities being affected by DDoS attacks [19].

Devices that have not been updated become tools for cybercriminals. The recently discovered bot network, "Meris," which includes about 250,000 infected devices, has also become a tool for DDoS attacks. Most of these devices are not computers but routers, switches, access points for Wi-Fi, and other devices sold by one and the same Latvian company, MicroTik. Admittedly, MicroTik discovered and remedied the current vulnerability as early as 2018, but due to the nature of the devices, users are rarely in contact with MicroTik, and the majority have not made the necessary updates. This, in turn, has made MicroTik devices a tool in the hands of cybercriminals.

Although DDoS attacks remind us how complicated everyday life has become for IT security managers, there are good opportunities for stopping this type of attacks in time. Companies that operate with a modern infrastructure in application and network security, in combination with active protection against DDoS attacks, have very good chances of handling such attacks.

2.3 Previous Works

By conducting a detailed survey of existing research works regarding DDoS attacks in SDNs, it can be observed that most of the focus has been placed on developing techniques for detecting DDoS attacks, using different SDN-based architectures. These techniques mainly focus on using SDN technology in different layers (principally network, application, and transport layers) to detect and mitigate attacks. Although only minor emphasis has been given to healthcare systems, many of the approaches with different techniques can be employed in suitable settings. Techniques vary across a wide spectrum; for example, using a cloud environment, as in [20], that analyzes the effects of DDoS attacks in a hybrid cloud, as in [21]. The amount of work that has been done is significant, with a focus on using AI and machine learning, with different algorithms, to classify the traffic and detect possible attacks using intrusion detection systems (IDS), as in [22], or to mitigate them, as in [23,24], and [25,26].

Researchers have also developed IDS to mitigate packet drop/modification attacks, badmouthing attacks, on-off attacks, and collusion attacks based on trust similarity [27]. Table 1 shows a summary of related works with different types of mechanisms used against DDoS attacks.

Table 1: Summary of related works based on security against DDoS attacks

Source	Defense type	Location	Controller	Objective
Dao et al. [28]	Detect and mitigate	Both data and control planes	ISP	Propose a context-aware security approach to detect attacks in small networks.
Phan et al. [29]	Detect	Control plane	Floodlight	Propose a high-accuracy detection framework that deals with a man-in-the-middle attack.

(Continued)

Table 1 (continued)

Source	Defense type	Location	Controller	Objective
Chin et al. [30]	Detect	Data plane	OpenDaylight	Detect flooding in SDN networks in an effective and scalable manner.
Tortonesi et al. [31]	Mitigate only	Data plane	Pox	Propose a DDoS mitigation approach that is able to perform edge defense.
Ravi et al. [32]	Detect and mitigate	Both data and control planes	Pox	Propose a novel DDoS detection and defense model depending on real time trained ML algorithms.
Wang et al. [33]	Detect and mitigate	Data plane	Pox	Orchestrate a deep learning (DL)-based intrusion detection system (IDS) to detect DDoS attack traffic in SDNs.
Mbasuva et al. [34]	Detect only	Data plane	OpenDaylight	Orchestrate a DL-based IDS to detect DDoS attack traffic in SDNs.

3 Evaluation

In this part, we introduce our evaluation environment. This includes the network topology, metrics used to measure the performance, and tools and methodology used for experiments.

3.1 Network Topology

In this section, we present a typical network topology (see Fig. 3) that is used for healthcare systems to analyze the performance in a normal situation, and the performance under DDoS attacks. In this system, OpenFlow switches, numbered as shown from 1 to 3, are connected to the SDN controller responsible for traffic flow management. The OpenFlow protocol is mainly used for connecting the control-plane SDN controller and forwarding plane switches. In addition, there are 18 hosts, representing different clients that generate data traffic, named either as devices or workstations. The network includes two other hosts; one serves as the patients' database and the other as the patient portal server, receiving data traffic from the clients.

3.2 Performance Metrics

Network performance measurements are determined by several KPIs. In this work, we measure the throughput of healthcare applications under different scenarios. First, we track throughput without DDoS attacks and challenges of the proposed TBDC system, where traffic travels over different routes with no network congestion.

After that, we track the throughput of healthcare applications without using the TBDC, then in the case of DDoS attacks through one route to the destination with no prioritization and queuing to split data flows. Finally, we compute the throughput of healthcare applications under DDoS attack conditions by integrating the proposed TBDC system to analyze the performance considering network resilience and quality of service (QoS) for multiple applications.

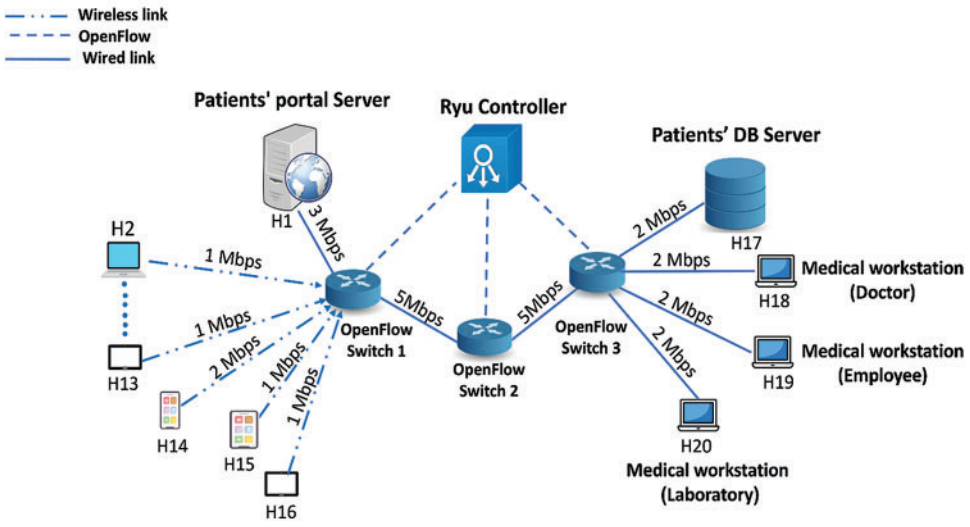

Figure 3: Evaluation network topology in the absence of attacks (normal traffic)

3.3 Experimental Setup

In this work, experiments were performed on Ubuntu 20, equipped with 5.9 GB RAM and a 1.80 GHz processor. Network emulation was done using Mininet 2.3.0 with a Ryu controller, as shown in Table 2. Mininet is a virtual network simulator that bears on OpenFlow and software-defined networking using per-process virtualization. It provides a rapid prototyping workflow for creating and customizing software consisting of hosts and switches on a single computer. The model is written in Python, and traffic was generated with and intercepted using Hping3 and distributed internet traffic generator (D-ITG). Results were analyzed with D-ITG.

Table 2: Emulation parameter values

Parameter	Values
Emulator	Mininet 2.3.0
Operating System	Ubuntu 20.04.3
Memory	5.9 GB of RAM
CPU	Intel core i7-8565U @ 1.80 GHz
Traffic Generator	DITG, Hping3
Link Bandwidth	1, 2, 5 Mbps
SDN Framework	Ryu
Open vSwitch	2.13.3

4 Results and Discussion

In this part, the evaluation of the proposed TBDC system is presented and discussed in detail.

4.1 Normal Network Performance of Healthcare Systems (Normal Traffic)

In this case, we implement the network shown in Fig. 3 using D-ITG. All applications installed on the different workstations/end devices connected to the switches are set to start sending to the

servers. These applications use paths shown in figures through the switches. These switches are typically deployed to work using the spanning tree protocol to prevent any storming of broadcasts. They are not implemented to increase any QoS factors. Here, we expect normal operation with no lost packets, so all to be delivered. Figures provided here show the performance of the network without challenge. Results in Figs. 6–8 show the normal behavior of the network, with the load assigned for this test, with extra delay due to link limitations. The performance is stable without any abnormalities, as expected, due to the appropriate capacity and threat-free nature. A maximum throughput of 53 packets per second (PPS) is reached, and stays stable until the end of 50s of emulation. A packet loss of zero and a low delay, close to zero, confirmed the stable behavior of the network in absence of attacks.

4.2 Effects of DDoS Attacks on Network Performance (Normal Traffic with DDoS Attacks)

In this scenario, we deploy the network with DDoS attacks, as shown in Fig. 4, using D-ITG and Hping 3. The attack, which is performed by 12 hosts from H2 to H13, targets the patient portal server connected to switch 1 using SYN and UDP attacks, through devices connected to switch 1. The effect of the new challenge (DDoS attacks) can now be seen in extra delay and packet loss, as well as reduced throughput, as shown in Figs. 6–8. As expected, attacks negatively affect the performance of the network, even with low loads assigned in the emulations. Throughput decreases significantly from 20% to 100%, leading to a 52% reduction in the mean throughput. Delay increases significantly, reaching 2.5 s in the worst case, resulting in the mean delay increasing to 1s. Packet loss increases to a peak of over 160 pps, increasing the mean packet loss to around 67 pps.

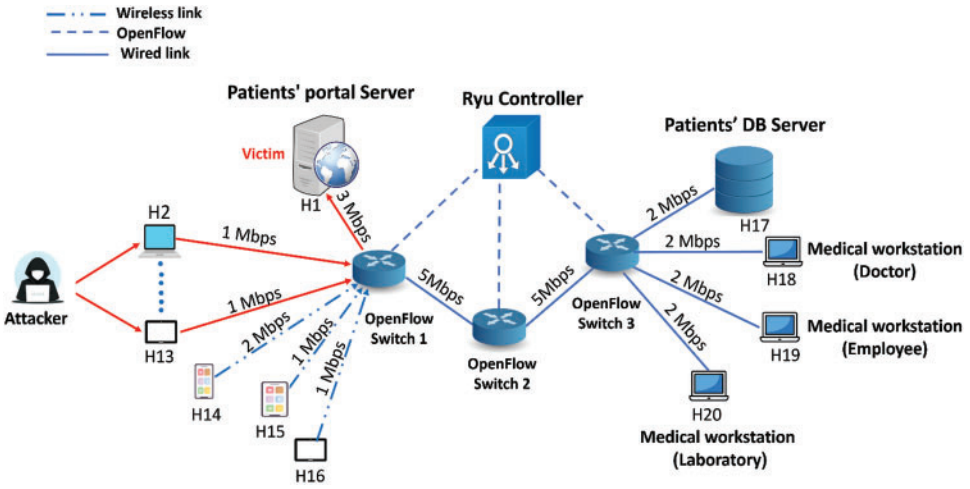

Figure 4: Evaluation network topology in normal traffic with DDoS attacks

4.3 TBDC Approach Deployment and Performance

In this work, we propose a threshold-based DDoS classifier (TBDC) aiming to classify suspicious traffic to help detect the possibility of DDoS attacks and block suspicious incoming traffic. To deploy the approach and analyze its performance, we designed a monitor application using a Ryu controller to extract features from the network. These features include datapath ID, input port, MAC address destination, output port, number of packets, number of bytes, and duration. (EventOFPFlowStatsReply) callback has been used to request the statistics of each flow from every switch on the network. A total of 265 samples were collected using the developed monitor application, out of which 217 represented normal traffic, while the remaining 48 samples included DDoS attack traffic.

The deployment included some pre-processing steps. First, samples having a value of zero in duration or packets were replaced with 0.9. The second step involved calculating the throughput in kbit/sec, packet rate, and bytes-to-packet ratio, using collected features. By analyzing the data in scenarios mentioned above, it became clear that throughput and packets per second are non-linear features; however, it can still be proved that bytes-per-packet is the only linearly separable feature between attack and normal samples, as can be seen in Fig. 5.

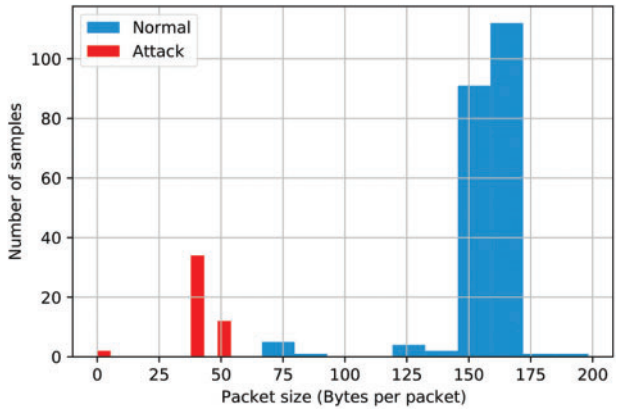

Figure 5: Distribution of samples by size (bytes per packet) for an SDN with DDoS attacks

We performed a two-sample z-test, which is a statistical test used to determine whether two population means with a sample size greater than 30 are different. The null hypothesis states that the attack and normal packet size means are equal, while the alternative hypothesis states that the attack and normal packet size means are not equal. The z-test can be calculated using Eq. (1), with a significance level of 0.05 that determines whether we can accept or reject the null hypothesis. The test statistic for the two-sample z-test is 46.315, and the corresponding p-value is 0. Since this p-value is less than the significance level, we have sufficient evidence to reject the null hypothesis, which implies that the mean packet size significantly differs between attack and normal samples. Therefore, it is possible to use the feature of bytes per packet as a threshold to separate normal and attack traffic.

In the emulation scenario, we chose a threshold value of 57 bytes/packet to differentiate between attack and normal samples, since the two classes are linearly separable using this feature, achieving an accuracy of 99% on the collected data. Accuracy was calculated by dividing the correctly classified samples by the total number of samples. The predictions of TBDC were compared with the ground truth of the collected data.

The TBDC system has five main functions: GetAllFlows(network); GetFlowFeatures(flow); Zero Replacement(feature); CalculateByteToPacket(number_of_bytes, number_of_packets); and Block-Flow(input_port, mac_address_destination, output_port).

The pseudocode of the TBDC algorithm is illustrated in Algorithm 1. Initially, GetAllFlows (network) determines all the active flows on network topology. The function results in a set of active flows. This function is followed by a "for" loop iterating over all the active flows. For each iteration, input port (input_port), receiver Mac address (mac_address_destination), output port (output_port), number of packets (number_of_packets) and number of bytes (number_of_bytes) are collected using GetFlowFeatures(flow). The preprocessing method was implemented using ZeroRe-placement(feature), which replaces zero values of the number of packets with 0.9.

This method was followed by CalculateByteToPacket (number_of_bytes, number_of_packets), which performs division feature transformation on number_of_bytes and number_of_packets. Finally, the proposed threshold is compared to the transformed features which block the flow of the attacker, using the BlockFlow (input_port, mac_address_destination, output_port) function if they fall behind the threshold.

$$Z = \frac{(\overline{x}_1 - \overline{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{1}$$

Equation 1: Two-sample z-test equation

where $\overline{x}_1$, μ_1, and σ_1^2 are the sample mean, population mean and population variance respectively for the attack sample. $\overline{x}_2$, μ_2, and σ_2^2 are the sample mean, population mean and population variance respectively for the normal sample.

Algorithm 1: TBDC algorithm.

Functions:
GetAllFlows(*network*): get the active flows for a given network.
GetFlowFeatures(*flow*): get the statistics for a given flow.
ZeroReplacement(*feature*): replace zero values for a given feature with 0.9.
CalculateByteToPacket(*number_of_bytes, number_of_packets*): calculate bytes to packet ratio.
BlockFlow(*input_port, mac_address_destination, output_port*): generate a rule to block a given flow.
Input:
network: the switches in the proposed topology.
Output:
decision _block: Decide whether to allow or block a given flow.

```
begin
    Flows = GetAllFlows(network)
    for flow in Flows do
            input_port, mac_address_destination, output_port, number_of_packets, number_of_bytes
            =\
            GetFlowFeatures(flow)
            number_of_packets = ZeroReplacement(number_of_packets)
            packet_size = CalculateByteToPacket(number_of_bytes, number_of_packets)
            if packet_size <= 57 do
                    decision _block = True
                    BlockFlow(input_port, mac_address_destination, output_port)
            else do
                    decision_block = False
            end
    end
    return decision_block
end
```

When looking at Figs. 6–8, it is clear that the performance of the network was significantly improved by applying the proposed TBDC approach to block unwanted traffic. The significant throughput increase reached 190.3% on average. Packet loss was reduced by 99.7% on average due to mitigation using the TBDC approach. Similarly, delay was also reduced by 95.5% on average relative to the normal level associated with DDoS attack traffic.

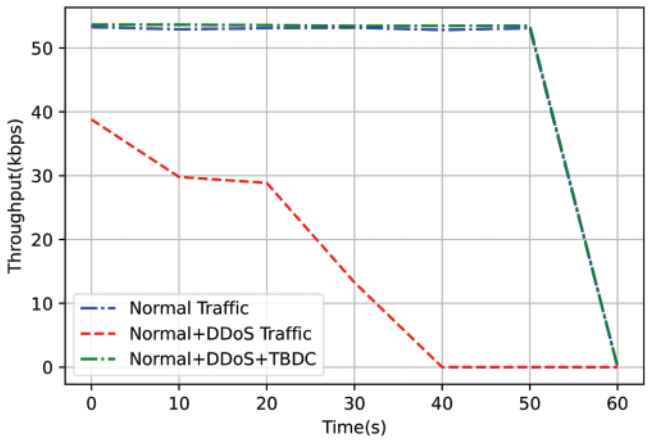

Figure 6: Throughput evaluation for normal traffic scenario *vs*. normal traffic with DDoS attacks *vs*. TBDC

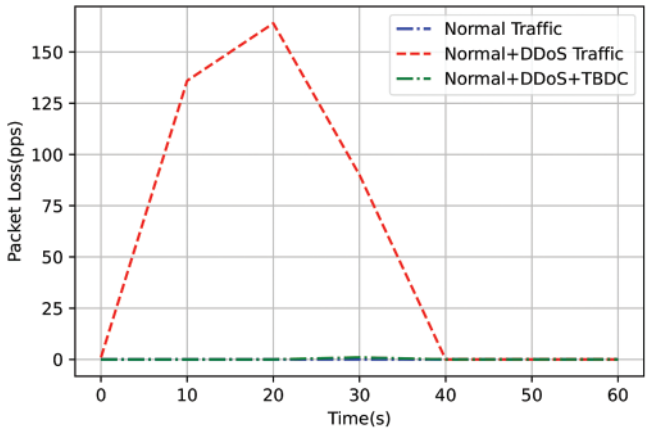

Figure 7: Packet loss evaluation for normal traffic scenario *vs*. normal traffic with DDoS attacks *vs*. TBDC

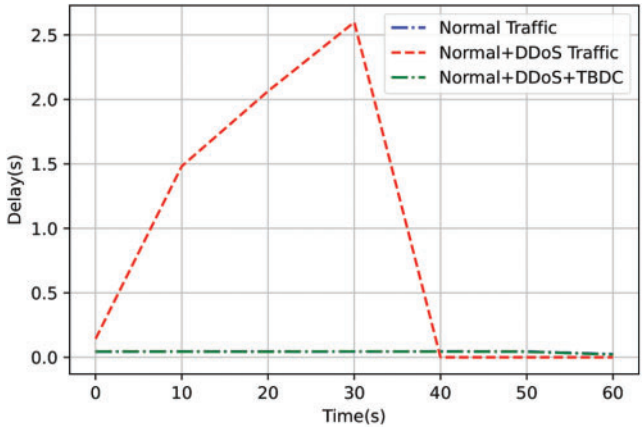

Figure 8: Delay evaluation for normal traffic scenario *vs*. normal traffic witt DDoS attacks *vs*. TBDC

TBDC also outperformed other IDSs on the accuracy metric, achieving an accuracy of 99% as shown in Table 3.

Table 3: Comparison between related work and TBDC

Source	Accuracy
Phan et al. [29]	98.13%
Ravi et al. [32]	96.28%
Wang et al. [33]	95.41%
TBDC	99%

5 Conclusions and Future Work

To conclude, in this paper, we proposed a low-complexity mitigation technique for DDoS attacks in healthcare systems' SDNs. Despite the existence of various complex mitigation techniques, the low-complexity TBDC approach can be used easily and efficiently to achieve a performance similar to that in attack-free traffic. Throughput has increased by 190.3% on average, while packet loss and delay were reduced by 99.7% and 95.5% on average respectively relative to the normal level associated with DDoS attack traffic. By making use of a generic healthcare SDN model, we showed that using a threshold-based technique is efficient in mitigating possible attacks. By analyzing network performance using the TBDC technique, significant improvement was observed in the total system throughput, as well as a reduction in both delay and packet loss, which leads to performance similar to that of an attack-free network.

TBDC still has limitations as it is currently dedicated to DDoS attacks only. Furthermore, the estimation of the threshold value is dependent on the network usage and behavior. TBDC has a beneficial impact on the whole spectrum of the health sector in defense against the increasing rate of cybersecurity threats such as phishing, ransomware, and especially DDoS attacks, not only during a healthcare crisis such as the COVID-19 pandemic but also in the long term.

For the next steps, we intend to analyze the TBDC approach performance for different, more diverse network topologies, because we only showed here a limited healthcare system network example to measure the performance of the TBDC solution. In addition, we plan to extend system deployment and emulations while dynamically rerouting traffic in the presence of DDoS attacks and present an IDS using ML algorithms.

Funding Statement: The authors extend their appreciation to Researcher Supporting Project Number (RSPD2023R582), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Jarschel, M., Zinner, T., Hoßfeld, T., Tran-Gia, P., Kellerer, W. (2014). Interfaces, attributes, and use cases: A compass for SDN. *IEEE Communications Magazine, 52(6),* 210–217.

2. Bedhief, I., Kassar, M., Aguili, T. (2018). From evaluating to enabling SDN for the internet of things. *IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8. Piscataway, IEEE.
3. Jang-Jaccard, J., Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences, 80(5),* 973–993.
4. Singh, N., Hamid, Y., Juneja, S., Srivastava, G., Dhiman, G. et al. (2023). Load balancing and service discovery using Docker Swarm for microservice based big data applications. *Journal of Cloud Computing, 12(1),* 1–9.
5. Lian, Z., Zeng, Q., Wang, W., Gadekallu, T. R., Su, C. (2022). Blockchain-based two-stage federated learning with Non-IID data in IoMT system. *IEEE Transactions on Computational Social Systems*. https://doi.org/10.1109/TCSS.2022.3216802
6. Wang, W., Chen, Q., Yin, Z., Srivastava, G., Gadekallu, T. R. et al. (2021). Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks. *IEEE Internet of Things Journal, 9(11),* 8883–8891.
7. Zhang, L., Zou, Y., Yousuf, M. H., Wang, W., Jin, Z. et al. (2022). BDSS: Blockchain-based data sharing scheme with fine-grained access control and permission revocation in medical environment. *KSII Transactions on Internet & Information Systems, 16(5),* 1634–1652. https://doi.org/10.3837/tiis.2022.05.012
8. Rich, S. (2022). Enhancing cybersecurity to protect america's data. Public Policy Center, University of Iowa. ppc.uiowa.edu/publications/enhancing-cybersecurity-protect-americas-data
9. Ray, S., Mishra, K. N., Dutta, S. (2022). Detection and prevention of DDoS attacks on M-healthcare sensitive data: A novel approach. *International Journal of Information Technology, 14(3),* 1333–1341.
10. Williams, C. M., Chaturvedi, R., Chakravarthy, K. (2020). Cybersecurity risks in a pandemic. *Journal of Medical Internet Research, 22(9),* e23692.
11. Hasan, R., Zawoad, S., Noor, S., Haque, M. M., Burke, D. (2016). How secure is the healthcare network from insider attacks? An audit guideline for vulnerability analysis. *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, pp. 417–422. Piscataway, IEEE.
12. Ponnusamy, V., Humayun, M., Jhanjhi, N. Z., Yichiet, A., Almufareh, M. F. (2022). Intrusion detection systems in internet of things and mobile ad-hoc networks. *Computer Systems Science and Engineering, 40(3),* 1199–1215. https://doi.org/10.32604/csse.2022.018518
13. Malasri, K., Wang, L. (2009). Securing wireless implantable devices for healthcare: Ideas and challenges. *IEEE Communications Magazine, 47(7),* 74–80.
14. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2(1),* 1–22.
15. Khan, S., Gani, A., Wahab, A. W. A., Guizani, M., Khan, M. K. (2017). Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art. *IEEE Communications Surveys & Tutorials, 19(1),* 303–324.
16. Alenazi, M. J., Cetinkaya, E. K. (2020). Resilient placement of SDN controllers exploiting disjoint paths. *Transactions on Emerging Telecommunications Technologies, 31(2),* e3725.
17. AlZoman, R., Alenazi, M. J. (2020). Exploiting SDN to improve QoS of smart city networks against link failures. *2020 Seventh International Conference on Software Defined Systems (SDS)*, pp. 100–106. Piscataway, IEEE.
18. Meng, W., Choo, K. K. R., Furnell, S., Vasilakos, A. V., Probst, C. W. (2018). Towards Bayesian-based trust management for insider attacks in healthcare software-defined networks. *IEEE Transactions on Network and Service Management, 15(2),* 761–773.
19. Qadri, Y. A., Nauman, A., Zikria, Y. B., Vasilakos, A. V., Kim, S. W. (2020). The future of healthcare Internet of Things: A survey of emerging technologies. *IEEE Communications Surveys & Tutorials, 22(2),* 1121–1167.

20. Wang, B., Zheng, Y., Lou, W., Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and software-defined networking. *Computer Networks, 81,* 308–319.
21. Zaalouk, A., Khondoker, R., Marx, R., Bayarou, K. (2014). OrchSec: An orchestrator-based architecture for enhancing network-security using network monitoring and SDN control functions. *IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9. Piscataway, IEEE.
22. Alzahrani, A. O., Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet, 13(5),* 111.
23. Yungaicela-Naula, N. M., Vargas-Rosales, C., Perez-Diaz, J. A. (2021). SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access, 9,* 108495–108512.
24. Gu, X., Wang, H., Ni, T., Ding, H. (2013). Detection of application-layer DDoS attack based on time series analysis. *Journal of Computer Applications, 33(8),* 2228–2231.
25. Revathi, M., Ramalingam, V. V., Amutha, B. (2021). A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework. *Wireless Personal Communications, 127,* 1–25.
26. AlZoman, R. M., Alenazi, M. J. (2021). A comparative study of traffic classification techniques for smart city networks. *Sensors, 21(14),* 4677.
27. Kavitha, A., Reddy, V. B., Singh, N., Gunjan, V. K., Lakshmanna, K. et al. (2022). Security in IoT mesh networks based on trust similarity. *IEEE Access, 10,* 121712–121724.
28. Dao, N. N., Park, J., Park, M., Cho, S. (2015). A feasible method to combat against DDoS attack in SDN network. *2015 International Conference on Information Networking (ICOIN)*, pp. 309–311. Piscataway, IEEE.
29. Phan, T. V., Bao, N. K., Park, M. (2016). A novel hybrid flow-based handler with DDoS attacks in software-defined networking. *International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 350–357. Piscataway, IEEE.
30. Chin, T., Mountrouidou, X., Li, X., Xiong, K. (2015). An SDN-supported collaborative approach for DDoS flooding detection and containment. *MILCOM 2015-2015 IEEE Military Communications Conference*, pp. 659–664. Piscataway: IEEE.
31. Tortonesi, M., Michaelis, J., Morelli, A., Suri, N., Baker, M. A. (2016). SPF: An SDN-based middleware solution to mitigate the IoT information explosion. *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 435–442. Piscataway, IEEE.
32. Ravi, N., Shalinie, S. M. (2020). Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. *IEEE Internet of Things Journal, 7(4),* 3559–3570.
33. Wang, M., Lu, Y., Qin, J. (2021). Source-based defense against ddos attacks in SDN based on sFlow and SOM. *IEEE Access, 10,* 2097–2116.
34. Mbasuva, U., Zodi, G. A. L. (2022). Designing ensemble deep learning intrusion detection system for DDoS attacks in software defined networks. *16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–8. Piscataway, IEEE.