DYNAMIC THRESHOLD GENERALIZED ENTROPIES-BASED DDOS DETECTION AGAINST SOFTWARE-DEFINED NETWORK CONTROLLER

HUSSEIN A. AL-OFEISHAT [1]

[1] Computer Engineering Department, Al-Balqa Applied University, Al-Salt, Jordan.

E-mail**:** [1] Ofeishat@bau.edu.jo

ABSTRACT

This study presents a new strategy that combines dynamic thresholding with generalized entropy approaches to identify Distributed Denial-of-Service (DDoS) attacks in Software-Defined Networking (SDN) environments. This research intends to address the limitations of traditional detection approaches, such as high false-positive rates and restricted adaptability. We conducted thorough testing in various simulated SDN scenarios to assess the efficacy of our approach compared to existing methodologies. The findings exhibited a substantial enhancement in both the precision of detection and the decrease in false positive occurrences, signifying a remarkable progression compared to existing techniques. This research not only fills an important void in the realm of SDN security but also lays the foundation for more adaptable and efficient ways for detecting DDoS attacks. The results have practical implications for improving network security by providing a strong answer to the changing danger of DDoS attacks in intricate network environments. In summary, this study offers a fresh viewpoint to the field of SDN security research, proposing a possible change in DDoS detection methods towards more flexible and entropy-driven methodologies.

Keywords*:* *Network; SDN Controller; Telecommunications Infrastructures; Renyi; Networking; SDN.*

1. INTRODUCTION

The proliferation of information and communication technology over the past few decades has greatly increased both network traffic and the complexity of procedures required to analyze the enormous volumes of data generated during this time [2]. Current conventional network architecture struggles to handle heavy traffic loads, which can lead to packet loss and delivery delays. Several issues with personal information and data privacy may result from this. Researchers have proposed SDN, or software-defined networking, as a possible solution due to its programmability, flexibility, and security advantages over conventional networking [2, 10].

In the early 2000s, Feamster et al. [7] created software-defined networking (SDN) architecture to overcome traditional network restrictions. Due to this, many firms and individuals participated in research and development, which improved the technology's performance, scalability, dependability, security, and ability to manage vast network traffic [8, 14, 27].

In the coming years, software-defined networks (SDNs) will replace conventional networks in charge of regulating traffic. With SDN, network traffic is better managed, resulting in lower expenses for the data center. Figure 1 demonstrates that software-defined networking (SDN) will be widely deployed in data centers to manage network traffic in the near future according to a 2018 Cisco report [5].

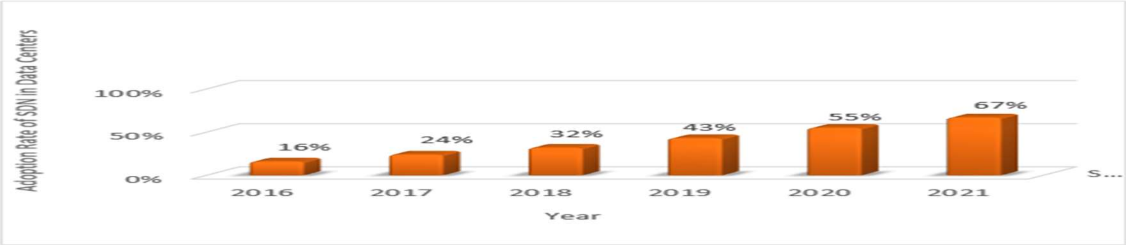


Figure 1 : SDN Adoption, 2016–2021 [5].

By 2020, as depicted in Figure 1, SDN technology will be widely utilized by data centers due to its cheaper cost and greater efficiency in controlling network traffic. This demonstrates the importance of SDN for the development of IT and data exchange in the future. Therefore, this justifies the efforts made in this research to strengthen the safety of the SDN environment.

The SDN enables complete control over network properties, allowing them to evolve with the changing requirements of enterprises. Existing networks (conventional networks) address these requirements, but they prevent administrators from exercising full control over the network without risking vendor lock-in by configuring all operations and controlling them via vendor-centric proprietary hardware and software. To keep up with the most recent developments in network technology, a reorganization is necessary in which switches are designed to forward packets and receive instructions rather than relying on their own resources (service providers) to handle newly arriving packets [6]. Changes in network administration and fresh strategies for old problems have resulted from the advent of software-defined networking (SDN). Therefore, SDN is different from traditional networks in several respects. An important conceptual distinction is a split between the control plane and the data plane. This isolation permits SDN to manage the entire network with the help of a centralized controller, which boosts the network's efficiency and flexibility (4, 13, 25). Unmatched packets from the switch's flow table will be forwarded to the controller via an application programming interface. Some researchers have drawn parallels between the functions of the SDN controller and the human brain in terms of their shared responsibility for controlling and monitoring the activity of network traffic to ensure its smooth and efficient operation [8]. Figure 2 from Zhang et al. [33] provides a bird's eye view of the three tiers of SDN architecture.

The rising popularity of software-defined networking (SDN) is largely attributable to the requirements of big data management, which call for a programmable controller with the flexibility to accommodate a wide range of network traffic flows and the ability to configure new instructions or rules to process incoming traffic [16]. DDoS attacks are becoming more severe to destroy the SDN controller. The most serious type of network attack is a distributed denial of service (DDoS) attack, which can happen at a variety of attack rates and has far-reaching impacts on data security, productivity, and the economy. Statista's 2016 forecast, shown in Figure 2, projects that SDN revenue would have topped USD 28.1 billion by 2022 [28]. These results show that SDN is worth adopting because of the savings it can provide in the long run, and the technology is only going to get better over time. However, as SDN grows in popularity, it also becomes more susceptible to assaults that aim to undermine its fundamental capabilities—including those related to management, adaptability, and security—to disrupt networks and cause disruptions.

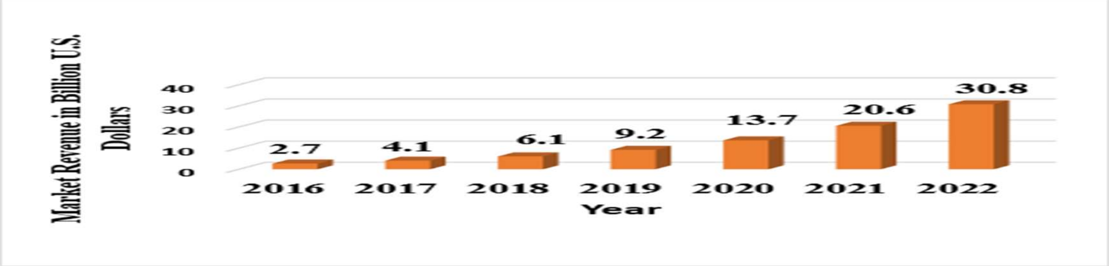


Figure 2 : From 2016 through 2022, worldwide revenue from the software-defined networking (SDN) market is depicted in [28].

In this research, we provide a statistical method for detecting distributed denial-of-service (DDoS) assaults against a software-defined network (SDN) controller with a high detection rate and a low false positive rate (single or multiple hosts). The following goals have been set for this paper to help it accomplish its primary purpose:

- A Distributed Denial-of-Service (DDoS) assault on the SDN controller can be detected by looking at two properties of the packet header. The method used here is based on information theory, and this study generalizes it.
- The second objective is to boost detection while decreasing false positives. The

incoming traffic volume will determine the threshold adjustment.

- Thirdly, we'd like to supply a rule-based detection mechanism for denial-of-service attacks.

A statistical strategy for detecting DDoS assaults on the SDN controller is proposed as the main contribution of this research. Regardless of the volume or origin of the DDoS attack traffic, this approach has a high detection rate and a low false positive rate (single or multiple hosts). Considering these goals, the following is a summary of the study's contributions:

- A technique enabling an SDN controller to identify a distributed denial of service (DDoS) assault, either on a single host or across several hosts, is developed using information theory.
- To lower the false positive rate of the detection system, a dynamic threshold is used to adapt to the attack traffic flow.
- To ensure the efficacy of the proposed strategy, we employ a rule-based attack detection system.

The beginig of the study was by reviewing the relevant literature Abu Alghanam et, [1] al and analyzing the potential issues with current detection methods to better understand the challenges we will face in our effort to secure the SDN controller. The research gap and the difficulties it brings are supposed to be highlighted by highlighting the shortcomings of the methodologies that are currently being used. The next step involves proposing a solution to the research challenge [3]. The solution is composed of four distinct phases that mutually guarantee both a high detection rate for DDoS attacks and a low rate of false positives [19]. The proposed approach employs a statistical traffic analyzer, a dynamic threshold, and a rule-based detection algorithm to accomplish this. The third step is to create a plan to achieve the research goal and put it into action. Which better utilizes the proposed approach by employing all its processes to detect a DDoS assault against an SDN controller using tools that collect statistics. The fourth phase focuses on assessing how well the research has done its job. By detailing the results of a pilot test of the proposed method and examining its performance [15]. The proposed technique is put through its paces to ensure it can effectively detect DDoS attacks while reducing the number of false positives. After comparing it to other attack detection strategies, we draw conclusions.

Despite advancements in network security, DDoS attacks in Software-Defined Networking (SDN) environments remain a formidable challenge. Current detection methods, while effective to a degree, often struggle with high false-positive rates and lack adaptability to diverse attack scenarios. This study identifies a critical gap in the existing methodologies: the need for a more dynamic and flexible approach to accurately detect DDoS attacks with minimized false positives.

This research addresses the identified gap in DDoS detection within SDN environments by introducing a novel method that combines dynamic thresholding with generalized entropy. Our approach not only contributes to reducing false positives significantly but also adapts efficiently across varied attack scenarios, a notable advancement over existing techniques. Thus, this study not only fills a crucial gap in SDN security research but also lays the groundwork for future developments in network defense strategies.

2. METHODS AND MATERIALS

2.1 Data collection and preprocessing

The first step in any analysis is collecting and formatting the relevant data. As of right now, the dataset is still in the process of being prepared. The results of this phase will be judged in part by the characteristics and data preparation techniques used. To identify malicious behavior, such as a distributed denial-of-service assault, detection techniques look at the packet headers. That's why the suggested solution relies on the controller in an SDN network having access to the traffic flow data gathered from the hosts [31].

One of the first things that must be done with the suggested method is to record and preprocess the traffic. In this step, we obtain data from a trustworthy source, record both normal and abnormal packets, then remove or ignore elements in the packets that aren't essential for spotting DDoS attacks. To make the suggested method more effective than previous detection methods, a first stage generates a set of packet properties. These steps, which occur in three distinct stages, are explained in more depth below [22].

Packet Capturing

The first step in any analysis is collecting and formatting the relevant data. As of right now, the dataset is still in the process of being prepared. The results of this phase will be judged in part by the characteristics and data preparation techniques used. To identify malicious behavior, such as a distributed denial-of-service assault, detection techniques look at the packet headers. That's why the suggested solution relies on the controller in an SDN network having access to the traffic flow data gathered from the hosts [20].

One of the first things that has to be done with the suggested method is to record and preprocess the traffic. In this step, we obtain data from a trustworthy source, record both normal and abnormal packets, then remove or ignore elements in the packets that aren't essential for spotting DDoS attacks. To make the suggested method more effective than previous detection methods, a first stage generates a set of packet properties. These steps, which occur in three distinct stages, are explained in more depth below [32].

Packet Filtering

As such, it is crucial to filter out irrelevant packet traffic (or "traffic") in order to detect attacks more accurately. To accomplish this, we ensure that there are no weak links in the packet processing procedure. This research suggests stripping packets of non-critical header information in favor of those that can better detect DDoS attacks as a means of protecting the SDN controller. To eliminate unnecessary inbound packets, we now apply filtering methods [23].

1. UDP packet Filtration

Only UDP packets will be considered for this research; all other traffic packets will be disregarded. Figure 3 is a flowchart depicting the filtering of UDP packets. The second step, packet feature filtration, takes the filtered UDP packet and selects features that aid in detecting DDoS attacks of varying traffic volumes [30].

2. Packet Features Selection

The second step in filtering network packet traffic is to pick the most crucial attributes of the filtered packets that have a demonstrable influence on improving the detection accuracy of the proposed approach, as shown in Table 1. The proposed method makes use of these features, which are retrieved from the packet header. Discarding information that isn't directly pertinent to the discussion at hand [11].

Table 1: List of the Packet Header Features

Fields	Description
Source IP	Sender host IP address
Destination IP	Receiver host IP address

Selecting the most critical properties of the filtered packets that have a demonstrable influence on enhancing the detection accuracy of the proposed approach is the second step in filtering network packet traffic, as illustrated in Table 1. These characteristics are collected from the packet header and used in the suggested technique. Ignoring details that aren't crucial to the conversation at hand.

Flow Construction

The features obtained at this point will be used to construct the packet flow. The creation of the flows relies on a time window (t), which can be adjusted based on observation of network activity or through experimentation. This "flow" can be seen as belonging to one of two groups. There are two types of traffic flows: the first is governed by the IP address of the computer doing the sending, and the second by the IP address of the computer doing the receiving [17].

The stage produces two types of flows reflecting the source IP address and the destination IP address, which are then fed back into the stage. In this stage, we make strides toward our primary research objective.

Joint Entropy in the Generalized Renyi Algebra

The suggested method detects both low- and high-rate DDoS attempts against the SDN controller using information extracted from packet headers. The attacking host or hosts could be isolated, or they could be part of a larger network that targets numerous hosts at once. Unicast Datagram Protocol (UDP) packets exhibiting erratic behavior are characteristic of distributed denial of service (DDoS) attacks. Some current detection techniques use the Shannon entropy method to discover unpredictability in UDP packets [12,18]. However, relying on the Shannon entropy may lead to a poor DDoS attack detection rate and a high false positive rate when used to identify low-rate DDoS attacks that target numerous victim servers. Unfortunately, the Shannon entropy method is not particularly helpful for detecting DDoS attacks because it relies on a single header packet feature as input to calculate the entropy value and then compares that value to a fixed threshold to determine if the network traffic flow exhibits the behavior of an attack [29].

Combining the joint entropy method with the Renyi approach, we offer generalized Renyi joint entropy in this research. To measure the degree of association between two independent random variables (x and y), such as the source IP address and the destination IP address in a packet's header, the generalized Renyi joint entropy is presented. You may express the suggested Renyi Joint Entropy Method as a formula, and it looks like this [33]:

$$\boldsymbol{H_{RJ\alpha}(x,y) = \frac{1}{1-\alpha}\, log_2\left(\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i y_j)^{\alpha}\right)} \quad (1)$$

This situation involves a probability distribution between the source IP (x) and the destination IP (y) at time t is denoted by $P(x_i y_j)$, where $\boldsymbol{H_{RJ\alpha}(x,y)}$ is a positive value denoting a Renyi Joint Entropy, and t, the time interval.

How likely it is that the source is to have caused the IP (x) and the destination IP (y) at time t is denoted by $\boldsymbol{H_{RJ\alpha}(x,y)}$, where $P(x_i y_j)$ is a number in the positive, where x and y represent the IP addresses of the computer doing the sending and receiving.

The suggested method uses IP frequency data to calculate the likelihood of each source and destination IP address pair, and then adjusts the Renyi Joint Entropy to maximize the detection rate$(X = xi, Y = yj), i = 1, 2, \dots, N\ and\ j = 1, 2, \dots, M, P(x_i y_j) > 0$.

When all hosts have an equal chance of receiving any given packet, the Renyi Joint Entropy is at its highest. The most likely time for a packet to reach its target host should coincide with the lowest possible Renyi Joint Entropy.

Multiplying the probabilities over a given time of the collected source IP addresses (xi) and destination IP addresses (yj) yields the Renyi Joint Entropy. In Equations 1 and 2, you may calculate the probability of xi and yj.

$$\boldsymbol{p_{x_i} = \frac{x_i}{n}} \quad (2)$$

$$\boldsymbol{p_{yj} = \frac{y_j}{n}} \quad (3)$$

When all hosts have an equal chance of receiving any given packet, the Renyi Joint Entropy is at its highest. The most likely time for a packet to reach its target host should coincide with the lowest possible Renyi Joint Entropy. Multiplying the probabilities over a given period of the collected source IP addresses (xi) and destination IP addresses (yj) yields the Renyi Joint Entropy. In Equations 1 and 2, you may calculate the probability of xi and yj.

Renyi Joint Entropy value is the most crucial parameter for detecting anomalous driving behavior at present. The data we gathered in this stage will help us move closer to our first and second study objectives. Next, we will use the Renyi joint entropy to set a dynamic threshold.

Dynamic Threshold

Compared to static threshold-based methods, which are typically used for DDoS attack detection, the proposed method significantly improves detection accuracy while concurrently decreasing the false-positive and false-negative rates. The single threshold value used by these methods also makes it impossible to differentiate between benign and malicious transmissions. In the case of DDoS attacks, the difference in traffic volume between a low and high attack may not be significant enough to warrant the adoption of a higher threshold in attack detection systems [21].

As a result, a fixed threshold cannot be relied upon to reduce the false-positive rate. Because it relies on observation and experimentation, finding the threshold value is a time-consuming process. In contrast, a dynamic threshold's flexibility and performance improvements make it the preferred method for detecting DDoS attack traffic at both low and high rates. In order to determine the acceptable range of incoming network traffic, this stage is being conducted. The dynamic threshold is calculated based on the following assumptions:

- The amount of information currently being sent to the controller from the network.
- The rate at which attack traffic will, in a certain amount of time, get close to the Renyi joint entropy value of the target(s).

Using Equation 4, we can determine the degree of randomness in the traffic flow of these packets (i.e., their source and destination IP addresses) over a given time interval t.

$$\boldsymbol{PF} = \boldsymbol{H_{RJ\alpha}(x, y)} \qquad (4)$$

The probability of a packet successfully arriving at its destination is denoted by PF, and $\boldsymbol{H_{RJ\alpha}(x, y)}$is the randomness (Renyi Joint Entropy) of the packets in the traffic flow at time t.

Variations in attack activity as well as true packet behavior have the potential to influence the estimation of the threshold. After gathering and analyzing pertinent data on the incoming traffic flow, as well as computing the Renyi Joint Entropy, the dynamic threshold will be established by applying the exponentially weighted moving average study technique.

To find a dynamic threshold, this study makes use of the study approach. As a consequence of this, the Renyi joint entropy value and the statistics of network traffic flows collected by the SDN controller form the basis of the dynamic threshold in this work. As a result of this, the amount of time required to calculate a suitable threshold for the proposed approach is reduced, which ultimately results in an improvement in the DDoS attack detection effectiveness. Equation (1) displays the fundamental study formula (5).

$$Study\ formula_i = (1 - \alpha) \cdot Study\ formula_{i-1} + \alpha \cdot PF_i \qquad (5)$$

The threshold that was selected for this investigation has a starting point value of 1.31, where study formula i is the value that is being measured by the study formula and study formula (i-1) is the value that was projected to be the approach before the research was carried out. The present value is PFG, as calculated by Mousavi [18]; the present value is a Renyi joint entropy value over a finite period (a single window size), which acts as a buffer against PF noise and as a steadying influence on EMWA. The present value is a Renyi joint entropy value over a single window size; $i \geq 1, 0 < \alpha < 1$.

When looking to improve the mechanism that detects DDoS attacks in network traffic flows, it is recommended that a dynamic threshold value be used rather than a static one. It does this by comparing the Renyi Joint Entropy value to a dynamic threshold and sending an alert to administrators if the value is lower than the threshold. This allows it to detect traffic from DDoS attacks. In every other regard, the activity on the network is typical.

It is anticipated that the proposed method will benefit from the utilization of the generalized method of Renyi Joint Entropy in conjunction with the dynamic threshold for both low- and high-rate cyberattacks against SDN controllers. These cyberattacks can be triggered by either a single host attacking a single victim or by multiple hosts attempting to attack multiple victims. In that case, the second purpose of our research will have been accomplished successfully.

These procedures, when applied to an SDN network, protect the controller from DDoS attacks by utilizing a statistical method to forecast the behavior of network traffic. The SDN network itself is used to implement these processes. Our research generalizes Renyi's joint entropy, modifies its threshold to account for variations in network traffic, and uses rule-based detection to identify both moderate and severe distributed denial of service (DDoS) assaults. These modifications were made possible by the fact that Renyi's joint entropy could be generalized. The entire process of the suggested method is shown in its entirety from beginning to end in Figure 3.

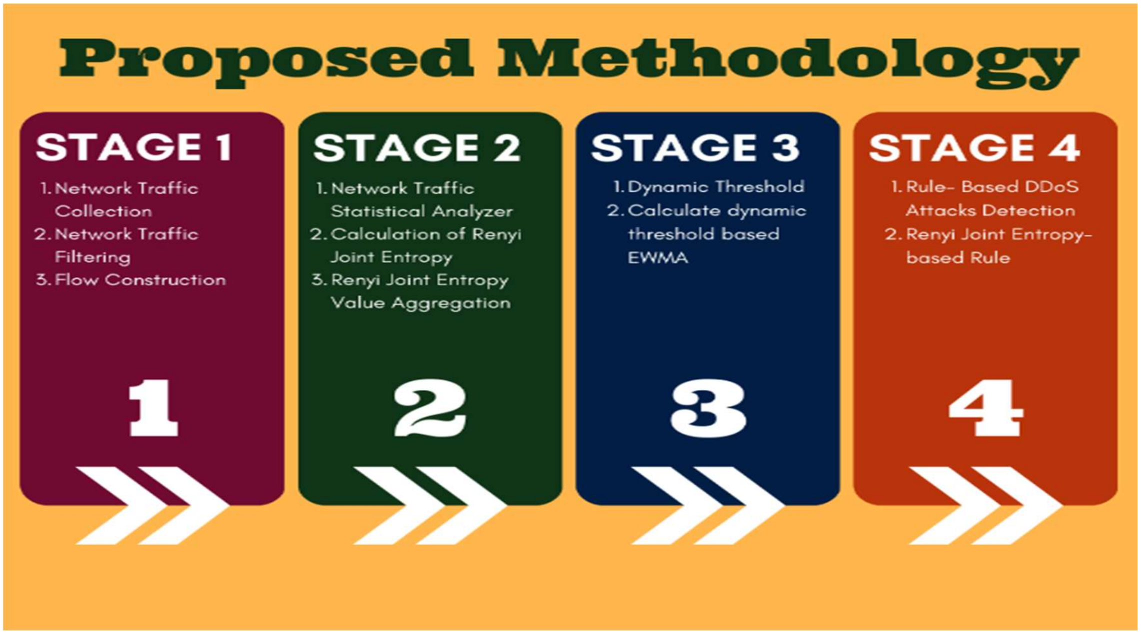

Proposed Methodology
STAGE 1
1. Network Traffic Collection
2. Network Traffic Filtering
3. Flow Construction
1
STAGE 2
1. Network Traffic Statistical Analyzer
2. Calculation of Renyi Joint Entropy
3. Renyi Joint Entropy Value Aggregation
2
STAGE 3
1. Dynamic Threshold
2. Calculate dynamic threshold based EWMA
3
STAGE 4
1. Rule- Based DDoS Attacks Detection
2. Renyi Joint Entropy-based Rule
4

Figure 3 : Proposed methodology.

The approach that is going to be described here was devised expressly with the objectives of this research in mind. The approach can be broken down into a total of four distinct stages. To begin, while the phase of processing is in progress, packets are recorded with the purpose of filtering out all incoming traffic. It does this by combining individual packets into flows, each of which has its own set of characteristics, and then only sending a subset of these flows to the controller. The second phase, which utilizes the Generalized Renyi Joint Entropy Equation to identify both low-rate and high-rate DDoS attacks on the SDN controller, is the phase that is the most essential. As a consequence of this, this phase will perform the role of the major cautionary signal for drivers who exhibit odd driving habits. In the third stage, you will need to determine the dynamic threshold to account for the variable rates of network traffic that are headed toward the controller. In the final phase, the Renyi Joint Entropy value is compared to the dynamic threshold based on a set of rules to determine the presence of DoS attacks when they take place. The procedures that make up the suggested strategy are outlined in the next chapter in their entirety.

3. Experiment Result

As can be seen in Table 2, the assault packet is being sent out at a rate of 0.03% per second, which is equivalent to 166 packets being sent out each second. A Distributed Denial of Service attack is considered a high rate by Kia anytime packets are sent at a rate of 0.03 per second or more (2015). It is also essential to highlight the fact that this pace is comparable to the normal levels of traffic. As a result of the fact that the testbed contains 61 "normal" hosts in addition to 3 "attacker" hosts, 61 packets are transmitted per second. That speed is sufficient to deliver 500 assault packets in fewer than five seconds. During this same period, 305 normal packets have been transmitted in the last 5 seconds. However, only 61 of those packets make it to the controller since the router will only forward those that have a unique IP source address. In just five seconds, this equates to a 19% increase in the ratio of assaults.

3.1 Dataset

The lack of availability of benchmark datasets necessitated the creation of synthetic datasets within the scope of this research. All the traffic that occurs within the SDN topology is logged and incorporated into the suggested strategy as input. The datasets that have been generated include both valid and illegitimate traffic on the network. There are a total of eight different datasets produced, and each of these datasets represents a different assault scenario.

Table 2: Dataset Summarization.

Dataset name	Normal Traffic Number	Attack Traffic Number	Attack traffic ratio
Dataset 1	63	5	7%
Dataset 2	63	33	34%
Dataset 3	63	2	7%
Dataset 4	63	11	34%
Dataset 5	61	15	19%
Dataset 6	61	166	62%
Dataset 7	61	15	19%
Dataset 8	61	166	62%

Table 2 summarizes the total number of packets in both the network's regular and attack traffic flows. Further, at the given time interval t, the proportion of malicious traffic to overall network traffic is displayed. The following formula can be used to determine the ratio of attack traffic (Sahoo et al., 2018):

$$ATTack\ Traffic\ Ratio = \frac{Attack\ Packet}{packet\ total} \times 100\% \quad (6)$$

Attacks on a Single Host

The suggested method is put to the test in a single-host attack scenario, where it must consistently detect low- and high-rate DDoS attacks launched from a single host at one or more victim hosts while maintaining a low false positive rate. You need to have a plan of defense in place for the following four situations: Here are four examples of assault types: This host is vulnerable to four distinct forms of attack: There are four types of attacks: (i) those that affect just one victim host (SSL); (ii) those that affect many victims (SSH); (iii) those that originate from several hosts (SML); and (iv) those that affect many victims but originate from just one (SML) (SMH). For a variety of applications, the relative amount of attack traffic to total network traffic is summarized in Table 3.

Table 3 : The Traits Of Attacks On A Single Host.

Scenarios /5minutes	Total Number of Normal Traffic	Total Number of Attack Traffic	Attack Percentage
SSL	18900	1500	7%
SML	18900	1500	7%
SSH	18900	9900	34%
SMH	18900	9900	34%

Each 5-minute (300-second) frame displays the packet distribution for a more accurate depiction of the data. This means that during a period of five seconds, the suggested method will provide data on the sum of sixty separate traffic flows. Every five minutes, data will be given on the average detection rate and false positive rate across sixty traffic flows. Table 3 shows that in all single-host assault scenarios, the average number of normal packets transmitted in 5 minutes is 18900.

Meanwhile, the rate of attack packets is typically proportionate to the volume of traffic, whether that be low or high. Table 1 shows that even a low-powered DDoS assault can unleash as many as 1500 packets in 5 minutes.

High-velocity distributed denial-of-service attacks typically involve 9,900 packets. It is important to note that both low attack traffic (which accounts for 7%) and high attack traffic (which accounts for 34%) contribute to the overall attack volume. Below, we detail the method's efficacy against single-host attacks.

Low-Frequency Attack against a Single Host (SSL)

In the first scenario, controllers that are the targets of low-rate DDoS assaults from a single server assaulting a single victim will be put to the test. To identify low-rate DDoS attacks, this technique uses Renyi Joint Entropy values and a configurable threshold. Figure 4 displays the average Renyi joint entropy and dynamic threshold over a 5-minute period.

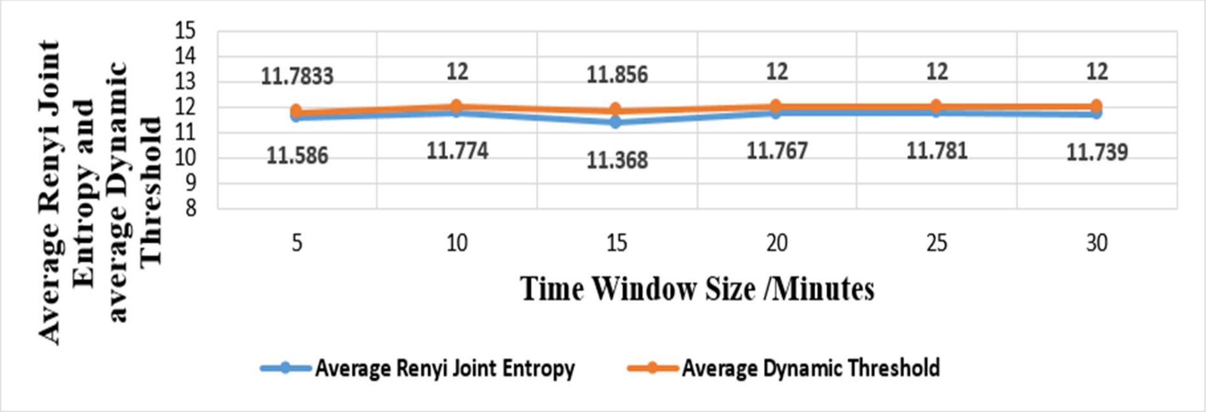
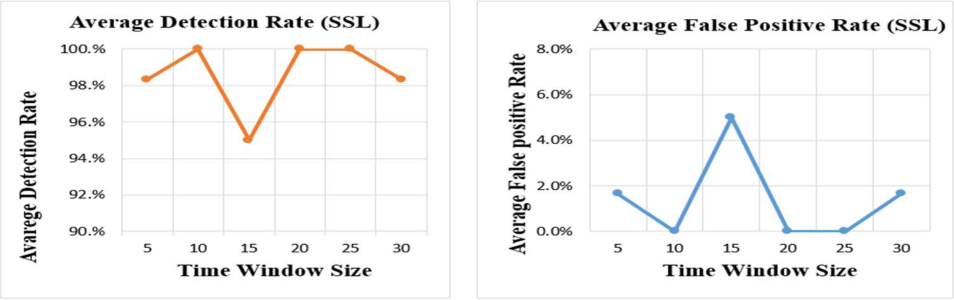


Figure 4 : Renyi Joint Entropy and Dynamic Threshold Mean Values in Scenario 1 (SSL).

When Renyi Joint Entropy values are slightly below the dynamic threshold values, the Renyi Joint Entropy-based rule states that there are low-rate DDoS attacks in the network. This occurs 30 minutes after the experiment has concluded.

The approach's detection rate for slow DDoS attacks is shown in Figure 5a and its false-positive rate is shown in Figure 5B. The DDoS assault packets in this scenario have several spoofed source IP addresses, yet they all originate from the same compromised server. No more than 7 percent of all traffic can be malicious. Packets are delivered to the controller if there is no matching IP address in the flow table. Furthermore, even if the controller is under attack, the approach can still gather the future sent packets for processing.



(A) (B)

Figure.5 : Normalized Rate of Detection for Scenario 1 (SSL) Rate of False Positives in Scenario 1 on Average (SSL)

The data demonstrate that the suggested approach can detect a low-rate DDoS attack on the controller with a 95% to 100% success rate. Contrarily, the percentage of false positives might be anything from 5% to 0%.

According to Table 4, the average detection rate for Scenario 1 within a 5-minute timeframe is 85%, with a false positive rate of 5%.

Table 4: Methodological evaluation of a study According to the first possible outcome

SSL/Minutes	Average Detection Rate (5-minute time window)	Average False Positive Rate (5-minute time window)
5	98.33%	1.66%
10	100.00%	0.00%
15	95%	5%
20	100.00%	0.00%
25	100.00%	0.00%
30	98.33%	1.66%
Total Average	98.61%	1.39%

Even though the DDoS attack traffic rate is relatively modest (0.2/sec) and similar to normal traffic, the suggested technique demonstrates a high detection rate (average 98.61%) and low false positive rate (average 1.39%). Unlike other methods, Renyi Joint Entropy makes advantage of not just one but two features of packet headers. It also employs a variable threshold that adapts to both attack and network activity levels.

A High-Rate Attack Against a Single Host (SSH)

The second scenario involves simulating a high-volume distributed denial-of-service (DDoS) attack on the controller using a single victim host to assess the true-positive and false-positive rates of our proposed method. Renyi joint entropy values and a movable threshold play a significant role in this technique for identifying high-velocity DDoS assault traffic. Five-minute averages of the Renyi Joint Entropy and the dynamic threshold are displayed in Figure 6.

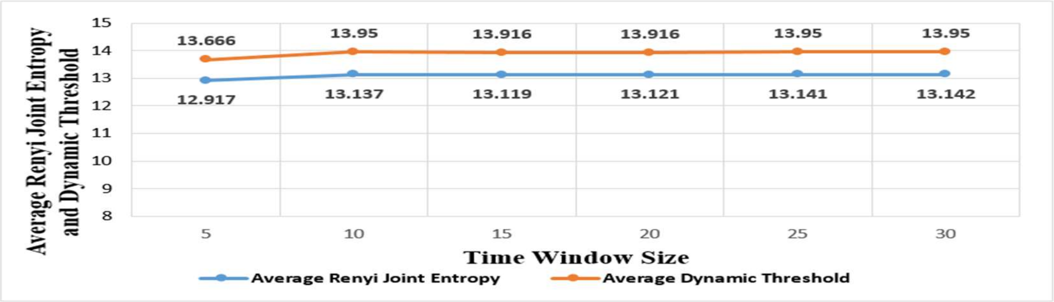
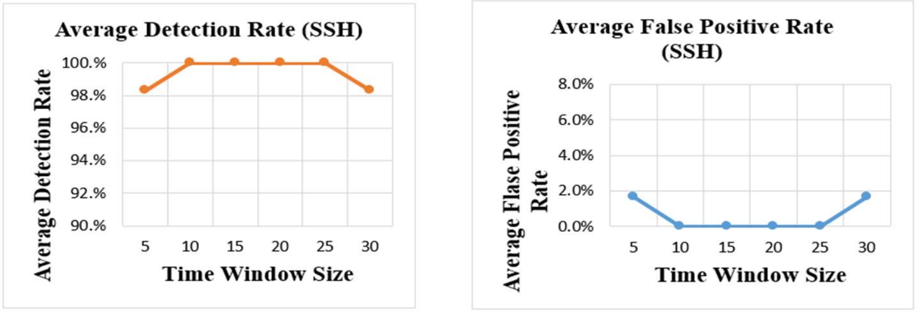


Figure .6 Calculating the Mean Renyi Joint Entropy and the Dynamic Threshold for Scenario 2 (SSH).

It can be seen in Figure 6 that after only 30 minutes of the experiment, the average Renyi Joint Entropy values have dropped dramatically below the dynamic threshold values, indicating that the victim of the DDoS attack has been severely degraded in terms of security.

Figure 7a displays the detection rate of the method, while Figure 7b displays the false positive rate, both of which are important for detecting high-rate DDoS attacks. An attack in which a single host takes on the role of an aggressor, forging attack packets with faked source IP addresses, and then conducting a distributed denial-of-service assault against the controller. The packets will be sent to the controller because there is no IP address information in the flow table. Even if the controller is under assault, the method can nevertheless gather the future transmitted packets for processing.



(A) (B)

Figure . 7 The Typical SSL Detection Rate in Case 2 (SSH) Rate of Misdiagnosis in Scenario 2 (SSH).

Our calculations show that the proposed approach can detect the highly targeted DDoS attack on the controller with a probability of between 97.33% and 100%. From a near-zero (0.0%) to a near-perfect (1.6%) rate of false positives. According to Table 5, after 30 minutes, the average detection rate is 99.44% with a false positive rate of 0.55%.

Table 5: The ramifications of the assessment of the study. Per the alternative scenario proposed.

SSH/Minutes	**Average Detection Rate** (5-minute time window)	**Average False Positive Rate** (5-minute time window)
5	98.33%	1.66%
10	100.00%	0.00%
15	100.00%	0.00%
20	100.00%	0.00%
25	100.00%	0.00%
30	98.33%	1.66%
Total Average	99.44%	0.55%

Table 5 demonstrates that with the given Renyi Joint Entropy and dynamic threshold, the suggested method has a high detection rate (average 99.44%) and a low false positive rate (average 0.55%) for identifying DDoS assaults.

Single-Host, Low-Rate Attack against Many Intended Targets (SML)

For the third part, we use a single host attack to test how well our method can detect a DDoS attack directed at a controller. This method can detect low-rate DDoS attacks by employing Renyi Joint Entropy attacks and dynamic threshold attacks. On display in Figure 8 are the mean values of the Renyi joint entropy and the dynamic threshold throughout the course of each 5-minute time period.

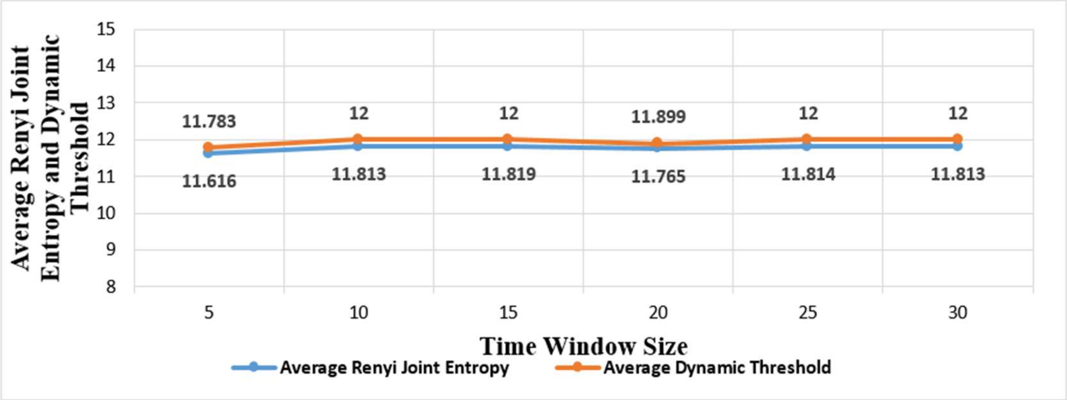
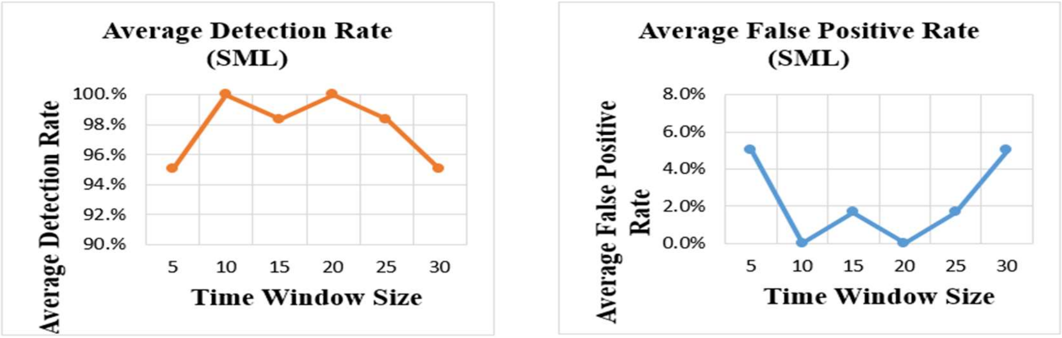


Figure 8 : Statistics for Scenario 3's Mean Renyi Joint Entropy and Dynamic Threshold (SML).

The accompanying graph shows that within the first 30 minutes of an experiment, Renyi Joint Entropy values are extremely near to dynamic threshold levels. During a distributed denial of service (DDoS) attack on a single host, the amount of traffic experienced by any individual victim would be proportionally lessened. A low-rate DDoS attack on a single host, on the other hand, is easier to identify due to the increased attack traffic at that site. The proposed method was successful in detecting and differentiating attack traffic associated with a DDoS attack, even if the attack traffic rate was low and the targets were dispersed.

Figure 9a displays the detection rate achieved by the approach, whereas Figure 9b displays the false-positive rate. As a result of a single hacked server, several victims are hit by a distributed denial of service (DDoS) attack. Without an IP address in the flow table, the packets will be sent to the controller. This approach can still collect and handle future packets even if the controller is under assault.



(A) (B)

Figure 9: Normalized Scenario 3 SSL Detection Rate (SML) Rate of Misdiagnosis in the Third Scenario (SML).

The results demonstrate that the suggested approach has a detection rate of 95%-100% for DDoS attacks, meaning it can detect even the low-rate DDoS assault against the controller that hits several targets. However, false-positive rates are always around 5%. Since the low-rate attack traffic was split amongst multiple victims, the detection rate and false positive rate are both higher than in the first scenario. Table 6 below summarizes the average detection rate and false-positive rate while trying to detect low-rate DDoS assaults.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6: Method Evaluation Using the SML Targeted Attack on a Single Host.

SML/Minutes	Average Detection Rate (5-minute time window)	Average False Positive Rate (5-minute time window)
5	95.00%	5.00%
10	100.00%	0.00%
15	98.33%	1.66%
20	100.00%	0.00%
25	98.33%	1.66%
30	95.00%	5.00%
Total Average	97.78%	2.22%

The presented method has a high detection rate and a low false positive rate even when the DDoS attack traffic rate is low (0.2/sec) and similar to regular traffic. Combining Renyi Joint Entropy, which examines two aspects of packet headers rather than one, with a dynamic threshold that varies dependent on the volume of attack traffic results in a high detection rate.

High-velocity, a single-host attack aimed at a large number of hosts (SMH)

In the fourth scenario, we want to see how well our suggested method can spot a high-rate distributed denial of service (DDoS) attack against the controller, in which a single host is responsible for attacking many other hosts. This method can detect high-velocity DDoS attack traffic by using the Renyi joint entropy and a dynamic threshold. Figure 10 depicts the average Renyi joint entropy and dynamic threshold values over a 5-minute time interval.

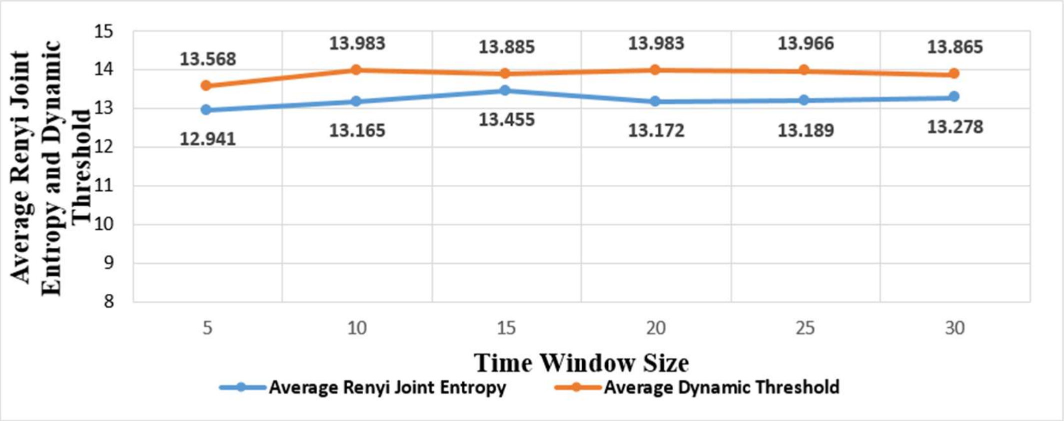
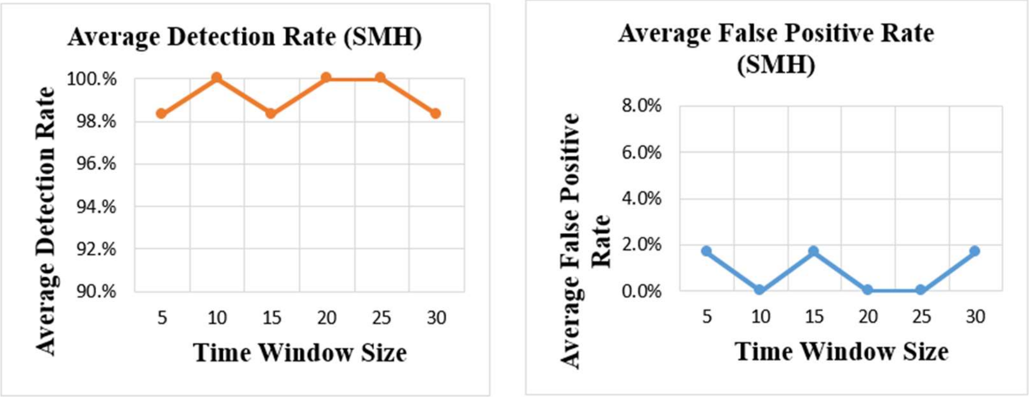


Figure 10 : Renyi Joint Entropy and Dynamic Threshold Scenario 4 Means (SMH).

As can be seen in Figure 10, the Renyi Joint Entropy values are far lower than the dynamic threshold values after the first 30 minutes of the experiment, but they can approach the latter as the attack traffic is spread among numerous victims rather than a single victim host. So, if you suspect a low-rate DDoS attack on your network, you can check for it using the Renyi Joint Entropy-based Rule.

How effective is this method in detecting and dismissing potential DDoS assaults, and how often does it incorrectly identify true DDoS attacks on numerous servers? With an attack traffic ratio of up to 34%, a single server can conduct a DDoS attack on the controller. An attacker can launch an attack using a large number of spoofed IP addresses from only a single compromised workstation. In the absence of a matching IP address in the flow table, data is forwarded to the controller. Further, the method can continue to collect the subsequent transmitted packets for processing even if the controller is under attack.



(A) (B)

Figure 11 : Estimated False Positive Rate for Average SSL Detection in Scenario 4 (SMH).

As can be seen in the statistics above, the proposed technique has a detection rate of between 98.33% and 100% against the high-rate DDoS assault against the controller. Contrarily, estimates for the percentage of false positives range from zero to 1.6 percent. Table 7 displays the suggested method's average detection rate and false positive rate for all 5-minute time intervals.

Table 7: Measurement of Method Using Attack Scenario 4 for a Single Host

SMH/Minutes	Average Detection Rate (5-minute time window)	Average False Positive Rate (5-minute time window)
5	98.33%	1.66%
10	100.00%	0.00%
15	98.33%	1.66%
20	100.00%	0.00%
25	100.00%	0.00%
30	98.33%	1.66%
Total Average	99.17%	1%

Despite the high traffic rate of the DDoS attacks (0.03/sec) and their targeting of several victims, the proposed approach can detect the assaults with a high detection rate (average 99.17%) and a low false positive rate (average 1%). The suggested method's impressive effectiveness can be attributed in large part to Renyi Joint Entropy, which uses two attributes of packet headers rather than one and dynamic thresholds that adjust to attack rates. Here, we take a look at how well the suggested approach can spot low- and high-rate DDoS attacks on the SDN controller, both against a single victim and across several targets. We've tried out the method in eight different simulated environments, each with a different amount of attack traffic.

The proposed method is compared to the competing technique using simulated case-by-case comparisons of its detection rate and false-positive rate medians. Furthermore, the results of the comparisons demonstrated that the proposed method outperformed the existing method in terms of detection accuracy and false positive rate across all the simulation scenarios. In addressing the persistent challenge of DDoS attacks in SDN environments, this study focused on improving detection accuracy and reducing false positives. The chosen evaluation criteria, detection rate and false-positive rate, are pivotal in assessing the effectiveness of DDoS detection methods, directly impacting network security and resource management. Compared to traditional approaches, our method demonstrated a notable improvement in detection accuracy while maintaining a lower false-positive rate, thereby directly addressing the critical needs identified in the research problem.

Our criteria for analysis, while grounded in established metrics, introduced a novel combination of dynamic thresholding and generalized entropy. This approach differed from previous studies that relied more on static parameters, offering a more adaptable and responsive solution to varying attack scenarios. The outcomes of this study not only align with known facts about DDoS detection challenges but also extend the understanding by demonstrating the viability of a more dynamic approach. By juxtaposing our findings with existing knowledge, it becomes evident that while some aspects of DDoS detection remain consistent, the introduction of more adaptable methods like ours can significantly enhance detection capabilities in SDN environments. This study, therefore, not only addresses the immediate research problem but also contributes to the broader discourse on network security..

4. CONCLUSION

This study aimed to tackle the significant issue of Distributed Denial-of-Service (DDoS) assaults in Software-Defined Networking (SDN) systems. Our research intended to improve the accuracy of detecting DDoS attacks and limit the occurrence of false positives by utilizing innovative techniques such as dynamic threshold and extended entropy methods.

The research employed various approaches to evaluate the suggested approach in eight distinct simulated scenarios, each characterized by different levels of attack flow. This methodology facilitated a thorough evaluation across several scenarios, guaranteeing the strength and dependability of the findings. Our strategy outperformed existing strategies in terms of detection accuracy and achieved lower false-positive rates in all simulation settings. The results of this study make a substantial contribution to the subject of network security in the context of Software-Defined Networking (SDN). The authors showcase the effectiveness of utilizing dynamic thresholding in conjunction with generalized entropy for the purpose of identifying DDoS attacks, which are a continuous and ever-changing menace in the realm of network security. This development not only helps protect SDN controllers from potential overload and exhaustion, but also assures a more secure and efficient network management system. These discoveries have the potential to guide future study in various important domains. Continued investigation into enhancing the detecting algorithms to achieve higher levels of accuracy and efficiency is still a top focus. Furthermore, it is imperative to expand this research to real-world settings and larger networks in order to authenticate the suitability and efficacy of the suggested approach in more diverse and intricate scenarios.

To summarize, the research effectively tackles a major obstacle in SDN security, providing a hopeful new approach for countering DDoS attacks. The potential for additional progress in this field is immense, with ramifications that might greatly improve the security and efficiency of contemporary network infrastructures.

Acknowledgments To everyone who helped and contributed to the completion of this research paper, we would like to convey our sincere gratitude. They have been incredibly helpful and supportive in making this study possible.

Conflict of interest The author declares that he has no conflict of interest.

Funding The author(s) received no specific funding for this work.

Data Availability The data used in this study are available upon request from the corresponding author. The data will be made available in a secure and confidential manner, consistent with applicable laws and regulations. Any requests for data will be reviewed on a case-by-case basis to ensure that the data are being used for legitimate research purposes.

REFERENCES

[1] O. Abu Alghanam et al., "A new parallel matrix multiplication algorithm on tree-hypercube network using iman1 supercomputer," International Journal of Advanced Computer Science and Applications, vol. 8, no. 12, 2017.

[2] M. A. Al-Adaileh et al., "Proposed statistical-based approach for detecting distribute denial of service against the controller of software defined network (SADDCS)," MATEC Web of Conferences, vol. 218, 2018.

[3] R. R. Althar and D. Samanta, "The realist approach for evaluation of computational

intelligence in software engineering," Innovations in Systems and Software Engineering, vol. 17, no. 1, pp. 17-27.

[4] J. Chen, X. Zheng, and C. Rong, "Survey on software-defined networking," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9106, no. 1, pp. 115–124, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-28430-9_9

[5] Cisco, "Cisco Cloud Index: Data Center SDN to Skyrocket by 2021," 2018. [Online]. Available: https://www.sdxcentral.com/articles/news/cisco-cloud-index-data-center-sdn-skyrocket-2021/2018/02

[6] A. Cletus, B. Weyory, and A. Opoku, "Improving Social Engineering Awareness, Training and Education (SEATE) using a Behavioral Change Model," International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, 2022.

[7] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks," ACM Sigcomm Computer Communication, vol. 44, no. 2, pp. 87–98, 2014. [Online]. Available: https://doi.org/10.1145/2602204.2602219

[8] B. Görkemli, A. Parlakışık, and S. Civanlar, "Dynamic management of control plane performance in software-defined networks," in 2016 IEEE NetSoft, pp. 68–72, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7502445

[9] D. He, S. Chan, X. Ni, and M. Guizani, "Software-Defined-Networking-Enabled Traffic Anomaly Detection and Mitigation," IEEE Internet of Things Journal, 4662(c), pp. 1–1, 2017. [Online]. Available: https://doi.org/10.1109/JIOT.2017.2694702

[10] V. T. Hoang et al., "A comparative study of rice variety classification based on deep learning and hand-crafted features," ECTI Transactions on Computer and Information Technology (ECTI-CIT), vol. 14, no. 1, pp. 1-10, 2020.

[11] M. Kia, "Early Detection and Mitigation of DDoS Attacks In Software Defined Networks," Master's Thesis, Ryerson University, Toronto, ON, Canada, 2015.

[12] D. Kreutz and F. Ramos, "Software-Defined Networking: A Comprehensive Survey," ArXiv Preprint ArXiv: …, no. 49, 2014. [Online]. Available: https://doi.org/10.1109/JPROC.2014.2371999

[13] J. Liu, Y. Lai, and S. Zhang, "FL-GUARD: A Detection and Defense System for DDoS Attack in SDN," in Iccsp, pp. 107–111, 2017. [Online]. Available: https://doi.org/10.1145/3058060.3058074

[14] B. A. Mahafzah, A. A. Al-Adwan, and R. I. Zaghloul, "Topological properties assessment of optoelectronic architectures," Telecommunication Systems, pp. 1-29, 2022.

[15] R. Masoudi and A. Ghaffari, "Software defined networks: A survey," Journal of Network and Computer Applications, vol. 67, pp. 1–25, 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2016.03.016

[16] M. O. Mirabdullayevna, "PRINCIPLES FOR MODELLING THE SPATIAL AND TECHNOLOGICAL STRUCTURE OF FLOW CONSTRUCTION PROCESSES," Yosh Tadqiqotchi Jurnali, vol. 1, no. 1, pp. 239-244.

[17] S. Mousavi, "Early Detection of DDoS Attacks in Software Defined Networks Controller2014 ",. [Online]. Available: https://curve.carleton.ca/system/files/theses/31561.pdf

[18] M. Nooribakhsh and M. Mollamotalebi, "A review on statistical approaches for anomaly detection in DDoS attacks," Information Security Journal: A Global Perspective, vol. 29, no. 3, pp. 118-133, 2020.

[19] R. Oliveira et al., "A Scalable, Real-Time Packet Capturing Solution," in International Conference on Optimization, Learning Algorithms and Applications, pp. 630-637, Springer, Cham, July 2021.

[20] H. Peng et al., "Dynamic threshold neural P systems," Knowledge-Based Systems, vol. 163, pp. 875-884, 2019.

[21] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-AI integration perspective," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1328-1347, 2019.

[22] Y. P. Sahana, A. Gotkhindikar, and S. K. Tiwari, "Survey on CAN-Bus Packet Filtering Firewall," in 2022 International Conference on Edge Computing and Applications (ICECAA), pp. 472-478, IEEE, Oct. 2022.

[23] K. S. Sahoo et al., "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," Future Generation Computer Systems, vol. 89,

pp. 685–697, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2018.07.017

[24] S. Scott-Hayward, S. Natarajan, and S. Sezer, "Survey of Security in Software Defined Networks," Surveys & Tutorials, vol. 18, no. 1, pp. 623–654, 2016. [Online]. Available: https://doi.org/10.1109/COMST.2015.2474118

[25] Z. Shu et al., "Traffic engineering in software-defined networking: Measurement and management," IEEE Access, vol. 4, pp. 3246–3256, 2016. [Online]. Available: https://doi.org/10.1109/ACCESS.2016.2582748

[26] Statista, "Software-defined networking (SDN) market revenue worldwide from 2016-2022 (in billion U.S. dollars) Statistic," 2016. [Online]. Available: https://www.statista.com/statistics/668394/worldwide-software-defined-networking-market-revenue

[27] S. Takeno et al., "A generalized framework of multifidelity max-value entropy search through joint entropy," Neural Computation, vol. 34, no. 10, pp. 2145-2203, 2022.

[28] U. Tariq, M. Hong, and K. S. Lhee, "A comprehensive categorization of DDoS attack and DDoS defense techniques," in International Conference on Advanced Data Mining and Applications, pp. 1025-1036, Springer, Berlin, Heidelberg, Aug. 2006.

[29] T. Wang et al., "Privacy-enhanced data collection based on deep learning for internet of vehicles," IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6663-6672, 2019.

[30] S. Yoshida et al., "FPGA-based network microburst analysis system with efficient packet capturing," Journal of Optical Communications and Networking, vol. 13, no. 10, pp. E72-E80, 2021.

[31] H. Zhang et al., "A Survey on Security-Aware Measurement in SDN," Security and Communication Networks, 2018. [Online]. Available: https://doi.org/10.1155/2018/2459154

[32] M. A. Aladaileh et al., "Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates," Applied Sciences, vol. 12, no. 12, p. 6127, Jun. 2022. [Online]. Available: https://doi.org/10.3390/app12126127.