ANALYSIS OF DDOS ATTACK DETECTION TECHNIQUES FOR SECURING SOFTWARE-DEFINED NETWORKS

Danijel Čabarkapa, MSc[1]
Academy of Professional Studies Šabac, Serbia

Brankica Popović, PhD
University of Criminal Investigation and Police Studies, Belgrade, Serbia

Petar Čisar, PhD
University of Criminal Investigation and Police Studies, Belgrade, Serbia

Kristijan Kuk, PhD
University of Criminal Investigation and Police Studies, Belgrade, Serbia

INTRODUCTION

Some serious problems arise in the traditional TCP/IP networks, such as the limitations of standardized equipment that runs proprietary software, the difficulty of deploying and managing, the complexity of congestion control, and the large number of applications that create network bottlenecks. Today, network systems are becoming more complex and feature-rich, and network designers often need to modify network software to achieve their requirements (Dudeja, R. K. et al., 2022). The Software-Defined Networking (SDN) paradigm breaks vertical integration by radically separating the packet forwarding and the control plane, providing applications with a centralized and abstract view of network distribution. SDN attempts to move as much network functionality as possible into user-definable software, making more of the network system components programmable. Network virtualization is one of the key features facilitated by the SDN, and it allows multiple virtual networks and the SDN controllers to share the same physical network infrastructure (Villota et al., 2018).

1 d.cabarkapa@gmail.com

However, with the popularity of SDN, their security has become one of the key research subjects. The recent changes in the cyber threat scope indicate the increased activities of cybercriminal communities mostly focusing on malware, Web-based attacks, DDoS attacks, and various social engineering attacks. Today malware, ransomware, DDoS attacks, and phishing are the most important security threats particularly dangerous in SDN due to their strong destructiveness, simple implementation, and lack of simple and feasible countermeasures (Dong S. et al., 2019). Considering the SDN network programmability and automation, the question of how to develop more efficient defense solutions against DDoS in SDN has attracted intense research in recent years. There is a fact that there are different types of DDoS attacks on SDN and therefore any effort to secure those networks requires a comprehensive understanding of SDN architecture and recent technological advances used to address security issues.

From the perspective of the SDN which is a flow-based network model, we can classify DDoS attacks into two major types: attacks based on the volume of packets, and attacks based on the number of flows. Novel DDoS detection techniques are mostly flow-based, and with an aid of specific approaches can provide faster and more accurate results. Entropy-based network traffic anomaly detection techniques are attractive due to their simplicity and applicability in a real-time network environment. The main issue of the entropy approach is the fine-grained traffic analysis, accuracy of traffic variation detection, and the choice of the features that would provide accurate detection (Ibrahim J. et al., 2022). Machine learning (ML) algorithms can automatically build classification models based on training data, and classify traffic based on the features of flows. The authors' contribution in this paper involves presenting the problem and making an overview of protection against DDoS detection in SDN networks that encompasses techniques for entropy-based data processing and ML attack detection. We have extended the entropy-based attack detection approach with the anomaly classification method to ensure that the attack traffic can be identified quickly and effectively. A certain number of research papers show that the combination of the entropy approaches in the SDN traffic data processing and ML classification algorithms for attack detection are in line with the needs of the enterprise environments, which are specifically attractive for DDoS attacks.

The other part of this paper is organized as follows. Section 2 gives an overview of DDoS attack mechanisms and taxonomy. SDN layered architecture, virtualization, and DDoS security solutions for each of the three planes in SDN and they are discussed in Section 3. Section 4 addresses a brief introduction to the used entropy-based traffic analysis and DDoS attack detection and a discussion on the ML attack detection systems. In Section 5 we highlight some experimental works related to entropy and ML-based DDoS detection mechanisms. The conclusion of the paper is in Section 6.

DDOS ATTACKS OVERVIEW

DDoS attack aims at disrupting the availability of resources in the network. This task is achieved by a group of devices that are knowingly or unknowingly involved in the attack. Malicious user floods the network resources with a large amount of useless traffic to exhaust them as a result, malicious traffic gets served but legitimate packets starve for services because of packet overflow or congestion.

The operation of DDoS attacks follows several consecutive phases. The intruder initially starts to compromise multiple agent machines that are widely distributed geographically by scanning the vulnerabilities in these devices. Once an intruder successfully identifies certain system vulnerabilities, he can compromise these machines using a malicious program. By replicating the malicious file in multiple agents, the intruder can control many devices that can reach several thousands or millions (commonly referred to as bots) to initiate DDoS attacks without the awareness of the rightful owner of the device. The discovery of vulnerabilities and exploitation process of the agents are usually performed automatically, for instance, by sending e-mail messages with the attack code attachment. The groups of bots, known as a botnet can get orders remotely from an intruder, i.e. botmaster. The botmaster can perform large-scale DDoS attacks to flood a legitimate service or network by sending a control command to the botnet agents to generate useless traffic without getting noticed. Consequently, the victim resources become overwhelmed with a crushing volume of traffic in a short duration, which significantly slows down the system service or the ability of the network to respond to legitimate users (Gupta B. et al., 2009).

DDoS attacks could be broadly classified as volume-based attacks, protocol-based attacks, and application-based attacks (Zargar S. et al., 2013; Bonguet, A. et al., 2017). A taxonomy of some common types of DDoS attacks is presented in Fig. 1.

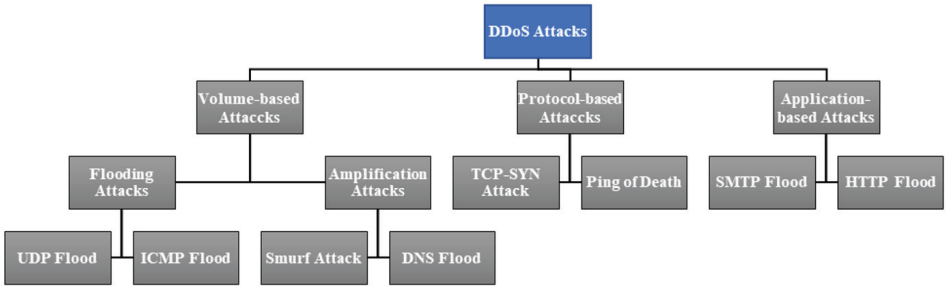

Figure 1 – *Taxonomy of DDoS attacks (adopted from [4])*

In a volume-based (volumetric) DDoS attack, the target is flooded with heavy traffic, aiming at exhausting its bandwidth. It results in congesting the bandwidth of attacked target. These attacks include flooding and amplification attacks (Ding

D. et al., 2021). There are three common types of volumetric flood attacks: User Datagram Protocol (UDP) flooding, Domain Name System (DNS) flooding, and Internet Control Message Protocol (ICMP) flooding. Amplification attacks exploit a disparity in bandwidth consumption between an attacker and the targeted network resource. Protocol-based DDoS attacks exhaust the resources of devices by exploiting the network protocols. These attacks do not rely on the volume of traffic but on the combination of traffic that could affect the application. TCP-SYN flood and Ping of Death are examples of such attacks. Application-based DDoS attacks aim at crashing the application or underlying device itself by exploiting application layer protocols. Such attacks include Hypertext Transfer Protocol (HTTP) flooding and Simple Mail Transfer Protocol (SMTP) flooding (Zhou L. et al., 2022).

TCP-SYN flood attack exploits the 3-way handshake protocol of TCP. The targeted host receives SYN messages from the attacker, opens a TCP connection with it and waits for acknowledgement (ACK) message, but it never gets the response as the attacker never replies or the request is sent from spoofed IP addresses. The host keeps on waiting for replies, resulting in DDoS to legitimate requests. HTTP flood attack does not require spoofed addresses or a high amount of data to be sent to attack a server. Simple HTTP requests GET and POST are sent requiring a huge amount of data in response consuming a large amount of bandwidth and taking down the server. These attacks are the most common DDoS attacks, as they are difficult to detect. In UDP flood attack, the attacker sends a large number of packets to random ports on the target and the targeted host constantly checks for applications on that port. As no listening application on that port is found, it replies with ICMP destination unreachable packet. This process consumes more resources, ultimately making the host unreachable. In Ping of Death, the attacker sends malicious packets to the target. In general, the maximum allowed packet size with a header is 65.535 bytes, and the Ethernet frame size is 1500 bytes. Attacker sends an ICMP echo-request (ping) with more than 65.535 bytes that may cause memory buffer overflow at the target host while reassembling the packet, resulting in DDoS to legitimate packets.

SOFTWARE-DEFINED NETWORKING ARCHITECTURE AND DDOS SECURITY MECHANISMS

The architecture of a SDN network can be divided into three planes: data plane, control plane, and application plane (Cabarkapa D. et al., 2022) considering a three-layer SDN architecture model, as we can see in Fig. 2. The control plane contains one or more logically centralized SDN controllers where the logic is centralized, as well as the global view of the network. Such a control plane manages the network, including applications in the application plane and the OpenFlow

switches in the data plane. Control functionality is removed from network devices, which will become simple packet forwarding network nodes. The application plane contains SDN applications that are intended to perform various functionalities: enforcing security mechanisms, performing network traffic management and virtualization, or running services on the SDN. SDN application plane consists of one Application Logic module and one or more NorthBound Interface (NBI) Drivers. The SDN is programmable through applications that interact with the underlying data plane devices. Higher-level logic can be implemented directly through these applications on top of controllers, which communicate through NBI Agents APIs (REST, JSON, etc.) (Zhou W. et al., 2014). The SDN Datapath comprises a SouthBound Interface (SBI) Agent and a set of one or more traffic forwarding engines and processing functions. The data plane is the combination of forwarding devices managed by the control plane through its SBI that implements the OpenFlow protocol.

OpenFlow is the most widely accepted and deployed SBI standard for SDN and represents a protocol that is used for the communication between the controller and forwarding devices. An OpenFlow protocol can handle high-level routing, packet forwarding, and secure connection between the control plane and data plane. The main component of a SDN network is the OpenFlow switch. The OpenFlow switch specification determines the components and basic functions of the SDN-enabled switch. OpenFlow switch consists of one or more flow tables. Flow tables determine data processing and forwarding with the help of flow entries. Each flow entry determines how data will be processed and forwarded in a network (Open Networking Foundation, 2015).

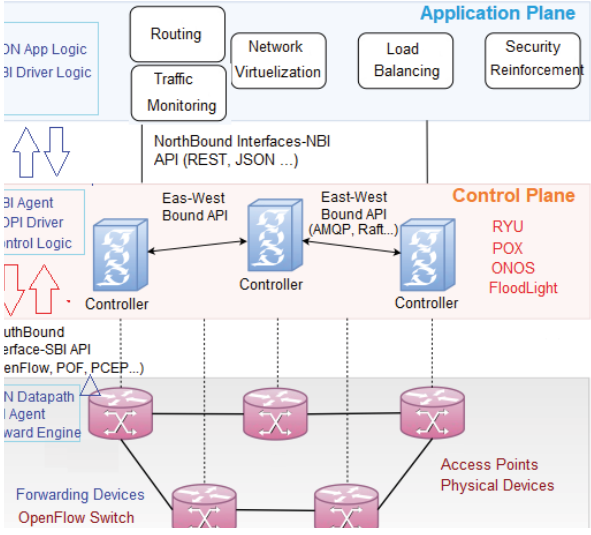

Figure 2 *Overview of a typical layered SDN architecture (adopted from [6])*

A fundamental characteristic of SDN is the logically centralized, but physically distributed controller component. The controller maintains a global network view of the underlying forwarding infrastructure and programs the forwarding entries based on the policies defined by network services running on top of it. The controller tracks the topology by learning the existence of OpenFlow switches and other SDN devices and by tracking the connectivity between them. All the controller functions are implemented via changeable modules, and the feature set of the controller may be adjusted to specific requirements of SDN networks. Currently, there is a variety of open-source SDN controllers available for the community: POX, RYU, FloodLight, ONOS, ODL, OpenDayLight, etc. (Berde P. et al., 2014; POX Github). To evaluate the controller performance in a detailed way, the paper (Cabarkapa D. et al., 2021) presented different performance aspects of the RYU and POX controller, such as throughput and latency, under simple tree-based and complex fat-tree-based network topologies. Work (Shalimov A. et al., 2013) presented a framework named HCprobe to compare seven different SDN controllers. To compare the effectiveness of these controllers, the authors performed additional measurements like scalability, reliability, and security along with latency and throughput.

The network virtualization (NV) process lies at the basis of SDN architecture. The SDN and the overlay concept were devised to adapt the network to global virtualization, as well as the necessary advanced technologies in software-defined data centers (Čisar P. et al., 2018). NV is one of the key features enabled by the SDN, and it allows multiple virtual networks and the SDN controllers to share the same physical network infrastructure. With the addition of NV techniques SDNs have gained a new dimension. This has allowed network slicing and multi-tenant hosting on existing physical network resources. FlowVisor (Sherwood, R. et al., 2009) is the most popular SDN-based implementation to utilize virtual networks by leveraging OpenFlow functionality to abstract the underlying hardware.

Security becomes more critical in the underlying SDN infrastructure and the rapid increase in the number of devices connected to the SDN networks not only increasing the data traffic but also raising concerns on security aspects of communications. SDN provides increased security features as the network control plane is detached from the forwarding plane and is programmed directly by the controller. Flow rules in the OpenFlow switches can be effectively modified for mitigation purposes. Due to SDN's programmable nature, whenever a malicious activity is detected in the network, required programs can be implemented for dealing with the malicious activities. However, this innovation also introduces various security challenges. Generally, DDoS attacks have become major threats to SDN networks. In such attacks, by exhausting resources, SDN application services are disabled, and the network performance is downgraded. Potential attacks can be executed in all three planes of

SDN architecture, and the DDoS attacks are divided into three categories: application-layer, control-layer, and data-layer attacks (Jimenez M. et al., 2021).

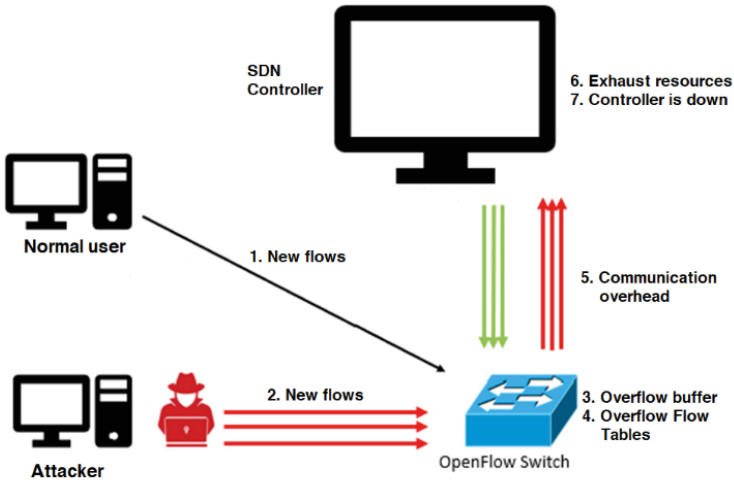

Figure 3 *Schematic view of DDoS attack in SDN (adopted from [16])*

DDoS attacks on the SDN controller are launched by sending a massive amount of network traffic with spoofed source IP addresses from different sources, as shown in Fig. 3 (1 and 2). These spoofed IP addresses do not match any existing flow rules in the flow table of the switch, resulting in a table miss case. Such a case results in generating massive packet-in messages sent to the SDN controller from the victim switch, which consumes communication bandwidth, memory, and CPU in both the control and the data plane of SDN. Since the victim switch buffers packet-in messages before sending them to the controller, if several new flows are received within a very short time, the buffer fills up (3). This results in higher consumption of the control plane bandwidth and delays the installation of new flow rules received from the controller. The forwarding table fills up, and therefore, upon receiving a new flow rule from the controller, it is unable to install it and hence dropping the packet (4). The switch would not be able to forward packets until there is free memory in its forwarding table, resulting in delays and dropping of incoming packets. On the controller side, a high arrival rate of packet-in messages exceeding the controller processing capability results in overwhelming the controller and making it unreachable to legitimate traffic (5, 6, and 7). This could fail the entire SDN network. Table 1 presents a few DDoS attacks possible on various SDN layers. Some of the DDoS attacks that are specific for the SDN networks are: buffer saturation attacks, flow table overflow, and resource exhausting (NBI interface, OpenFlow bandwidth, or TCAM memory of switches).

Table 1 – An overview of DDoS attacks on SDN planes

SDN Plane	Possible attacks
Application plane	NBI API exhaustion, Application layer DDoS (HTTP flooding, SMTP flooding)
Control plane	Resource depletion, OpenFlow bandwidth exhaustion, Amplification attacks
Data plane	TCAM exhaustion, Switch DDoS, Traditional DDoS (TCP-SYN flood, TCP flood, ICMP flood …)

ENTROPY AND MACHINE LEARNING-BASED DDOS ATTACKS DETECTION IN SDNS

Entropy is a degree of the uncertainty and randomness of a certain stochastic process. In network traffic analysis entropy can measure the randomness of packets entering the network. Entropy-based techniques rely on the traffic feature distribution and are categorized as (1) TCP header-based (including IP addresses, ports, or flags) (2) volume-based (including IP or port-specific percentage of flows, packets, and bytes), and (3) behavior-based (dealing with the degree of inbound and outbound communications). In anomaly detection techniques entropy is used to present the level of randomness in a data distribution. The changes in a data structure in a distribution obtained from the acquisition process will change the entropy value. If the entropy change is significant, it is considered to be unusual behavior in network communication or an anomaly, which often indicates security threats. The main issue of the entropy approach is the accuracy of traffic variation detection and the choice of the features that would provide accurate detection.

For proper functioning of the entropy calculation, the flow-based anomaly detection relies on the Shannon information entropy H_{IE} given in equation (2):

$$p_i = \frac{x_i}{\sum_{i=1}^{N} x_i} \quad 0 \leq p_i \leq 1 \tag{1}$$

$$H_{IE}(X) = -\sum_{i=1}^{N} p_i \log(p_i) \tag{2}$$

The variable X_i represents the destination IP address of the *i-th* packet, and the empirical probability p_i of X_i is calculated by using equation (1). The total number of packets in the window is denoted as N and $i = 1, 2, \ldots N$. A window is an interval for which entropy is to be calculated and consists of a certain number of incoming packets (window size) and a fixed time interval.

The entropy threshold value is determined based on the entropy fluctuation in normal traffic scenarios. When multiple incoming data packets are received on the same switch/host port in a window and the number of data packets exceeds the size of the window, DDoS attacks are detected. If the entropy value is higher than or equal to the threshold, the next calculation will be carried out normally and entropy calculation for new incoming packets is performed. If the entropy value falls below the threshold, the incoming packet is recorded. During an attack, if the computation entropy of a specified window continuously drops below the threshold, the target port on the specified switch is blocked. The main issue of the entropy approach is the accuracy of traffic variation detection and the choice of the features that would provide accurate detection. To better characterize entropy deviation, some research papers have also used normalized entropy values (Tsallis and Rényi) relative to the margin of tolerance, allowing entropy analysis more directly (Basicevic I. et al., 2021). Several traffic features (e.g., flow size, source/destination ports, IP addresses, etc.) have been suggested as candidates for entropy-based anomaly detection. However, there may be difficulties in understanding the analysis capabilities provided by a set of entropy metrics used in conjunction with one another. The information entropy determination can quickly process a large amount of traffic data with little cost of calculation, but its accuracy relies on the selection of the threshold and it has certain drawbacks.

Recently, the implementation of ML techniques in network design, security, and management has provided the possibility of generating new network applications. ML tries to construct models that can learn to make decisions directly from data without following predefined rules. Data from past experiences is provided as input to the ML algorithm, which extracts patterns and builds a model to represent the data. This model describes the existing patterns in the data, so when it is given new unknown data, it should be able to make well-informed decisions. ML-based Intrusion Detection System (IDS) learns to classify events into the appropriate classes (normal or attack activity) based on experience given by the training set of rules. Each record, i.e. instance in the training set is represented by a given set of features and a class label indicating the attack type that the instance represents. Training sets for network attack detection contain records about network connections formed from the raw traffic data gathered from the network. Once trained and validated, the detection system is capable to detect both the attacks described in the database and their modifications, the attacks previously unknown to the system.

Detection of DDoS attacks at a proper time is crucial to protect normal activities on the SDN network. The important fact for any DDoS detection solution is distinguishing between legitimate traffic and DDoS attack traffic. It gets more challenging when the network is congested with legitimate traffic, and there is a need to segregate attack traffic safely without affecting the regular traffic. In such

cases, using statistical thresholds or a policy-based approach to detect threats may be inaccurate. This promotes the development of ML-based algorithms to categorize network data as either benign or malicious. Self-learning features of ML algorithms improve the efficiency of the detection strategy. ML-based DDoS detection usually involves the following three major steps, as shown in Fig. 4: (a) data preprocessing phase (b) training phase, and (c) testing phase. For all the proposed solutions in the available literature, the dataset is first preprocessed to transform it into the format suitable to be used by the ML algorithm. This stage typically involves encoding and normalization. Sometimes, the dataset requires cleaning in terms of removing entries with missing data and duplicate entries, which is also performed during this phase. The preprocessed data is then divided randomly into two portions, the training dataset, and the testing dataset. Typically, the training dataset comprises almost 80% of the original dataset size, and the remaining 20% forms the testing dataset. The ML algorithm is then trained using the training dataset in the training phase. The time taken by the algorithm in learning depends upon the size of the dataset and the complexity of the proposed model. The training time for the ML models requires more training time due to their deep and complex structure. Once the model is trained, it is tested using the testing dataset and evaluated based on the predictions it made. After that, the network traffic instance will be predicted to belong to either benign (normal) or attack class.

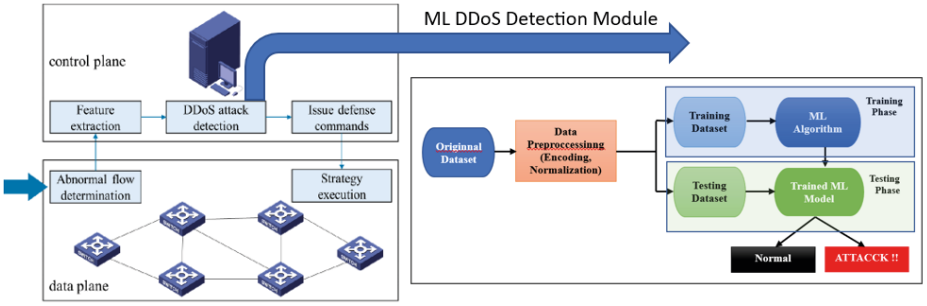

Figure 4 – *Generalized SDN ML-based DDoS detection system*

Existing DDoS detection ML algorithms generally fall into three categories: supervised learning, unsupervised learning, and semi-supervised learning (Sudar K. et al., 2020). Supervised learning is a method in which training data are labeled. To construct the classifier, the computer "learns" from the labeled patterns and uses them to predict labels for new data. In unsupervised learning, the training data have not been labeled, and the computer "learns" by analyzing data features to create the classifier. In a semi-supervised approach, the input training dataset typically consists of both labeled and unlabeled data, usually a small amount of

labeled data and a large amount of unlabeled data. Each algorithm has its benefits and drawbacks, as well as its application domain. Among these algorithms, the accuracy of DDoS attack detection ranges from 95% to 99.9%. That means that no algorithm can guarantee 100% detection in all the available architectures and diverse situations. Therefore, if only one algorithm is used for all situations, the results may not be as reliable as the predictions based on the dataset that was used to train the model. Hence, the best performing algorithms have been identified and combined to get better results under varied circumstances. The set of optimal features which were selected by different feature selection methods is used as an input for different machine learning classifiers. Among the many available ML classifiers, Decision Tree, Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Random Forest (RF), and Decision Tree (DT) are the most prominently used in DDoS detection systems (Ismail. et al., 2020).

There are two main ML-based approaches currently used to detect attacks on the SDN: simulation-based and public dataset-based approaches. In the first approach, researchers established SDN topology with legitimate hosts to generate normal traffics, and other hosts act as nodes to create DDoS attack traffics. They use public tools, such as Scapy, to simulate DDoS attacks. The network features, such as source/destination IP or port, entropy, flow packets, etc. are extracted from the collected traffic for normal and malicious data separately. All of these samples are random shuffling in a *.csv* file to create the row data which are used in the training model. The ML model can be used further to classify the normal and intruded DDoS packets inside the SDN network. This approach is fast and simple to analyze but with many restrictions. Firstly, the created dataset has a very small size and therefore, it is not enough to give accurate results, and these attacks are not realistic to represent the diversity of anomalies that are present in the current SDNs. Secondly, the number of extracted features is insignificant, and the small number of features is not enough to cover the behavior of all attacks (Ahuja N. et al., 2021). The selection of the proper public dataset has a significant impact on the evaluation of SDN IDS. Most of the publicly available datasets are not realistic, and they lack variety in the type of attack to cover all security trends found in the networks today. The most available datasets fail to give acceptable accuracy when deployed with intrusion systems. There are several datasets such as KDDCUP'99, CICIDS2017, ISCX2012, Kyoto, UMASS, ADFA, and DEFCON have been used for DDoS attack systems (Sahoo K. et al., 2020).

Table 2 – *ML attack detection model performance metrics overview*

Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$	Precision (PR) $\frac{TP}{TP+FP}$	Recall (TPR) $\frac{TP}{TP+FN}$	Specificity (TNR) $\frac{TN}{TN+FP}$
It is the ratio of correctly classified instances to the total number of instances	It is the ratio of correctly predicted attacks to all the instances predicted as attacks	It represents the ratio of all instances correctly classified as attacks to all the instances that are attacks	It represents the ratio of the number of correctly classified normal instances to all normal instances
True Positive (TP) - The number of correctly predicted attack instances			
True Negative (TN) - The number of correctly predicted normal instances			
False Positive (FP) - The number of incorrectly predicted attack instances			
False Negative (FN) - The number of incorrectly predicted normal instances			

The performance of the ML detection model is evaluated using the performance metrics like the accuracy, precision, recall, and specificity metrics and are computed as shown in Table 2. Recall (sensitivity, detection rate) is defined as true positive rate (TPR), i.e. the ratio of true positives and the sum of true positives and false negatives. The attack detection system with high recall has a low incidence of false negative alarms FNR (false negative rate), which means that a small number of attacks is incorrectly identified as normal network activities. The detection system with high TPR is used in critical areas of computer networks where the attack may not pass undetected. TNR (specificity) represents the ratio of true negatives and the sum of true negatives and false positives, and a detection system with high TNR has a low incidence of false positive alarms (FPR), which means that a small number of legitimate network activities are incorrectly identified as attacks. Although a compromise between TPR and TNR is usually made in practice, there are situations when it is necessary to use a system that will generate a small number of both false negatives and false positives. In these situations, a system with high detection accuracy is required.

To accurately distinguish entropy change caused by an anomaly, from the regular variation that is the result of stochastic traffic behavior, some approaches combine entropy and Artificial Intelligence (AI) techniques. AI is the development of intelligent machines representing a system that observes its environment and takes over activities that increase its chances of success using computer models. The advantages of applying AI are the ability to establish models that categorize the schemes used in detection, flexibility, and adaptability concerning precisely defining thresholds and rules, as well as the ability to learn. A group of authors in the paper (Vukovic I. et al., 2020) discusses the phases, components, categories,

and types of DDoS attacks and emphasizes detection solutions based on classification with information entropy and AI techniques. AI is used as an enhanced classification method, and the results in the paper (Kuk K. et al., 2017) highlight that the Monte Carlo approach presented via the BFTree classifier provides the best classification accuracy compared with other predictive models based on data mining classifiers. Furthermore, the authors in the paper (Cisar P., et al., 2022) represent an overview of the recently proposed artificial immune networks (AIN). The structures and learning algorithms of a few typical AINs are discussed.

HIGHLIGHTS OF EXPERIMENTAL WORKS

In this section, we highlight some experimental works related to the previously discussed entropy and ML-based DDoS detection mechanisms. Some works use statistical analysis, reporting on the complexity and operating costs of handling attacks. Other works have specific contexts to run the experiment, with particular configurations and constraints, and have designed the testing environment based on the specific parameters that correspond to their implemented approach. All experimental works are focused on the ML classification algorithms and consider the analysis of the entropy-based preprocessed network traffic data.

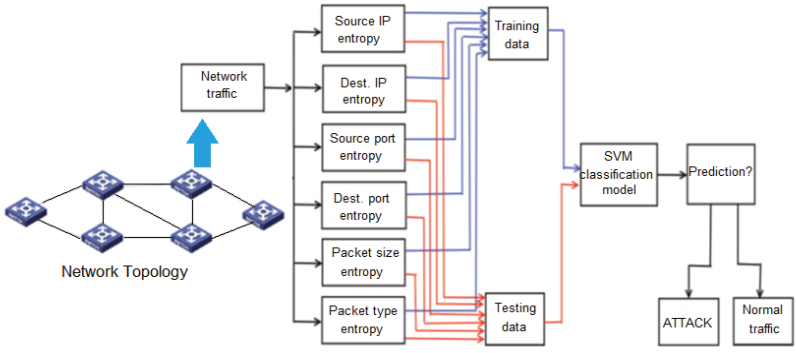

Figure 5 – *An example of a hybrid Entropy-SVM attack detection system [10]*

Starting with the experimental work (Dong Li et al., 2018), the authors proposed a model to detect DDoS attacks in SDN that is a combination of both entropy of network features and a Support Vector Machine (SVM) supervised classification algorithm. The model extracts several key features from the packet-in messages and measures the distribution of each feature by using entropy, then uses a trained SVM algorithm to detect the DDoS attack. SVM attempts to solve an optimization

problem that consists of finding decisions boundary in the feature space that separates data from different classes. In this solution, the entropy of network features is calculated first. The five packet features (srcIP, srcPort, destIP, destPort, packIn) are all random variables, so we firstly extracted and then calculated their entropy in a one-time window according to equation (2). During the time of the attack, entropy values are derived from its normal behavior, and used for detecting the anomaly in the traffic.

The SVM algorithm is composed of two steps, the first one is the features extraction, and the second step is the classification. Step 1 (initialization) represents that the features are extracted from all the training packets set and the entropy will be used to measure the distribution of each feature. Then, the calculated feature entropy will be used to train nonlinear one-class SVM. Step 2 (DDoS attack detection) represents that for each new test packet, authors extract features and calculate the entropy which will be given to the trained SVM model to decide if it is normal or abnormal. If the result is abnormal, it means that a DDoS attack happens.

To evaluate the performance of the proposed solution, the authors used Mininet emulation software to build the SDN network topology. The controllers are Floodlight and the Virtual-Machine server that has 64GB RAM and 32 core CPU. The experimental network adopts a three-layer structure: core, convergence, and access layer. Two controllers belong to the core layer, four switches belong to the convergence layer, and another four belong to the access layer. There are 50 hosts in the experimental topology. For simulating a real network environment, normal traffic should be triggered as background traffic, and it is produced by the traffic generator D-ITG periodically and the traffic ratio is TCP:UDP:ICMP = 85:10:5. The packet sending speed is about 1000 packets/s.

In the training step, the normal traffic is generated by the hosts in the network. The software Hping3 is used to simulate DDoS attacks with the spoofed source and destination IP address with an attack duration of 30 seconds. Once the time window is determined, the entropy is calculated to be a 6-dimensional vector. These vectors are the sample of the SVM model. The sample is divided into two groups, one triggered by normal, and the other triggered by DDoS attack traffic. DARPA1999 public data set is also used to train the SVM model. In the training step, the parameters of SVM are set to be fixed and used to analyze the real testing data.

Once the model is trained, the next step is to identify the type of attack and attacked hosts in the testing phase. An ML model is accurate if it correctly predicts the attack type during the attack. Further, the performance of the detection model is measured using the following metrics: PR (Precision), TPR (Recall), and F-score (detection time). Besides, the authors have used other ML algorithms, such as Decision Tree, Naïve Bayes, KNN, and Random Forrest, to analyze the traffic and detect DDoS attacks. The proposed detection solution outperforms all

other ML algorithms with higher accuracy and shorter detection time. Experimental results show that this solution gives 97.25% correctly classified instances and 2.75% misclassified instances, and the expected effect was achieved. The low false alarm rate is a good result and, on the other hand, it may be the proposed simulation of normal data flow if it is not comprehensive enough, which is what needs to be done in the future.

In the research paper (Yu S., et al., 2021) the authors proposed a cooperative DDoS attack detection framework based on entropy and an ensemble learning approach in SDN network environment, as shown in Fig. 6. The authors tried to solve how to reduce the burden of the controller and the SBI interface, as well as how to improve the attack detection speed while ensuring DDoS attack detection accuracy. Considering the programmable ability of the OpenFlow switch, data statistics and analysis are arranged on the edge switch, which can implement a part of the attack detection function to reduce the burden on the controller and improve the response speed of attack detection.

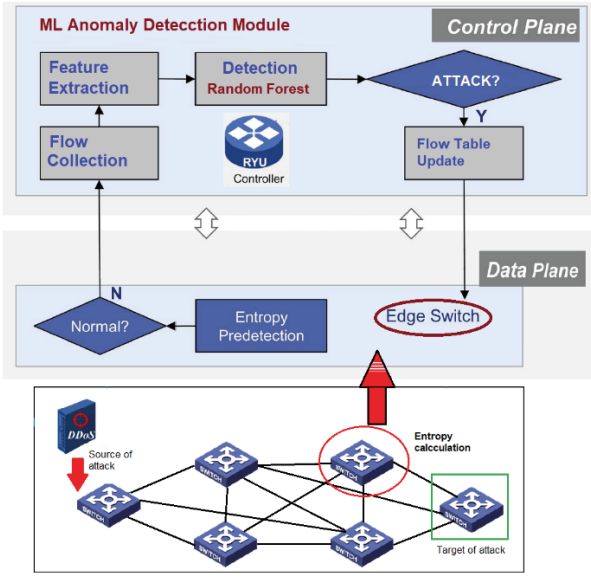

Figure 6 – *An example of a cooperative Entropy-RF detection system [26]*

During the experiment, Scapy software was used to inject normal traffic into the network as the background traffic, and then a TCP-SYN and ICMP flood attack were launched from the first switch (source of attack) to the last host (target of attack). The corresponding fast anomaly detection algorithm based on information entropy of the destination IP in the edge switch has been based on information entropy $H_n(X)$ (normal state) and $H_a(X)$ (attack state) values. Under normal cir-

cumstances, the information entropy value will fluctuate up and down in a small range. When a DDoS attack occurs, $H_n(X)$ and $H_a(X)$ satisfies the $H_n(X) - H_a(X) > \delta$ expression, the value of δ is determined according to the statistical information entropy under normal network state.

Considering the multiple feature tuple and requirement of less overhead in the detection process, the authors proposed the random forest algorithm (RF) to further detect the suspicious flow. Based on the consideration of ensuring detection accuracy while minimizing system overhead, the authors selected five most typical features to construct a 5-feature tuple (average number of packets, average number of packet bits, growth rate of port, growth rate of flow, and growth rate of source IP) for subsequent machine learning training and testing. Compared with other ML algorithms, RF random algorithm is a very convenient and practical algorithm which is more suitable for multivariate classification with less resource consumption and fast training speed. In the RF modeling process, the bagging sampling method was exploited to randomly select multiple training subsets from the original training set, while the CART algorithm was leveraged to generate K-decision trees to form the RF according to the principle of minimum impurity. The final anomaly decision was determined by voting the results of K-trees in the test set. Therefore, the test accuracy of the trained classification model on the test set is 0.997, indicating that this classification model has a very high accuracy for the detection of DDoS attack traffic.

CONCLUSION

Although SDN has many advantages, it also faces the threat of DDoS attacks, the most common security threat in contemporary networks. In response to this problem, we analyze the detection mechanisms of DDoS attacks over SDN, which combines information entropy and ML classification algorithms. The main issue of the entropy-based approach is the fine-grained traffic analysis, and accuracy of traffic variation detection. We have extended the entropy-based attack detection approach with the ML anomaly classification method to ensure that the attack traffic can be identified quickly and effectively. Different ML classification models are applied to the created dataset for classifying the traffic while performance evaluation is done with the help of performance indicators. For the effective validation of the ML classifiers, Random Forest and SVM are used with different topology scenarios. The efficiency of the anomaly classification method is validated through the presented experimental results. Our contribution addresses a practical implementation of the proposed method, using defined comprehensive architecture for flow-based anomaly detection that is based on the combined application of the entropy-based and ML techniques.

Finally, we believe that our work contributes to a better understanding of the DDoS attacks detection in SDN networks, despite the limited number of papers in this research field. Our further work will be oriented to the full implementation of the proposed architecture in a multi-controller and more complex SDN networks, focusing on better predicting the degree of certainty of detected network traffic anomalies.

REFERENCES

Ahuja, N., Singal, G. et al., (2021) Automated DDOS Attack Detection in Software Defined Networking, In Journal of Network and Computer Applications, Vol. 187, 2021, 103-108, https://doi.org/10.1016/j.jnca.2021.103108.

Basicevic, I., Blazic, N., Ocovaj, S. (2021) On the Use of Principal Component Analysis in the Entropy Based Detection of Denial-of-Service Attacks, Security and Privacy, Wiley, Vol. 4, Issue 1, doi: 10.1002/spy2.

Berde, P., Gerola, M., Hart, J. et al., (2014) ONOS: Towards an Open, Distributed SDN OS, In HotSDN: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ACM 2014, pp. 1–6. https://doi.org/10.1145/2620728.2620744

Bonguet, A., Bellaiche, M. (2017) A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing, In Future Internet, 9(3):43. https://doi.org/10.3390/fi9030043

Cabarkapa, D., Rancic, D. (2021) Performance Analysis of Ryu-POX Controller in Different Tree-Based SDN Topologies, Advances in Electrical and Computer Engineering, vol. 21, no. 3, 31-38, doi:10.4316/AECE.2021.03004

Cabarkapa, D., Rancic, D., Pavlovic, P., Milicevic, M. (2022) Investigating the Impact of Tree-Based Network Topology on the SDN Controller Performance, Facta Universitatis, Series: Automatic Control and Robotics, Vol. 21, No 1, 25-35, doi: 10.22190/FUACR211223003C

Cisar, P., Erlenvajn, D., Maravic-Cisar, S. (2018) Implementation of Software-Defined Networks Using Open-Source Environment, In Technical gazette, Vol. 25, Suppl. 1, pp. 222-230, http://dx.doi.org/10.17559/TV-20160928094756

Cisar, P., Maravic Cisar, S., Popovic, B., Kuk, K., Vukovic I. (2022) Application of Artificial Immune Networks in Continuous Function Optimization, Acta Polytechnica Hungarica, 2022, accepted for publication

Ding, D., Savi, M., Pederzolli F., Campanella, M., Siracusa, D. (2021) In-Network Volumetric DDoS Victim Identification Using Programmable Commodity Switches, In IEEE Transactions on Network and Service Management, vol. 18, no. 2, 1191-1202, doi: 10.1109/TNSM.2021.3073597.

Dong, Li, Chang Yu et al. (2018) Using SVM to Detect DDoS Attack in SDN, In IOP Conf. Series: Materials Science and Engineering 466 (2018) doi:10.1088/1757-899X/466/1/012003

Dong, S., Abbas, K., Jain, R. (2019) A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments, In: IEEE Access, vol. 7, pp. 80813-80828, doi: 10.1109/ACCESS.2019.2922196.

Dudeja, R. K., Bali, R. S., Aujla, G. S. (2022) Internet of Everything: Background and Challenges, In: *Software Defined Internet of Everything*, 3-15, Springer, doi: 10.1007/978-3-030-89328-6_1

Gupta, B. B., Joshi, R. C., Misra, M. (2009) Defending against Distributed Denial of Service Attacks: Issues and Challenges, Information Security Journal: A Global Perspective, 18:5, 224-247, doi: 10.1080/19393550903317070

Ibrahim, J., Gajin, S. (2022) Entropy-based Network Traffic Anomaly Classification Method Resilient to Deception, Computer Science and Information Systems, Vol. 19, Issue 1, 87-116, doi:10.2298/CSIS201229045I

Ismail et al., (2022) A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks, In IEEE Access, vol. 10, pp. 21443-21454, doi: 10.1109/ACCESS.2022.3152577.

Jimenez, M. B., Fernandez, D., Rivadeneira, J. E. et al., (2021) A Survey of the Main Security Issues and Solutions for the SDN Architecture, In IEEE Access, vol. 9, pp. 122016-122038, doi: 10.1109/ACCESS.2021.3109564.

Kuk, K., Milentijevic, I., Randjelovic, D., Popovic B., Cisar P. (2017) The Design of the Personal Enemy - MIMLeBot as an Intelligent Agent in a Game-Based Learning Environment, Acta Polytechnica Hungarica, 14(4): 121-139, 2017, ISSN 1785-8860

Open Networking Foundation: OpenFlow Switch Specification, Version 1.5.1, (2015), https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

POX Github Documentation, https://noxrepo.github.io/pox-doc/html/ (Last accessed on June 2022)

Sahoo, K. S. et al., (2020) An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks, In IEEE Access, vol. 8, pp. 132502-132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

Shalimov, A., Zuikov, D., Zimarina, D. et al., (2013) Advanced study of SDN/OpenFlow controllers, CEE-SECR '13: Proceedings of the 9th Central & Eastern European Software Engineering Conference, no. 1, 1-6, https://doi.org/10.1145/2556610.2556621

Sherwood, R. et al., (2009) FlowVisor: A Network Virtualization Layer, Deutsche Telekom Inc., R&DLab, Stanford University, Nicira Networks, Tech. Rep. OPENFLOW-TR-2009-1.

Sudar, K., M., Deepalakshmi P. (2020) Comparative Study on IDS Using Machine Learning Approaches for Software Defined Networks, International Journal of Intelligent Enterprise, Vol. 7, no.1-3, pp. 15-27, doi: 10.1504/IJIE.2020.104642

Villota, W., Gironza, M., Ordoñez, A., Caicedo, R. O. M. (2018) On the Feasibility of Using Hierarchical Task Networks and Network Functions Virtualization for Managing Software-Defined Networks, In IEEE Access, vol. 6, 38026-38040, doi: 10.1109/ACCESS.2018.2852649.

Vukovic, I., Popovic, B., Cisar, P. (2020) Application of Artificial Intelligence in Detection of DDoS attacks, Thematic conference proceedings of international significance, International scientific conference 'Archibald Reiss Days', Belgrade, University of Criminal Investigation and Police Studies, Belgrade, 10(2): 557-566.

Yu, S., Zhang, J., Liu, J. et al. (2021) A Cooperative DDoS Attack Detection Scheme Based on Entropy and Ensemble Learning in SDN, EURASIP Journal on Wireless Communications and Networking, 90, 2021, https://doi.org/10.1186/s13638-021-01957-9

Zargar, S. T., Joshi, J., Tipper, D. (2013) A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, In IEEE Communications Surveys & Tutorials, vol. 15, no. 4, 2046-2069, doi: 10.1109/SURV.2013.031413.00127.

Zhou, L., Zhu, Y., Xiang, Y. (2022). A novel feature-based framework enabling multi-type DDoS attacks detection, In Special Issue on Decision Making in Heterogeneous Network Data Scenarios and Applications, Springer, https://doi.org/10.1007/s11280-022-01040-3

Zhou, W., Li, L., Luo M., Chou, W. (2014) REST API Design Patterns for SDN Northbound API, 28th International Conference on Advanced Information Networking and Applications Workshops, 358-365, doi: 10.1109/WAINA.2014.153.