Entropy-KL-ML:Enhancing the Entropy-KL-based Anomaly Detection Method on Software-Defined Networks

Journal:	*Transactions on Network Science and Engineering*
Manuscript ID	TNSE-2022-02-0259
Manuscript Types:	Regular Paper Submission: Area 3 - Computer Networks
Keywords:	5.1 Data center flow control < 5. Data center networks, 6.3 Computer network security and privacy < 6. Computer networks, 6.7 Software defined networks < 6. Computer networks

Entropy-KL-ML:Enhancing the Entropy-KL-based Anomaly Detection Method on Software-Defined Networks

Nadia Niknami and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA

***Abstract*—The Software-Defined Networking (SDN) concept allows network innovations by leveraging a centralized controller that commands the whole network. The controller manages the functionality of the entire network. In the event of a controller failure, the entire network goes down. Because of vulnerabilities between the control plane and the data plane, Denial of Service (DoS) attacks often pose the greatest risk to SDN environments. The paper discusses a method to detect this attack before it leads to failure of the controller. The proposed combined anomaly detection method, which is called Entropy-KL-ML in this paper, uses entropy along with KL-divergence and ensemble learning to detect any uncertainty in incoming packets within time slots. An attack is detected when the calculated entropy passes or falls below the given threshold. KL-divergence and ML classifiers make the detection more accurate. We also present a new method for selecting features based on grouping the features that reduces the computational overhead of the controller. With an anomaly detection method in SDN, it is essential to provide a balance between overhead, accuracy, and processing time. Through a real-world data set and some anomaly detectors, we demonstrate that the Entropy-KL-ML method detects anomalies with greater accuracy and fewer overheads.**

Index Terms*—Anomaly detection, feature selection, classification, controller, Denial of Service (DoS) attacks, entropy, SDN

I. Introduction

Software-defined networks (SDNs) are basically the same as conventional networks, except for one crucial difference: the network controller [1]. Security issues for SDN are a significant concern. Due to SDN's design, in which the control plane is separated from the data plane, it is difficult for attackers to access that vulnerable logic [2]. On the other hand, since SDN's are centrally controlled, they are prone to DOS attacks, malware traffic injections, etc [3]. Monitoring and measuring network traffic flows are the means by which data integrity is maintained within SDN. One of the most significant aspects of traffic monitoring is the detection of anomalies. As the controller monitors the network traffic, anomalies can be detected and the traffic can be managed accordingly. The collection of accurate network traffic statistics and detecting intrusions or anomalies is crucial to improving network management. An attack on the controller can be detected by looking for anomalies in incoming packets [4]. An anomaly detection framework based on SDN includes three planes: the data plane, control plane, and application plane. Fig. 1 illustrates details of each plane as well as the relationships between them. There are switches, routers, and hosts in the data plane. It is necessary for all the switches to have a connection with the control plane to report any packets traveling through it. Application plane allows the controller to monitor and extract some information regarding incoming packets or flow information. The application plane provides the controller with a report about traffic anomalies, so that the controller can determine if this traffic should be treated differently. The controller sends decisions to the switch. This decision includes actions and rules that should be implemented on given switches for this type of traffic.

This research was supported in part by NSF grants CNS 2128378, CNS 2107014, CNS 1824440, CNS 1828363, and CNS 1757533.

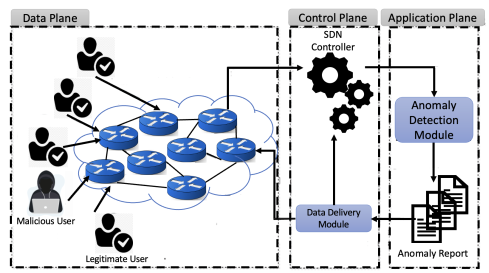


Fig. 1: SDN-based anomaly detection framework.

It is essential that we take into account workload, accuracy, response time, and overhead on the network when applying network monitoring policies, otherwise, the controller will not be able to effectively manage the network. Rather than examining every packet in a network, it is necessary to carry out analysis only on a part of the traffic in order to reduce overhead and processing time.Two approaches are available for detecting anomalies. The first is based on packet inspection. Packet collection modules of switches capture packets over underlying networks, whether they sample or not. The switches will then send captured packets to the controller, which will look for anomalies in the network. In SDN, packets are passed to the controller based on the controller resources, traffic flow between the switches and the controller, and switch table limits. For the second strategy, flow entries will be examined for anomalies. It is necessary for the controller to periodically send statistics request messages in order to collect flow entries in the switch. In a reply message, the switch must encapsulate its flow entries and send them to the controller. The purpose of this paper is to detect DoS attacks early. In order to detect anomalies in the distribution of traffic characteristics, such as source IP addresses and destination IP addresses as well as source ports and destination ports, we propose using a combined Shannon entropy and relative entropy, also known as KL-divergence. Entropy [5] and KL-divergence [6] are good measures of randomness. Entropy is used to analyze changes in traffic distribution, which reduces the controller's computational workload and KL-divergence computes how similar the new entry is comparing to the previous one. Additionally, applying these methods will provide additional information for categorizing dissimilar types of abnormalities, thereby improving detection capability [7] [8]. To calculate entropy, the controller must determine which attributes are more effective under different attack scenarios.

TABLE I: Main notations

Symbol	Meaning
E_X	Entropy of X
$KL_D(O\|\|A)$	KL-divergence of observed O and assumed A distributions
src_I/dst_I	Source/Destination IP address
src_P/dst_P	Source/Destination port
$src_I^{(S)}/dst_I^{(S)}$	Source/Destination IP address during short-term window
$src_I^{(L)}/dst_I^{(L)}$	Source/Destination IP address during long-term window
S	Source switch
n_α	Total number of packets in time interval α
t_α	Length of time interval α
R_α	Packet-in arrival rate in time interval α
τ	Threshold

The suggested solution presents a way to measure the unpredictability of incoming packets. The entropy measures the probability of an event occurring compared to the total number of events. During an attack, the entropy changes depending on what packet header fields are viewed and the type of attack, DoS and DDoS. During a DDoS attack, the entropy of the source IP addresses increases, while it decreases during a DoS attack. DDoS and DoS attacks, however, reduce the entropy of a destination IP address. Another strategy to detect changes in randomness involves comparing the entropy of one sample of packet header fields with that of another sample of packet header fields by using conditional entropy. In summary, we make the following contributions:

- We develop an entropy-based DoS attack detection algorithm by combining entropy with the KL-divergence statistical method.
- We leverage the KL-divergence property to aid entropy detection in the early stages of an attack.
- We improve the accuracy of an anomaly detection system by employing ML classifiers to Entropy-KL-Divergence anomaly detection. We called this method Entropy-KL-ML in this paper.
- We propose grouping features in the feature selection step to detect anomalies more accurately.
- We propose the idea of zooming the time window from a more extensive range to a smaller time window if any anomaly is detected, which can reduce the computational overhead on the controller.
- The results reveal that Entropy-KL-ML method beats entropy-based and machine learning-based anomaly detection methods on average.

II. Related work

A. Statistical-Based Detection Techniques

Models based on statistical data are a type of mathematical approach. A statistical approach to detecting anomalies in computer networks allows for a deeper analysis of packets. The goal is to find a standard connection between random and non-random variables. The relationship between the variables allows the results to be predicted. Authors in [9] primarily focus on DDoS detection and mitigation using predefined DDoS data-sets, which can be difficult to generalize for different network services and legitimate users' traffic patterns. By leveraging the controller's broad view of the network, the authors in [10] suggest a solution that is both effective and resource-efficient at the same time. Authors in [11] propose a novel DoS attack detection method based on entropy measures by taking out the hyper-parameters of window size and threshold, this approach computes entropy progression from line-of-best to unit-time. Authors in [12] propose a statistical approach to detecting modern botnet-like malware. By detecting abnormal network patterns, the authors show that entropy-based methods are suitable for detecting modern botnet-like malware. Mehdi *et al.* in [13] detect the presence of security problems in different Software-defined networks by estimating the distribution of benign traffic using maximum entropy estimation. The traffic is divided into packet categories including protocols and destination port numbers. A baseline benign distribution for each category is developed using maximum entropy estimation.

B. Machine Learning-Based Detection Techniques

Several machine learning and data mining techniques have been used to explain Intrusion Detection Systems [14]. Researchers identify the attack as malicious traffic by separating it from normal traffic. In a machine learning-based IDS, high detection rates are achieved along with low false-positive rates. DDoS attacks can also be detected with machine learning techniques. The authors in /cite[jose2021towards] describe a system for detecting different flooding attacks over SDN network traffic. There are a variety of classification methods the system uses to differentiate between normal and attack traffic. Authors in [15] design and implement a secure guard to assist in solving the issue of DDoS attacks on the POX controller, including a feature vector for classifying traffic flow using the SVM. Vetriselvi *et al.* in [16] develop a machine learning-based IDS to detect the attacks in SDN. The proposed IDS is composed of two modules. Modules are responsible for detecting and classifying attacks. The first module is deployed within the SDN switches, while the second implemented within the controller. As a result of the proposed attack detection method, the controller load is reduced and switches are less dependent on the controller.

C. Combination-Based Detection Techniques

Researchers combine techniques to enhance the detection of anomalies [17]. A combination of machine learning techniques and statistical approaches was used. This resulted in a reduction of delays in malicious identification. Authors in [18] propose an adaptive clustering method that includes a ranking of features to detect DDoS attacks. In their approach, the identification of DDoS attacks is based on an incremental clustering algorithm and feature ranking method. Authors in [19] propose an anomaly detection method that incorporates entropy with SVM. In their approach first it is needed to extract and classify the features. The control plane is controlled by POX controller. Authors in [17] implement a method using both entropy and sequential probabilities ratio test methods to remove the uncertainty associated with the entropy threshold.

III. Preliminary

The Software-Defined Network (SDN) is a new network architecture that provides central control over the network. An SDN network enables a highly programmable network environment by decoupling the data plane and the control plane. Providing visibility into the entire network topology and decoupling control logic from data forwarding are two of SDN's most important features. Controllers deploy services, which define control policies, in the control plane, and distribute these policies to the data plane through a standard protocol, such as OpenFlow. Using SDN, longstanding problems in networking, such as routing, policy-based network configurations, and security, are addressed in new ways. Even with the numerous advantages of SDN technology, the security of such a network remains a concern, since decoupling increases attack surface [20].

Security issues connected to conventional as well as trending networking technologies can also be addressed by SDN. A centralized controller, however, makes SDN susceptible to DDoS attacks [21]. SDN controllers are targeted by DDoS attacks that flood them with malicious packets. One failure in the SDN controller could compromise the whole network. OpenFlow switches as forwarding devices are responsible for collecting and forwarding data in an SDN environment. SDN controllers request flow statistics from OpenFlow-capable SDN switches periodically to gain situational awareness in the network. In each request, all flow table entries and counter values are queried and downloaded to the controller. The DoS detection feature can benefit from these features.

A. Anomaly Detection

Data flow information is analyzed and anomalies are detected by the anomaly detection mechanism. It obtains a large number of flow feature vectors by virtue of the SDN controllers [22]; A Denial-of-Service attack is a type of cyber-attack in which the attacker attempts to exhaust the network resources and make them inaccessible to the intended users. This disrupting the services can be done temporarily or permanently. The purpose of a DDoS attack is to bring down a target's services using several sources [23]. Typically, this is accomplished by flooding the target with superfluous requests to overload the system. As a result of a DDoS attack, the victim's incoming traffic is generated from various sources. This means that blocking a single source does not suffice to prevent the attack.

Since attacks can imitate the behavior of legitimate traffic, distinguishing legitimate from illegitimate traffic can be quite challenging [24]. With the SDN, the central controller receives periodic updates regarding the network, which enables it to detect attacks such as denial-of-service (DoS) attacks and network scanning [25]. By applying anomaly detection mechanisms to the gathered information, these kinds of attacks can be detected. DDoS attacks occur when an attacker's host generates a large amount of packets and sends them to the targeted switch. Valid users suffer from inaccessibility of controllers due to controller exhaustion and overuse of bandwidth. As a result, legitimate packets are dropped. It is obvious that DDoS attacks disrupt the SDN network as a whole. Typically, DDoS attacks target the communication channel's bandwidth, as well as other resources, such as memory, CPU, and system power. Since the rules of the packets do not match those of the flow table, the switch sends packet-in messages to the controller. By using FlowMod rules, the controller updates flow tables by sending them back to the switches [22]. The flow of packets between switches and controllers exhausts the controller's resources, the switch's flow table, and the controller's bandwidth.

B. Entropy

To summarize feature distributions, entropy can be used as a measure of uncertainty and randomness.

$$E_X = \sum_{i=1}^{n} -p(x_i) log(x_i), \tag{1}$$

where X is the feature that can take values $\{x_1, ..., x_n\}$ and $p(x_i)$ is the probability mass function of outcome x_i. To detect DDoS, entropy is often used for measuring the randomness of packets coming into a network. As the randomness increases, the entropy increases, as well. Traffic features can be measured by their entropy as a measure of regularity. A high entropy value indicates a scattering of features, while a low entropy value indicates convergence of features. A DoS attack, in which numerous packets are destined for the same IP address and port, will significantly reduce entropy. Thus, they can be detected as a drop in entropy [5]. Depending on the number of existing flows, entropy values calculated for each feature can generate extensive datasets. The values of entropy are represented as $E_n(X)$ and $E_a(X)$, respectively, based on the network's normal and abnormal states. During a normal state, the information entropy increases and decreases within a small range. During a DDoS attack, traffic to a specific IP address increases drastically, resulting in a lower entropy value. In

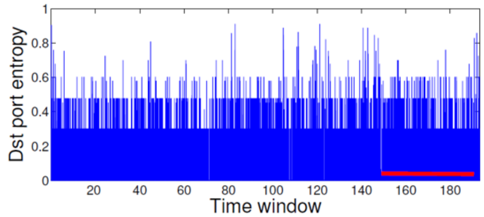


Fig. 2: Destination port entropy.

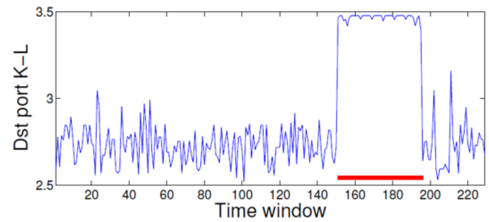


Fig. 3: Destination port relative entropy.

this case, $E_n(X)$ and $E_a(X)$ satisfy $E_n(X) - E_a(X) > \delta$. According to the statistical information entropy of the network when it is in its normal state, the value of δ is determined. Detecting DDoS using entropy involves two essential components: window size and threshold. The size of the window is determined either by the time period or by the number of packets. This window is used to measure uncertainty in the incoming packets by calculating the entropy. The window size serves as a unit of measurement for incoming traffic. In the header field, the targeted parameter is measured for every window. To detect an attack, a threshold is needed. An attack is detected when the calculated entropy passes or falls below a threshold, depending on the scheme. The window size and threshold are used to detect this pattern.

Another version of entropy is conditional entropy, which measures the uncertainty of a variable, Y associated with a random variable, X. The variable X is known in most cases. An average dependence between two features is captured by conditional entropy. By measuring conditional entropy, one can determine whether it is possible to predict the first feature based on the importance of the second feature. Based on the first feature, it shows how much uncertainty is left about the value of the second feature. A promising approach would be to detect network anomalies by monitoring this relationship between features.

$$E_{(src|dst)} = \sum\nolimits_j -p(dst_j) \sum\nolimits_i p(src_i|dst_j) log(p(src_i|dst_j)) \quad (2)$$

In the above formula, $p(dst_j)$ represents the percentage of packets arriving at certain destination address j, or dst_j, among examined packets. $p(src_i|dst_j)$ is the proportion of packets originated from source address i in the total number of packets are supposed to arrive at dst_j. All other combinations such as $E_{(src|length)}$ and $E_{(src|dst_P)}$ can also be achieved in the same manner, where $length$ represents the length of the packet and dst_P represents the destination port. For analysing traffic distribution, the use of entropy will provide greater detection capability than volume-based methods [26]. A further advantage of the entropy method is that it provides additional information for categorizing dissimilar anomalies.

C. KL-divergence

Different time intervals should be considered when modeling network behavior. An ongoing attack is suspected if the network behavior varies from one interval to the next. In addition to the degree of uncertainty, we must also consider the extent to which the assumed and observed distribution of traffic on the network differ. When the assumed and observed distributions of traffic are denoted as A and O, the difference between two probability distributions over $x_1, ..., x_n$ can be found as follows:

$$KL_D(O||A) = \sum\nolimits_{i=1}^{n} -O(x_i) log(O(x_i)/\ A(x_i)). \quad (3)$$

This type of entropy is called relative entropy, also known as the Kullback-Leibler divergence [27]. Intuitively, Kullback-Leibler divergence or KL-divergence measures how far an observed distribution is from the true distribution. If two distributions perfectly match, then $KL_D(O||A) = 0$, otherwise it can take values between 0 and ∞. A lower KL divergence value indicates that the approximation is closer to the true distribution. In addition to computing the distance between two PDFs, it can detect the start of a new attack. In contrast, another attack is ongoing [28]. Fig. 2 and Fig. 3 illustrate the different results of entropy and KL-divergence respectively. In this scenario, there is a DDoS attack in the interval $[150, 195]$. The red lines indicate the interval, containing the attack. While the entropy value for the destination port failed to detect this DoS attack, the KL-divergence value for the destination port shows the start and end of the attack.

IV. Entropy-KL-ML Anomaly Detection Method

Monitoring traffic and metrics in SDN switches provides security. It also comprises a collection and preprocessing module, a learning module, a detection module, and a flow management module. The proposed model detects anomalies in packets and flow-level traffic instances passing through SDN switches.

An illegal packet is one whose entropy is less than the threshold; otherwise, it is compared to another threshold value. Generally, a larger value indicates that the packet came from an authorized user. However, picking a threshold value is difficult. This is mostly determined by how many false positives there are. While the entropy of the source IP addresses increases during a DDoS attack, it decreases during a DoS attack. Both DoS and DDoS attacks cause destination IP address entropy to decrease. Using conditional entropy, a method for predicting the correspondence between source and destination IP addresses can be made easier to distinguish the abnormal traffic because DDoS attacks consist of multiple sources converged at one destination.

As entropy alone would not be sufficient to detect the attack, it would be highly dependent on the threshold chosen to detect the attack. The KL-divergence alone would not be sufficient as it couldn't distinguish between the beginning and end of an

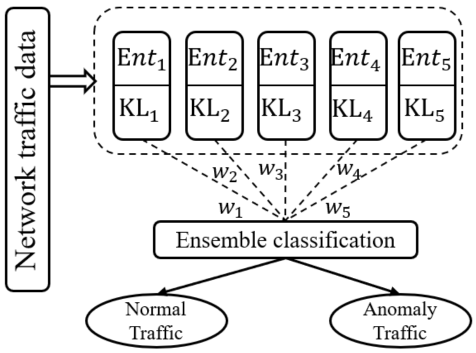


Fig. 4: Anomaly detection method.

attack. These results are stronger when they are combined. Fig. 4 illustrates the combination method of entropy and KL-divergence with different features and different weights. Weight can be set based on the importance or correlation of the given features and the problem. The majority of current DDoS detection methods in a single control plane are based on machine learning technology, which has been shown to be an effective classifier. We also include ensemble learning [29] [30] in this paper to perform more accurate detection of abnormal flow. By using ensemble learning, we use multiple learning algorithms to obtain a better prediction than could be obtained by using any of the individual learning algorithms alone. In our proposed framework, Entropy-KL-ML, multiple base components jointly reached to the final decision. The result of entropy and KL-divergence for each group of features is a new feature for classifiers. For instance, if there is SVM classifier in the ensemble learning section of this framework, the values of w_1, w_2, w_3, w_4, and w_5 would be set by SVM classifier. Ensemble machine learning helps us find the importance of features and have an accurate result at the end of the anomaly detection process.

Distributions can be categorized into two types. The first is flow header features such as IP addresses, ports, and flow sizes. The second class consists of behavioral features, like the number of different destination addresses or source addresses that a host communicates with. Analyzing the distribution of traffic features can help detect the attack discussed above. A traffic feature is a field in a packet header. It is possible that short-lived anomalies are not picked up by anomaly detection algorithms when statistics are collected for a long period of time. A collection made every few seconds, however, can generate a great deal of traffic on the network, as well as a great deal of workload on the controller.

A. Feature Processing

Identifying essential features improves classification accuracy and reduces computational complexity. Combining feature selection methods would increase classifier performance by identifying features that are ineffective individually but effective collectively in order to detect anomalies, removing unnecessary features, and identifying features that are highly correlated with the output class. The features that have been used to detect the presence of DDoS attacks are as follows:

Algorithm 1 Entropy-KL Divergence-ML-based Anomaly Detection (Entropy-KL-ML)

Require: Received packets and Threshold τ

$n_\alpha \leftarrow 0$, $t_\alpha \leftarrow 20$
$\xi(src_I), \xi(dst_I), \xi(src_P), \xi(dst_P) \leftarrow \{\}$
$\xi(src_S) \leftarrow \{\}$
Update τ_α
repeat
  if Packet-in message arrives **then**
    $src_I, dst_I, src_P, dst_P, src_S \leftarrow$ *Parse*(packet-in)
    Update $\xi(src_I), \xi(dst_I), \xi(src_P), \xi(dst_P)$
    Update $\xi(src_S)$
    $n_\alpha = n_\alpha + 1$
until (End of Analysing)
$R_\alpha \leftarrow n_\alpha / t_\alpha$
$\pi \leftarrow E - KL_{(src|dst)}, E - KL_{(src|dst_P)}$
$\phi \leftarrow \text{Entropy}_{Grouping}[\text{src}_I, dst_I, src_P, dst_P, src_S, R_\alpha]$
Flag $\leftarrow$ Evaluate π and ϕ with τ_α
return Flag

- Number of flows
- The average number of packets per flow
- The average of flow duration
- The entropy value of source IP address
- The entropy value of destination IP address
- The entropy value of protocol
- Source port
- Destination port
- Conditional entropy value of source and destination

It is unclear what feature distributions perform the most effectively. A number of feature distributions have been proposed in the past. The most recommended features are: 1) header-based features such as addresses, ports, and flags 2) volume-based features such as host-specific percentages of flows, packets, and bytes 3) behavior-based features such as in/out connections for a particular host. Considering combination and relations on different features of packets and flows such as packet type, src_I, dst_I, (src_I, src_P), (src_I, dst_P), (dst_I, src_P), (dst_P, src_P), and (src_P, L) would be helpful in this regard.

Algorithm 1 represents the details description of the proposed combined anomaly detection method. Source IP address (src_I), destination IP address (dst_I), source port (src_P), destination port (dst_P), source switch (src_S), and packet-in message arrival rate (R) are the features employed in the anomaly detection part of algorithm. $E_{src_I}, E_{dst_I}, E_{src_P}, E_{dst_P}$, and E_{src_S} stand for the entropy value of source IP address, destination IP address, source port, destination port, and source switch respectively. The periodic time interval t_α used to start the detection periodically and set to 30 seconds here. The dictionary $\xi(src_I)$, $\xi(dst_I)$, $\xi(src_P)$, $\xi(dst_P)$, and $\xi(src_S)$ are used to store src_I, dst_I, src_P, dst_P, and src_S calculated in the given time window respectively. Our idea in the proposed combined anomaly detection method, Entropy-KL-ML, is using the time window

TABLE II: Feature Selection

#	No. Features	Selected Features
0	1 feature	E_{src_S}
1	1 features	$E_{(src_I \mid dst_I)}$
2	2 features	E_{src_I}, E_{dst_I}
3	4 features	E_{src_P}, E_{dst_P}, E_{src_I}, E_{dst_I}
4	5 features	E_{src_P}, E_{dst_P}, E_{src_I}, E_{dst_I}, E_{src_S}
5	6 features	E_{src_P}, E_{dst_P}, E_{src_I}, E_{dst_I}, E_{src_S}, R

zooming method. Initially, the detector considers a larger time window to check traffic behavior over the given network. If any suspicious traffic or any anomaly is detected, the detector changes the time window to a smaller time window to monitor the traffic accurately. Using time window zooming can reduce controller computational overhead. As a result, the controller only monitors a small time window when necessary, rather than monitoring all traffic in detail.

B. Grouping Features

A set of elements can be analyzed using the entropy method, but it does not give much insight into contributing elements. A similar packet distribution is observed in attack flows [31]. In addition, when an attack begins, the flow and packet distribution of the entire traffic changes. In order to view packet counts in the attack flows, it is helpful to group partial flows in each time window based on specific criteria. Grouping the features improves the detection result in the case of decreasing the overhead. It allows us to analyze the anomaly in larger groups. If there is abnormal behavior in the group, we need to check the sub-group of the suspicious group in order to find the anomaly accurately [31]. After parsing the packet-in, the controller can find source IP address, destination IP address, source port, and destination port. To group the features, we need to quantify them. Based on the number of occurrence of specific feature (i^{th} source IP address) and the total number of occurrence for all of this feature, the value of entropy quantifies given feature. The quantified value of source IP address is as $E_{s_{IP}} = \sum_{i=1}^{k} \frac{\omega_i}{\Omega} log(\frac{\omega_i}{\Omega})$, where ω_i is the occurrence number of i^{th} source IP address in $S_{IP} = \{\omega_1, \omega_2, ..., \omega_k\}$, k is the number of different source IP addresses, and Ω stands for total number of occurrence of source IP addresses. For grouping the number of packets, we can consider different group numbers based on the different range of packet counts [32]. For example, group #1 is for $count_{packet} = 0$, group #2 for $2 \leq count_{packet} \leq 10$, group #3 for $11 \leq count_{packet} \leq 100$, and group #4 for $count_{packet} \geq 101$. The range of packet count in each group should be considered based on the amount of traffic in the network.

Another feature is Packet-in arrival rate which can be computed by n_α/t_α where n_α presents total number of packets in time interval α and t_α is the length of the time interval. R_α stands for Packet-in arrival rate which equals to $R_\alpha = n_\alpha/t_\alpha$.

V. Performance Evaluation

This section of the paper describes the experiment employed to test the Entropy-KL-ML algorithm's performance, using different combinations of features and varying window times. In this section, our goal is to determine whether the proposed combined method can be used to detect DoS attacks accurately and whether it is more effective than pure entropy-based or pure ML-based anomaly detection methods. We use a hybrid evaluation of simulated traffic over a real system to evaluate the proposed combined approach. For the simulation, the dataset is based on real legitimate traffic and synthetic anomalies. For the experimental process, our system includes a controller, some SDN switches, and 16 hosts. There is an ONOS controller in the system, and we designate one of the hosts as an attacker to inject an anomalies into the network. We compare the Entropy-KL-ML method with combined and straightforward detection methods, including entropy, KL-divergence, and some well-known classifiers. We will show that it is possible that the classifier would have better detection if it uses the results of entropy on different features of traffics. We evaluate our proposed framework, Entropy-KL-ML, on a real system and two simulated networks. The framework is assessed based on the detection rate, accuracy, overhead, processing time, and false positive rate.

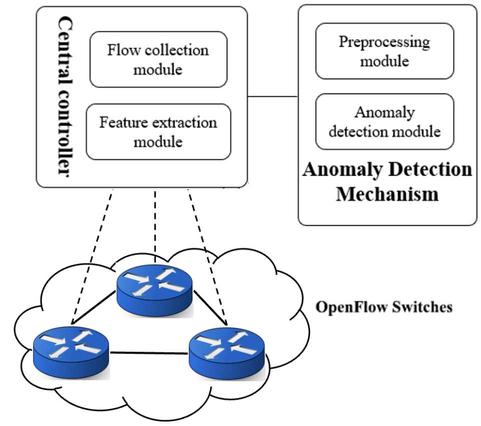


Fig. 5: Structure of Anomaly Detection in an SDN.

A. Testbed Data

According to Fig. 6, the data center structure consists of 35 servers, 16 SDN switches, and some regular L2 switches. Pica8 p-3922 switches are used to make this SDN network. Two networks were set up: a control network and a data network. An L2 switch in the control network connects all management ports of SDN switches and the SDN controller (gateway). SDN switches are configured as out-of-band controllers, which means they separate the control plane from the data plane. The dotted lines in Fig. 6 show the structure of this network in details. The topology of this network is a star structure. There are connections between the data ports on the SDN switch and the gateway in the data network. This is a complete binary tree three-level topology. The gateway is connected to the root SDN switch, and other servers to leaf SDN switches. We use Open Network Operating System (ONOS) as the SDN controller.

We use ONOS as the controller and use Mininet to generate different network topologies. Mininet allows the creation of

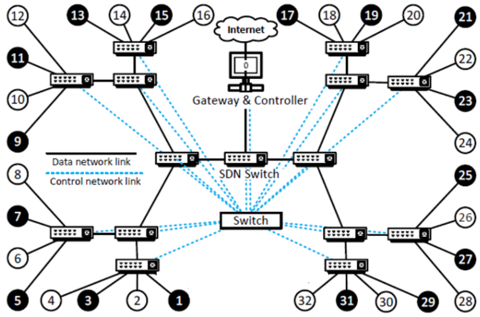


Fig. 6: Data center topology [33].

a realistic virtual network that runs real kernel, switch, and application code, and supports the development of OpenFlow applications. ONOS and Mininet can run on the same windows desktop with a 3.5GHz Intel Core i3 CPU and 16GB memory. We attached one host to each switch for the first topology, Standford, with 26 switches, 26 hosts, and 650 flows. However, for the second topology, we attached one host for each edge switch. The second topology is FatTree(4), with 20 switches, 16 hosts, and 240 flows. For each network, we generate a flow of the same rate between each host pair. DoS attack detection is not dependent on the network's size, but on the number of packets that make up the window. This is because we consider a simple network topology. During the experiment, we inject normal traffic into the network by using scapy, and then a DoS attack was launched from a switch to a host. Under normal network conditions, the entropy value for the traffic is about 0.8. We set $\delta = 0.2$ in the simulation.

To define decision strategy, we consider short-term and long-term entropy, as well as dynamic thresholds. Entropy of recent windows is short-term entropy. The long-term entropy, however, represents the entropy of earlier windows. Identifying an attack requires a particular method of decision-making. Decision strategies are Boolean-valued functions with entropy vectors and thresholds. As one example, one of the scenarios takes into account different thresholds for short-term and long-term scenarios. The function can be described as follows:

$$(E^{(S)}_{dst_I} < \tau_{Short} \quad \& \quad E^{(L)}_{dst_I} < \tau_{Long}) \tag{4}$$

and

$$(E^{(S)}_{src_I} < \tau'_{Short} \quad \& \quad E^{(L)}_{src_I} < \tau'_{Long}), \tag{5}$$

where $E^{(S)}_{dst_I}$, $E^{(L)}_{dst_I}$, τ_{Short}, and τ_{Long} are entropy of destination address in the short-term window and entropy of destination address in the short-term window, short-term threshold, and long-term threshold, respectively. Similarly, $E^{(S)}_{src_I}$, $E^{(L)}_{src_I}$, τ'_{Short}, and τ'_{Long} are entropy of source address in the short-term window and entropy of source address in the short-term window, short-term threshold, and long-term threshold respectively. The scenario is based on satisfying one of the conditions in Eq. V-A and Eq. V-A or both of them. Dynamic thresholds can be set in various ways. For instance, one scenario can portray as $\tau_t = \frac{1}{k}\sum_{j-t-k}^{t-1} \tau_j$, where τ_j stands for the threshold of time interval j, t is the current time, and k is an arbitrary integer. Therefore, threshold of current window t, τ_t, is the average of the last k thresholds.

TABLE III: Compare different methods with AC and FPR

Method	AC(%)	FPR
Entropy	65.13	0.06
Entropy and KL_D	76.12	0.05
Entropy, KL_D, and SVM	81.21	0.035
Entropy, KL_D, and RF	78.01	0.045
Entropy, KL_D, and Ensemble	82.10	0.034

In this study, different ML algorithms and feature selection methods are employed to detect DoS attacks. We examine SVM, KNN, Random Forest(RF), and Linear Regression(LR). The Support Vector Machine (SVM) [34] predicts the most appropriate decision function for separating two classes. Essentially, it is based on the definition of a hyperplane that separates two classes. The KNN approach [35] to anomaly detection is unsupervised. There is no predefined labeling of normal or anomaly, since the thresholds are the only factors determining the level of detection. The spikes in distance measures are potentially anomalous. A random forest [36] creates a number of trees to predict whether a class is normal or anomalous. The final class prediction is produced based on a majority vote of the class predictions for each tree.

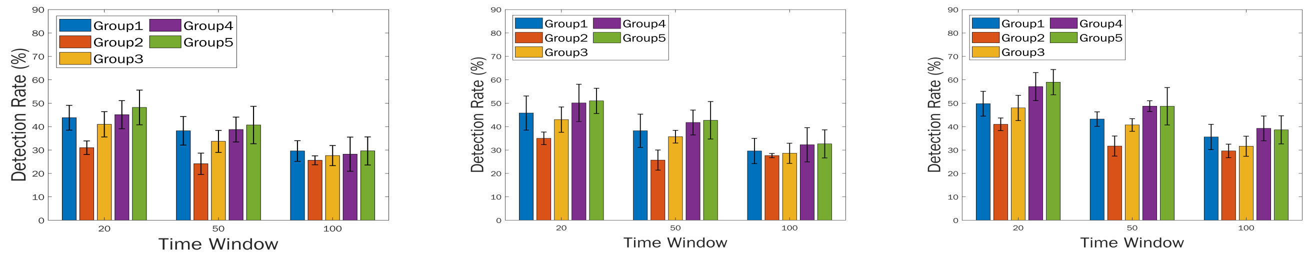
As one of the controller's responsibilities, it gathers information from the flow tables in the switches. The controller monitors the currently running flows and keeps track of how many packets are arriving with each flow. In this paper, we propose to use this property and enjoy incorporating some feature processing into the decision-making process. One of the factors that should be considered is the length of the monitoring window. There are several factors that determine window size. These factors include incoming data to each host, the number of switches connected to the controller, and how long it takes for the computation to complete.

We adopt process time, overhead, False positive ratio (FPR), and accuracy to evaluate the algorithm performances. The FPR indicates the probability that the detector incorrectly classifies the packet, when in reality it may be normal. This is computed as $FP/(TN + FP)$, where FP represents the number of normal flows identified as anomaly flows, and FN represents the number of normal flows identified as normal flows. The accuracy is the ratio of correctly classified samples to the total number of samples, indicating the classifier's discrimination abilities. Table III shows the result of anomaly detection with different methods based on the accuracy and false positive ratio. We have to pick the method with the lowest possible FPR and highest accuracy.

B. Results

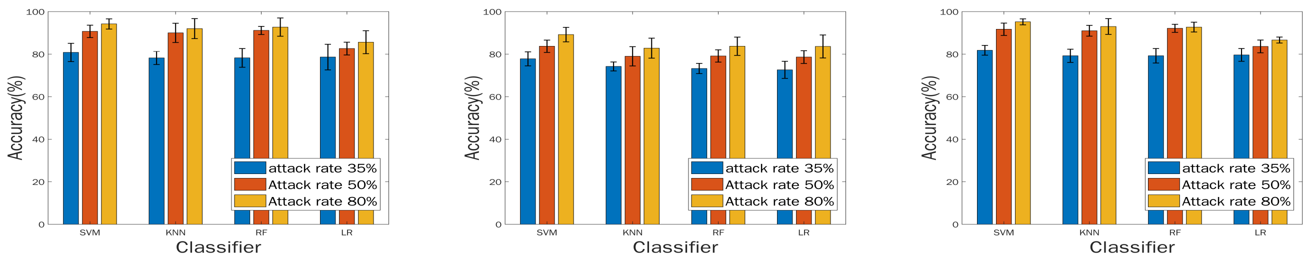
The proposed combined method is evaluated under different scenarios with different detection methods, different topologies, and different attack rates 35%, 50%, and 80%.

1) The Effect of Different Grouping methods on Detection Rate : A DoS detection rate is determined by the ratio of the number of flows that are classified as DoS to the total number of flows in a given time interval. In this section we are going to



(a) Attack rate 35%　(b) Attack rate 50%　(c) Attack rate 80%

Fig. 7: Evaluation of detection rate in scenario with different attack rate on different window time.



(a) Standford　(b) FatTree(4)　(c) Star

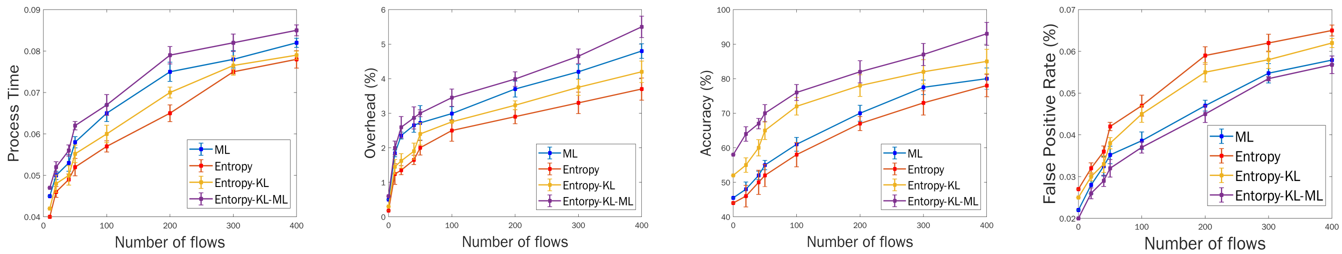
Fig. 8: Overall detection accuracy over different classifiers.

compare the performance of the proposed combined detection method with regards to the detection rate for the grouping approach. Fig. 7 illustrates the detection rate of Entropy-KL-ML anomaly detector under different scenarios with different attack rates. This figure shows the results according to the groups of features that are summarized in Table II. Results are presented in time windows of 20, 50, and 100. Results show that there is the increase in detection due to the small-time slot for anomaly detection. Due to the small time slots, the detection results are more accurate. At the same time, however, the controller has to perform more computational work. In implementing the proposed combined anomaly detection method, we used the idea of window time zooming. In addition, if we look at the results for different groupings, regardless of the length of the time window, entropy-based KL anomaly detection with the help of ML has higher detection rate on group 5 which includes E_{srcP}, E_{dst_P}, E_{src_I}, E_{dst_I}, E_{src_S}, and R. Therefore, grouping features is a successful method of detecting anomalies in the SDN. To evaluate the performance of the proposed combined anomaly detector, it is necessary to examine the processing time, overhead, and FPR.

2) The Effect of Classifier and Topology on Accuracy: This experiment tests whether Entropy-KL-ML method can detect anomalies over different topologies accurately. Additionally, we test whether the topology of the network impacts anomaly detection. There are two simulated topologies, Standford and FatTree(4), as well as one real system, which is a star topology (Fig. 6). In the Standford topology, we attached one host to each switch, while in the FatTree(4) topology, we attached one host to each edge switch. From Fig. 8, we see that topology does not have a considerable effect on anomaly detection results. The classifier is another parameter that the experiment considers. Results show that SVM can be used to predict the most appropriate decision function to distinguish between two normal and anomaly classes. Other classifiers are close to SVM, especially when the attack rate is 80%.

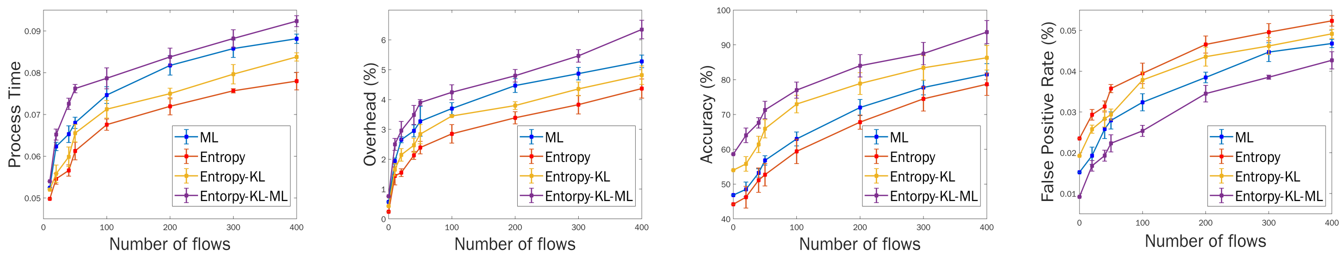
3) The Effect of Number of Flows on Processing Time, Overhead, Accuracy, and FPR: Our objective is to compare the proposed combined approach with other anomaly detection methods on the basis of processing time, overhead, accuracy, and FPR. The experimental results in Fig. 9 shows that Entropy-KL-ML method has the best accuracy among other anomaly detection approaches such as pure entropy, pure ML, and the combination of entropy and KL-divergence. As shown in Fig. 9(a), the proposed combined method has a larger processing time in comparison with other approaches, but it is not considerable. The entire process of proposed combined anomaly detection consumes some computational time. As shown by Fig. 9(b), when the number of flows increases, CPU utilization increases for all approaches, although the Entropy-KL-ML approach uses CPU at a much lower rate than the other approaches. As the results demonstrate, using KL-divergence in conjunction with entropy increases the overhead of the controller compared to the pure entropy approach. Even so, Entropy-KL-ML methods, which combine Entropy-KL and ML with some new processing on features, have significantly better results compared to other methods.

As shown in Fig. 9(c) and Fig. 9(d), for the Entropy-KL-ML approach there is higher accuracy (around 91.9%) and lower FPR (around 0.055%) in comparison with other approaches. Entropy-KL also has acceptable accuracy (81.7%) compared to other approach, but it is obvious that combination ensemble learning with Entropy-KL in the proposed combined approach helps anomalies detectors to make devise decisions in detection process. SDN needs a method for detecting



(a) Processing time (b) Overhead (c) Accuracy (d) FPR

Fig. 9: Evaluation of processing time, overhead, accuracy, and FPR



(a) Processing time (b) Overhead (c) Accuracy (d) FPR

Fig. 10: Evaluation of processing time, overhead, accuracy, and FPR in the short-long scenario.

anomalies that balances overhead, accuracy, and processing time. Although the proposed combined anomaly detection method has a high processing time, the high accuracy and low false positive rate make the Entropy-KL-ML approach a distinctive anomaly detector.

4) The Effect of Number of Flows on Processing Time, Overhead, Accuracy, and FPR in short-term scenario: Fig. 10 illustrates the results when we used a short-term and long-term time threshold for Entropy-KL-ML approach. Obviously, less processing time is a positive point for the approach because it can provide an early anomaly detection in the network. In the case of using short-term and long-term threshold, which is shown in Fig. 10(a), the processing time has been reduced, and there is a shorter processing time to detecting anomalies. Similarly, as Fig. 10(b) shows short-term and long-term entropy decreases the overhead of the controller even in based line approaches such as pure entropy. Fig. 10(c) and Fig. 10(d) present the results of measuring accuracy and FPR in the short-long scenario for different anomaly detectors. In the case of accuracy, considering short-term threshold improves the accuracy of detection from 91.9% to 93.6%. Also, there is much smaller FPR (around 0.0425) when we use short-term threshold for our approach compared to when we consider a predefined threshold which has FPR$= 0.056$.

Based on the experimental results, we can see that more features lead to better performance. Network traffic parameters, such as the entropy of source IP address and the entropy of destination IP address, are generally analyzed in pairs, as are the entropies of source port and destination port. KL-divergence, in combination with entropy, removed the uncertainty associated with entropy thresholds. The results illustrate that as we expected combining KL-divergence with entropy increases the accuracy and decreases the FPR in the process of anomaly detection. In addition, ensemble learning under proposed feature selection improves the detection results under different scenarios. By using a small-time slot for anomaly detection, there is a higher rate of detection of anomalies. This leads to more accurate detection results. It was found that the topology has a negligible effect on detecting anomalies, but the type of classifier has a significant impact on the results.

VI. Conclusion

Since the controller plays such a central role in SDN, there are some security and vulnerability concerns. Denial of Service (DoS) attacks may cause the controller to become unavailable. Analyzing and observing flows becomes much easier with statistical approaches that detect anomalies. Such approaches can be performed in real-time and are faster than other techniques. We propose in this paper a method for classifying the traffic in the SDN control plane as normal or abnormal using an entropy-based anomaly detection system. The Entropy-KL-ML method was implemented in order to identify DoS attacks. For detecting abnormal traffic variations, we combined entropy and relative entropy with machine learning algorithms. KL-divergence and entropy were used at the same time to reduce the uncertainty associated with the entropy threshold and improve the entropy results. Ensemble learning has also helped us avoid making misjudgments or misleading the entropy-based detectors.

References

[1] S. Singh and R. K. Jha, “A survey on software defined networking: Architecture for next generation network,” *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 321–374, 2017.

[2] P. Vaid, S. K. Bhadu, and R. M. Vaid, “Intrusion detection system in software defined network using machine learning approach-survey,” in *Proc. of IEEE 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021.

[3] S. Garg, A. Singh, G. S. Aujla, S. Kaur, S. Batra, and N. Kumar, "A probabilistic data structures-based anomaly detection scheme for software-defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3557–3566, 2020.

[4] G. Garg and R. Garg, "Detecting anomalies efficiently in sdn using adaptive mechanism," in *Proc. of IEEE 5th International Conference on Advanced Computing & Communication Technologies*, 2015.

[5] S. Oshima, T. Nakashima, and T. Sueyoshi, "Early dos/ddos detection method using short-term statistics," in *Proc. of IEEE International Conference on Complex, Intelligent and Software Intensive Systems*, 2010.

[6] L. Zhang, D. Veitch, and K. Ramamohanarao, "The role of kl divergence in anomaly detection," in *Proc. of ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, 2011.

[7] T. Radivilova, L. Kirichenko, and A. S. Alghawli, "Entropy analysis method for attacks detection," in *IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, 2019, pp. 443–446.

[8] M. Z. Abdullah, N. A. Al-awad, F. W. Hussein, Y. Zhao, S. Li, Y. Yang, P. Ding, H. Sun, C. Xiong, and Y. Li, "Implementation of entropy-based distributed denial of service attack detection method in multiple pox controllers," *Review of Computer Engineering Studies*, vol. 6, no. 2, pp. 29–38, 2019.

[9] R. M. A. Ujjan, Z. Pervez, K. Dahal, W. A. Khan, A. M. Khattak, and B. Hayat, "Entropy based features distribution for anti-ddos model in sdn," *Sustainability*, vol. 13, no. 3, p. 1522, 2021.

[10] S. M. Mousavi and M. St-Hilaire, "Early detection of ddos attacks against software defined network controllers," *Journal of Network and Systems Management*, vol. 26, no. 3, pp. 573–591, 2018.

[11] O. Subasi, J. Manzano, and K. Barker, "Denial-of-service attack detection via differential analysis of generalized entropy progressions," *arXiv preprint arXiv:2109.08758*, 2021.

[12] P. Bereziński, B. Jasiul, and M. Szpyrka, "An entropy-based network anomaly detection method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, 2015.

[13] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *International workshop on recent advances in intrusion detection*. Springer, 2011, pp. 161–180.

[14] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[15] S. E. Kotb, H. A. T. El-Dien, and A. S. T. Eldien, "Sguard: machine learning-based distrbuted denial-of-service detection scheme for software defined network," in *Proc. of IEEE International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021.

[16] V. Vetriselvi, P. Shruti, and S. Abraham, "Two-level intrusion detection system in sdn using machine learning," in *Proc. of International Conference on Communications and Cyber Physical Engineering*, 2018.

[17] B. H. Ali, N. Sulaiman, S. A. R. Al-Haddad, R. Atan, S. L. M. Hassan, and M. Alghrairi, "Identification of distributed denial of services anomalies by using combination of entropy and sequential probabilities ratio test methods," *Sensors*, vol. 21, no. 19, p. 6453, 2021.

[18] L. Zi, J. Yearwood, and X.-W. Wu, "Adaptive clustering with feature ranking for ddos attacks detection," in *Proc. of 4th International Conference on Network and System Security*, 2010.

[19] K. M. Aung and N. M. Htaik, "Anomaly detection in sdn's control plane using combining entropy with svm," in *Proc. of IEEE 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020.

[20] G. Garg and R. Garg, "Review on architecture and security issues in sdn," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 6519–6524, 2014.

[21] A. D. Lopez, A. P. Mohan, and S. Nair, "Network traffic behavioral analytics for detection of ddos attacks," *SMU data science review*, vol. 2, no. 1, p. 14, 2019.

[22] T. Jafarian, M. Masdari, A. Ghaffari, and K. Majidzadeh, "A survey and classification of the security anomaly detection mechanisms in software defined networks," *Cluster Computing*, vol. 24, no. 2, pp. 1235–1253, 2021.

[23] T. Srikanth, P. Branch, J. Jin, and S. Jugdutt, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, no. 1, 2020.

[24] T. Thapngam, S. Yu, W. Zhou, and G. Beliakov, "Discriminating ddos attack traffic from flash crowd through packet arrival patterns," in *Proc. of IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, 2011.

[25] M. H. Khairi, S. H. Ariffin, N. A. Latiff, A. Abdullah, and M. Hassan, "A review of anomaly detection techniques and distributed denial of service (ddos) on software defined network (sdn)," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2724–2730, 2018.

[26] R. N. Carvalho, J. L. Bordim, and E. A. P. Alchieri, "Entropy-based dos attack identification in sdn," in *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 627–634.

[27] N. M. AbdelAzim, S. F. Fahmy, M. A. Sobh, and A. M. B. Eldin, "A hybrid entropy-based dos attacks detection system for software defined networks (sdn): A proposed trust mechanism," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 85–90, 2021.

[28] Z. Goldfeld, K. Greenewald, J. Niles-Weed, and Y. Polyanskiy, "Convergence of smoothed empirical measures with applications to entropy estimation," *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4368–4391, 2020.

[29] Y. Zhong, W. Chen, Z. Wang, Y. Chen, K. Wang, Y. Li, X. Yin, X. Shi, J. Yang, and K. Li, "Helad: A novel network anomaly detection model based on heterogeneous ensemble learning," *Computer Networks*, vol. 169, p. 107049, 2020.

[30] N. Iftikhar, T. Baattrup-Andersen, F. E. Nordbjerg, and K. Jeppesen, "Outlier detection in sensor data using ensemble learning," *Procedia Computer Science*, vol. 176, pp. 1160–1169, 2020.

[31] S. Yu, W. Zhou, and R. Doss, "Information theory based detection against network behavior mimicking ddos attacks," *IEEE Communications Letters*, vol. 12, no. 4, pp. 318–321, 2008.

[32] H. Lotfalizadeh and D. S. Kim, "Investigating real-time entropy features of ddos attack based on categorized partial-flows," in *Proc. of IEEE 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2020.

[33] R. Biswas, S. Kim, and J. Wu, "Sampling rate distribution for flow monitoring and ddos detection in datacenter," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2524–2534, 2021.

[34] M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions," in *Proc. of 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2015.

[35] T. T. Dang, H. Y. Ngan, and W. Liu, "Distance-based k-nearest neighbors outlier detection method in large-scale traffic data," in *Proc. of IEEE International Conference on Digital Signal Processing (DSP)*, 2015.

[36] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *Proc. of International Conference on Data and Software Engineering (ICoDSE)*, 2017.