Stochastic Modelling and Computational Sciences

FEATURE DIMENSIONALITY REDUCTION OF APPLICATION LAYER DDOS ATTACK IN SDN ENVIRONMENT

Sarabjeet Kaur[1] and Abhinav Bhandari[2*]
[1]University Institute of Computing, Chandigarh University, Gharuan, Mohali, India
[2]Department of Computer Science & Engineering, Punjabi University, Patiala, India
[*]bhandarinitj@gmail.com

ABSTRACT
Distributed denial of service (DDoS) attacks has been one of the most threatening attacks on the network for many years. During the lockdown period, most of the organizations opted to work in online mode. Due to the increase in online activities, the internet shifted the dependency of every sector. The attackers turned on to breach the security of the application layer of the network. They kept making various organizations' web services unavailable to genuine users. The detection of applications is more complex than the network and transport layer, due to the difficulty in differentiating the legitimate user and the attacker. This involves a series of features to be analyzed to differentiate the user behavior and the bot. In this paper, we have worked on the user behavior features by analyzing the weblogs and the publicly available "InSDN" dataset of application layer DDoS attacks in Software Defined Networks. We have used three feature extraction techniques Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA) to find the different dimensions of the dataset. Using the extracted features, we have deeply investigated the similarities and dissimilarities between normal and attack traffic to further detect DDoS attacks on the Software Defined Network (SDN) application layer. Then, we implemented machine learning (ML) classifiers such as Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM) to find optimal results with different combinations of these feature extraction and classifier algorithms. The study's main purpose is to reduce feature dimensionality to get high detection accuracy using different ML classifiers.

Keywords: Application layer, DDoS attacks, Feature extraction, SDN, User behavior.

1. INTRODUCTION
In today's world, most organizations are highly dependent on internet services. For instance, the services like banking, education, defense, medical, etc., have transformed the traditional way of working with the immense growth and adoption of internet usage. Using the internet means these sectors have started offering their services through web-based applications. It means the availability of these organizations is online through applications. Here we cannot ignore the increasing security issues over the internet with its increased usage. It allows attackers to breach the security of such types of web applications. One of the crucial threats over the internet is distributed denial of service attacks. It is an attack to make the web server unavailable through multiple systems. Many attacks have happened over the year to make the web services of organizations unavailable. The idea of the denial of service attack is implemented by flooding a vast number of messages or traffic to the victim such that the victim's machine's regular operation is hampered. It may tend to restart, crash, or make the web server unavailable for a legitimate user. The denial of service attack (DoS) attack is from a single machine. When multiple systems are used to flood such an attack, it is distributed denial of service attack (DDoS). DDoS attacks are becoming more dangerous for organizations providing online services. When targeting the application layers of the network, the DDoS attack becomes challenging as the differentiation between genuine traffic and attack traffic is troublesome. This attack destroys the organization's dignity by making its services unapproachable. The DDoS attack is an effective weapon used by attackers, now generating income for cybercriminals. Denial-of-service attacks can be launched at any time, which may target your website anyway, let it be its working or the web resources, and may cause enormous amounts of service interventions and surely large amounts of financial losses. As discussed, the most prevalent and challenging attack is AL DDoS attack (Application Layer DDoS attack), which is difficult to detect. This is because in an AL DDoS attack, the attacker finds an open user session

and the attacker sends the auto-generated HTML web requests to a web server, usually done by programmers. That seems like a legitimate user does website surfing. Because the application-layer DDoS attacks resemble real traffic, it is quite a crucial task to defend and detect such attacks. So far, several studies on application-layer DDoS attacks have been reported. But in Software Defined Networking, researchers still do not consider it popular.

The detection of application layer DDoS attacks in legacy and SDN networks majorly depends on the feature set used for the purpose. The pre-processing of a dataset and feature extraction to get different dimensions through various machine-learning techniques is the most important task for getting high-rate detection results. There are a few techniques available that help to reduce the features based on scaling and standardization. After getting a new feature set through these techniques, the machine learning classifiers can be used to detect the application layer DDoS attack with more accurate result possibilities.

1.1 Motivation and Contributions

Most organizations today depend upon the internet facility to run their business. Online services like electronic mail, education, banking, entertainment, trading, medical, etc., ease customer dealing. These online activities use the protocols of the application layer of the network. This attracted the attackers to focus on the network's application layer activities and services. Attackers started targeting the webserver to choak its benefits for the legitimate user. They rely upon DoS/ DDoS attacks for the purpose. This is known as the application layer DDoS attacks. Attackers violate the HTTP protocol of the application layer by sending either slow HTTP-GET/POST requests or by sending such incomplete requests, known as low rate and high rate attacks. This, in turn, keeps the web server busy for an ample time, thus making it unavailable for the genuine user. Attackers launch such attacks by mimicking the behavior of the actual user. This is the reason that it becomes difficult to detect the attack traffic. Detecting the application layer DDoS attacks needs to observe the features of attacks clearly; it all depends upon the behavior of bots during the launch of an attack.

In its Q4 2021 report, Cloudflare mentioned the astound hike in DDoS attack occurrences. But in the Q4 2020 report of "Kaspersky Lab", these attacks have become less common. The reason for decline in the Q4 report was its increase at the beginning of the year due to switching offline mode to online mode. As per the Cloudflare report, there was a 75% increase from 2020 to 2021 in DDoS attack launches. The most extended DDoS attack happened in 2019 for 509 hours on target. There was an increase in AL DDoS attacks, recorded a 641% rise from quarter to quarter in attacks that targeted the manufacturing industry. As of December 2021, there were more network-layer DDoS attacks than all other attacks seen in Q1 and Q2 of 2021. The year 2021 was the year of DDoS attacks. In the Q3 2021 report of Cloudflare, New Zealand faced a vast hit of DDoS attacks. The DDoS attacks on VoIP providers in Q3 2021 shattered the virtue of companies in some countries like Canada and the US. In early and mid-July 2021, the attackers targeted the resources of the security agencies of Russia and Ukraine with massive dump traffic. In mid of August 2021, attackers made the web resources of the "Philippine human rights organization Karapatan" inaccessible for users. At the end of August 2021, the website of "Germany's Federal Returning Officer" becomes a victim of DDoS attacks due to the September 26 elections. So, on the whole, the attack actors have covered the whole year 2021, making suitable enough income targeting the online services of companies. The North Korean internet services were downed in January 2022 for 6 hours as per records.

We identified numerous user behavior, browsing, and weblog analysis features. In particular, our work aims to identify a set of features for better representation of user behavior and to investigate the relative traffic coming from DDoS attackers and normal users. Finally, we used FE and ML classifiers to show the prediction and score of accuracy by taking an “InSDN” dataset of the application layer DDoS attacks in SDN environment. The major contributions of our research are: -

- Finding the elementary application layer features.

- Comparing the existing feature extraction techniques applied for application layer DDoS attacks.
- Implementing different classifiers on the extracted features.
- Comparing the results of different classifiers with features extracted through different techniques.

The content of this paper is organized as follows: in Section 2, we have discussed related research published on application DDoS detection and feature discussion. In Section 3, the feature extraction techniques used for experimentation are briefly discussed. In Section 4, we have given detail of the meaningful feature set identified application layer DDoS attack features. Section 5 has two parts of experimentation as data Pre-processing and using Feature Extraction techniques with ML classifiers. This section presents and compares the obtained experimentation results. In Section 6, we conclude the paper with final remarks.

2. RELATED WORK

DDoS attack detection is the need of today's network security framework and has become a sore topic of network security discussion nowadays for researchers. Many researchers have attained significant achievements in detecting and defending DDoS attacks on the network. The episodes are increasing with internet use, and techniques to protect against those attacks are also emerging. Over time the attackers have moved towards the application layer DDoS attacks due to the substantial online dependency of users. In an application layer DDoS attack, the legitimate user could not reach up to online services due to their unavailability. It is the most crucial attack and hard to detect due to its resemblance to genuine traffic. As per the related research, the Web server is the main target of application DDoS attacks. The focus of current research is turning toward the detection of such attacks. These attacks can be either detected based on characteristics of traffic flow and information it carries or can be detected by analyzing user behavior to differentiate between DDoS attackers and legitimate users.

(Singh et al., 2017, 2018) [3][4], in their paper, discussed a few points that differentiate legitimate users from attackers. The authors highlighted a few features that generally represent legitimate users. The author's genuine user uses bookmarks to open the marked pages and navigates through pages using search engines and hyperlinks, whereas bots don't. They also say that legitimate user uses legible mouse clicks and scrolls on web pages. Genuine users accessed popular web pages and didn't repeat access patterns using legitimate web browsers. On the other hand, few features to recognize bots are also given by authors, and bots are generally dependent on the attacker's command to launch the attacks. Bots use fake identities to mimic legitimate users. Most of the time, the attackers use the same browsing pattern to access web pages. The attackers iterate their access patterns. Attackers concentrate on geographical distribution while launching an attack, unlike legitimate users.

(Sreeram et al., 2017) [1], used the Bat algorithm technique to detect application layer DDoS attacks. The authors took time frame length and session time as prime features for the related literature review. It signifies how long a user maintains the session with every request. After analyzing the in-depth involvement of these features, the author says that these are just considered random cases. So, in this article author worked on the absolute time interval for the detection process. (Saravanan et al., 2016) [2], considered a few features like flow similarity, behavior analysis, client legitimacy, and type of web page requested for detection mechanism deployed on proxy system. Authors compare the proposed mechanism with five related techniques earlier in the literature to show the accuracy. (Singh et al., 2017, 2018) [3][4], worked on user behavior analysis with the identification of various related features. Authors in a survey paper in 2017 [4], considered server load, target page, repeated page, single page, multiple page, and dominant page as key features to analyze the user behavior of browsing. This browsing pattern helped detect whether the request was from a legitimate user or attacker. In his paper from 2018 author invented a four-fold index request index, response index, popularity index, and repetition index and used a support vector machine classifier to detect application layer DDoS attacks.

(Daneshgadeh et al., 2019) [5], combined machine learning algorithms for attack detection using the attributes time Interval, source IP entropy, destination IP entropy, and received bytes entropy. The authors consider the IP address tracing and bytes received per request in the particular time interval for research. (Bravo et al., 2018) [6],

worked on user behavior analysis by tracing the attributes of mouse and keyboard activities. The user characteristics showing the pattern of mouse and keyboard clicks were detected and used for the attack detection mechanism. (Jaafar et al., 2019) [7], his recent review focused on the articles that worked on Session flooding, request flooding, asymmetric attack, and slow request/response attack. (Singh et al., 2015) [8], presented simulation study for application layer DDoS attack in a legacy network. Several features like Successful transaction, transaction throughput, response time, dropped transaction, Legitimate Queue Utilization, Normal Transaction Survival Ratio, failed, longest, and shortest transaction is the cornerstone of an article for detection mechanism. (Jazi et al., 2017) [9], focused on HTTP-based DDoS attacks in the network and used sampling techniques for detection. The key features considered by the authors were Request/response received/sent, no incoming/ outgoing packets/ sessions/ connections, and packet size. Another survey paper by (Odusami et al., 2020) [10], analyzed the attributes like User session, packet pattern, header information, time interval, and web user feature used to do meta-analysis and detect the application layer DDoS attacks by various researchers. The authors analyzed different attributes and their proportion of detection efficiency. (Bhandari et al., 2016) [11], presented another study with extensive analysis of flash events and DDoS attacks in a legacy network. Authors framed the differences between the flash crowd and DDoS attacks giving a clear picture of the type of traffic on a network. The study focused on the duration of requests received, then the number of requests received, unique IPs traced, and the average number of requests received on a network.

Apart from legacy networking, another concern nowadays is Software Defined Networking (SDN) security. Few researchers talked about application layer DDoS attacks in the SDN environment. However, the literature says that the Application layer is ignored by the researchers yet. Researchers either used SDN architecture to secure the network or worked on its security. It has been marked that the architecture of SDN itself is vulnerable to various network attacks. One among those is a DDoS attack. (Hong et al., 2018) [12], used SDN-assisted Slow HTTP DDoS attack defense methods to work on incomplete HTTP requests. The authors worked in the SDN environment to validate the research. The simulation parameters like web server, SDN controller, Attacker, Slow Client, and legitimate client were taken. (Benzaid et al., 2020) [13], proposed robust application layer DDoS self-protection architecture to detect application layer DDoS attacks in the SDN environment by considering the server's resource depletion and response time to requests as attributes. In another study by (Yungaicela et al., 2021) [14], the authors used a combination of machine learning and deep learning models. Source IP, Source port, destination IP, destination port, and protocol are the key features identified by the authors. (Park et al., 2021) [15], proposed HTTP DDoS Flooding Detector (HDFD) implemented using IP address analysis and source port number. Authors worked to defend the web server and prevent network resource depletion through HTTP DDoS attacks in Software Defined Networking paradigm. (Wang et al., 2021) [16], used Credibility-Based Countermeasure (CCSA) to find out client credibility. CCSA blocks the client with low credibility. The number of requests per client and source IP are taken as the main attributes.

In the article, A. Praseed et al. [20] (2019) highlighted the challenges related to application in Legacy network. Authors explored the critical features illustrating how the attacks are made by the attackers. As there are numerous statistical features of HTTP flood attacks. These features are required to be standardised. The authors discussed the studies from the literature where the Principal Component Analysis and Independent Component Analysis techniques are used to reduce the feature dimensions. This review highlighted the various features used for detecting the HTTP flood attacks with different detection techniques.

(Yungaicela-Naula et al., 2021) in [14] used feature extraction technique PCA (Principal Component Analysis) to perform the data cleaning. Authors considered 4 modules for collecting, pre-processing, detecting and managing the traffic flow using the CICFlowMeter. The article proposed the architecture with above 4 modules for detecting transport layer and application layer DDoS attack characteristics. This article is based on Software Defined Networking and used the MININET emulator with ONOS controller for experimentation. For collecting the traffic flow, CICFlowMeter which is developed by Canadian Institute for Cyber Security has been used. With this tool, bi-directional flow packets are generated and analyzed. Thereafter machine learning techniques are used

such as random forest, SVM, KNN, and MLP in the detection module. The datasets used for experimentation were CICDoS2017 and CICDDoS2016. The PCA mechanism resulted in reducing the set of features to 15 for both data sets for this 85% of the variance was used in PCA.

(Rustam et al., 2022) [21], the recent study implemented a few feature extraction techniques. Among those, PCA is also one of the techniques used by authors to perform experimentation evaluation. The study was validated using CICIDS 2017 dataset. Though the study talked about other layer DDoS attacks as well. The main focus of the author's research revolves around the application layer of legacy networks.

(Mohanad et al., 2021) [19], used feature extraction algorithms PCA, AE, and LDA for the datasets UNSW-NB15, ToN-IoT, and CSE-CIC-IDS2018. Although the author revealed that, based on the experimentation there is no feature extraction technique giving the best score for considered datasets. The study found that one technique is better than another one in the case of different machine learning classifiers with different datasets. The authors explored the effects of applying all three feature extraction techniques on different deep-learningning classifiers. The results were presented a as classification matrix of different datasets.

These feature extraction techniques are used far back by researchers for solving classification problems as highlighted in table 1. (Nojun et al., 2001) [17] in his study used the Independent Component Analysis algorithm for feature extraction. New features are identified to validate the proposed algorithm by the authors in the proposed study. The ICA algorithm extracted 7 features for multi-layer perception. (Ogundokun et al., 2022) [22] used Independent Component Analysis (ICA) technique for feature reduction verifying it with SVM classifier for finding best results. Authors used KDDCUP 99 dataset to validate the results. Another study by (Jaber, A. 2022) [23] implemented LDA algorithm for multilayer DDoS attacks prevention using a hypervisor. Mostly the researchers use Principal Component Analysis for feature reduction as found in literature study.

Table 1: Few articles which implemented PCA, ICA, and LDA

Feature Extraction Technique	Year	Reference	Attack type	Datasets Used
PCA	2019	"A. Praseed and P. S. Thilagam [20]"	AL DDoS	--
	2021	"Yungaicela-Naula, N. M., Vargas-Rosales, C., & Perez-Diaz, J. A. [14]"	AL DDoS	CICDoS2017, CICDDoS2018
	2022	"Jaber, A. [23]"	MultiLayer	--
	2022	"Rustam, Furqan & Mushtaq, Muhammad & Hamza, Ameer & Farooq, Shoaib & Jurcut, Anca & Ashraf, Imran. [21]"	AL DDoS	CICIDS 2017
ICA	2001	"Kwak, N., Choi, C.-H., & Choi, J. Y. [17]"	Other	--
	2019	"A. Praseed and P. S. Thilagam [20]"	AL DDoS	--
	2022	"Ogundokun, R.O., Misra, S., Bajeh, A.O., Okoro, U.O., Ahuja, R."	MultiLayer	KDDCUP 99
LDA	2022	"Jaber, A. [23]"	MultiLayer	--

3. Feature Extraction Techniques Used

Feature extraction techniques reduce the dataset features based on statistical analysis. These techniques extract the features or dimensions from the raw input of dataset features. The features are extracted with minimum loss of information. Though there are several techniques available for feature extraction. We have used three techniques on our database i.e PCA, ICA, and LDA. The result comparison of the three algorithm outputs is then compared by using ML classifiers.

PCA: - Principal Component Analysis technique of feature extraction. This is based on statistical calculation. It is an unsupervised machine learning algorithm that provides the set of extracted features. We may provide the

number of components to be reduced for our new dataset. PCA then uses the method to find agien vectors and based upon those highest values, the new dimensions are found. It provides the dimensions without changing the valuable information. The converted features are represented as different variations rather than data labels.

ICA: - Independent Component Analysis is another technique similar to the PCA feature extraction technique. It is basically a linear dimensionality feature reduction algorithm where the input of a set of features is given and ICA extracts the number of components.

LDA: - Linear Discriminant Analysis is a supervised reduction technique that maximizes the distance of mean for each class it leads to the better classification of the data set this algorithm can also be used as a machine learning classifier.

4. Application Layer Feature Identification

Analysing the features is the most critical phase of the detection process for application layer DDoS attacks. This selection of attributes affects the detection outcome and performance. With the selection of stable feature set, prediction accuracy can be significantly improved for application layer DDoS attack detection. We have to analyse the weblog attributes for this purpose deeply. The general structure of a web blog is as shown below: -

1) "::1 - - [24/Feb/2022:10:53:43 +0530] "GET /favicon.ico HTTP/1.1" 200 30894 "http://localhost/dashboard/phpinfo.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36""

2) "#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs-version cs(User-Agent) cs(Referer) cs-host sc-status sc-substatus sc-win32-status sc-bytes cs-bytes time-taken 2022-07-06 12:37:46 ---.--.---.-- GET /wp-content/plugins/CCSlider/includes/upload.php – (port no) - ---.--.--.--- HTTP/1.1 ALittle+Client – (Domain Name) 404 0 2 1436 207 315"

- The above are two weblogs taken from the Apache log file and the client website. The first one shows that page phpinfo.php has been accessed using Chrome browser with 98.0.4758.102 Ip address, and it shows the success code 200 for this access. In total, 30894 bytes were transferred during access. The second log taken from the server shows information like the date and time of access to the website and the Ip address of the client and user, which is kept hidden in the above example for security reasons. The method of entry is GET, as shown above. The protocol HTTP/1.1 is shown through which the request has arrived. It also specifies the port number through which the request has reached, and the web page upload.php has been accessed. Code 404 shows the error in page access. If the code is 200, it shows the request is successfully fulfilled. The client information has been removed from the log above. The information embedded in a weblog cannot be used directly for a dataset. It contains redundant information and some irrelevant information from the point of view of DDoS attacks, which was removed. Table 2 gives the set of application layer attributes used in the literature. We can figure out a few features to detect application layer DDoS attacks as: -

- "Access pattern"Access pattern is the complete flow of web page visits by attackers which is constantly repeated.
- "Average query string's length of client" Average of attempts made by attacker or user to access web page.
- "Client legitimacy" The legitimacy of a user sending a number of requests can be checked against the cluster of a previously known client.
- "Duration of the access" Access time interval length. For how much time a user has accessed the web page.
- "Request type (like GET/POST/OTHER)" The fractions of request types per connection.
- "Entropy of the requests" Entropy to measure the amount of frequent change in the packet and request flow in the form of an HTTP GET request at multiple time slots.

- "Flow similarity" Identifying the similarity in traffic flow.
- "Access path redundancy" Identifying the common path used by user to access particular domain.
- "HTTP GET request count"
- "IP address" Identifying the source IP addresses from where the traffic is originated.
- "Maximal, minimal and average packet size" Size of forwarding packet and average.
- "Size of TCP window" Number of packets received in particular time zone.
- "Maximal, the minimal and average time to live (TTL)"
- "Number of bytes sent in 1 second" Identify the number of bytes sent from the client to the server and vice versa.
- "Number of different resource paths of client"
- "Number of packets sent in 1 second" Per second packet transfer
- "Number of requests" number of open requests currently in a window
- "Number of users" number of legitimate users on server
- "Percentage of encrypted packets with different properties"
- "Session's requests" number of current open requests
- "Sum of the incoming payload of all clients of domain"
- "Sum of an outgoing payload of all clients"
- "Sum of response times of all clients of domain"
- "Sum of response times of client" Time taken to receive first packet from client
- "Users browsing process" The pattern of browsing sequence of user
- "Web page requested" Most popular web page visits
- "Total number of requests by a user" Number of requests received from a particular user.
- "Navigation between webpages" User way of accessing the web pages one after another sequentially.
- "Total number of requests for hot pages or popular pages" Requests for most visited pages.
- "Header data" Information wrapped up in the header of a packet can be used.
- "Type of webpages explored" It is important to analyse the type of pages generally attackers explore.
- "Frequency of page visits" How repeatedly an attacker visits a particular web page.
- "Number of web pages accessed at the same time" Another important feature is to see how many pages are opened by the same user at one time.
- "Number of bytes transferred" Amount of data transferred through one request.
- "Number of bytes transferred during uptime"
- "Session creation rate" Number and duration of sessions created by one user at a time.

- "Hyperlink depth sequence" Link of web page access.
- "Time at which request is received for a particular webpage"
- "Workload of request on the server" The amount of load a server has to bear against the request received.
- "Single URL repeat attack" If the same URL is targeted several times.
- "Multiple URL repeat attack" Set of URLs attacked several times.
- "Randomly selecting DDoS attack-based page link"
- "Time interval between request repeats" It's the time gap between a set of the same type of requests.

Table 2: Few elementary application layer attributes

Features	Publication Reference
Time frame length and session time	"Sreeram, I., & Vuppala, V. P. K. (2017)" [1]
Flow similarity, client legitimacy, and web page requested	"Saravanan, R., Shanmuganathan, S., & Palanichamy, Y. (2016)" [2]
Request index, response index, popularity index, repetition index	"Singh, K., Singh, P., & Kumar, K. (2018)" [3]
Server Load, target page, repeated page, single page, multiple pages, dominant page	"Singh, K., Singh, P., & Kumar, K. (2017)" [4]
Time Interval, Source IP Entropy, Destination IP Entropy, and Received Bytes Entropy	"Daneshgadeh, S., Kemmerich, T., Ahmed, T., & Baykal, N. (2019)" [5]
user's characteristics, mouse functions, and right click	"Bravo, S. & Mauricio, David. (2018)" [6]
Session flooding, request flooding, asymmetric attack, slow request/response attack	"Jaafar, G. A., Abdullah, S. M., & Ismail, S. (2019)" [7]
The successful transaction, transaction throughput, response time, dropped transaction, Legitimate Queue Utilization, Normal Transaction Survival Ratio, failed, longest, shortest transaction	"Singh, Barjinder & Saluja, Krishan & Bhandari, Abhinav. (2015)" [8]
Request/response received/sent, no of incoming/outgoing packets/sessions/connections, packet size, DR	"Jazi, H.H., Gonzalez, H., Stakhanova, N. and Ghorbani, A.A. (2017)" [9]
User session, packet pattern, header information, time interval, web user feature	"Odusami, M, Misra, S, Abayomi-Alli, O, Abayomi-Alli, A, Fernandez-Sanz, L. (2020)" [[10]
Duration, number of requests, no of unique Ips, Average no of requests	"Bhandari, A., Sangal, A. L., & Kumar, K. (2016)" [11]
Incomplete HTTP requests	"Hong, K., Kim, Y., Choi, H., & Park, J. (2018)" [12]
CPU, RAM usage, response time	"Benzaid, C., Boukhalfa, M., & Taleb, T. (2020)" [13]
Source IP, Source port, destination IP,	"Yungaicela-Naula, N. M., Vargas-

destination port, protocol	Rosales, C., & Perez-Diaz, J. A. (2021)" [14]
IP address, Source Port Number	""Park, S., Kim, Y., Choi, H., Kyung, Y., & Park, J. (2021)"[15]
Number of requests, Source IP	"Wang, Y. C., & Ye, R. X. (2021)" [16]

5. Experimentation

As per the literature, the researchers either use other layer datasets or use synthetic datasets to validate the research of the application layer DDoS attacks. We have taken SDN dataset with HTTP DDoS flow and normal flow traces.

5.1 Data Pre-processing

For the selected dataset (InSDN), the sample file opened after running the code is shown in figure 1. The figure doesn't cover all columns and rows as it's a huge dataset. We have split the dataset as 80% for training and 20% for testing. The dataset is changed for any null value or not a number value using the below python instruction

```
# Changing option to use infinite as nan

pd.set_option('mode.use_inf_as_na', True)
```

We have demonstrated just a sample screenshot of fewer attributes. Figure 1 shows that the CSV file read has 79540 rows × 49 columns. Though we have taken a screen shot of a few visible values.

```
df = pd.read_csv('DDoS-Normal.csv', delimiter=',', low_memory=False)

# Creating filter

df_filter = df.isin([np.nan, np.inf, -np.inf])

# Masking df with the filter

df = df[~df_filter]
```

Out[1]:

	Protocol	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean	Fwd Pkt Len Std	...	Pkt Size Avg	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	6	2087666	1	2	0.0	0.0	0	0	0.0	0.00000	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
1	6	2419	1	1	0.0	0.0	0	0	0.0	0.00000	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
2	6	398	2	1	31.0	46.0	31	0	15.5	21.92031	...	41.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
3	6	3019	3	1	0.0	0.0	0	0	0.0	0.00000	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
4	6	12769672	2	3	0.0	0.0	0	0	0.0	0.00000	...	0.00	1401.0	0.0	1401.0	1401.0	10200000.0	0.0	10200000.0	10200000.0	Normal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
79543	17	3246	1	3	33.0	719.0	33	33	33.0	0.00000	...	196.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
79544	17	3003056	3	1	522.0	174.0	174	174	174.0	0.00000	...	217.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
79545	6	281	1	1	0.0	0.0	0	0	0.0	0.00000	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
79546	17	618	1	1	95.0	178.0	95	95	95.0	0.00000	...	225.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal
79547	17	61	1	1	95.0	178.0	95	95	95.0	0.00000	...	225.50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Normal

79540 rows × 49 columns

Figure 1: - Dataset Representation

The code above reads the dataset file DDoS-Normal.csv using a python programming language. Here 'df' represents the data frame. Then figure 2 shows the graph distribution of a few attributes. We have virtualized the

taken dataset using the analysis library of Python, Pandas. The attributes are presented graphically using the below python code: -

"plot Per Column Distribution(df1, 16, 4)"

It draws a distribution graph of 16 columns in histogram form. A histogram is a better way to represent the values across datasets through visualization. Histograms make groups of values and display a count of the data points whose values are in a particular bin. Here df represents the data frame of the the dataset. It shows the insights of a dataset at a high level. Here we create a random distribution of the the dataset. 16 is the value taken for some columns, and 4 is the number of attribute representations per row. We have shown a row with 4 columns in figure 2. The dataset shows traces of normal traffic, and HTTP flood attack traces. The column of labels, as shown in figure 2, distribution graph of a few sampled columns of dataset. The machine learning algorithm trains and tests such data for normal and attack traffic.

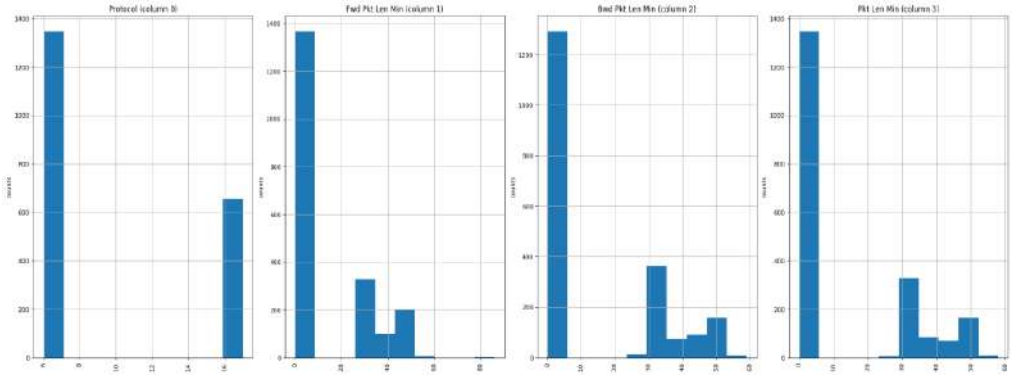

Figure 2: - Distribution graphs of sampled columns

These attributes have been presented through a correlation matrix using the below python command

"plotCorrelationMatrix(df, 19)"

The output of the above command is a graphical representation in figure 3 showing the attributes on X-axis and Y-axis presenting their correlation

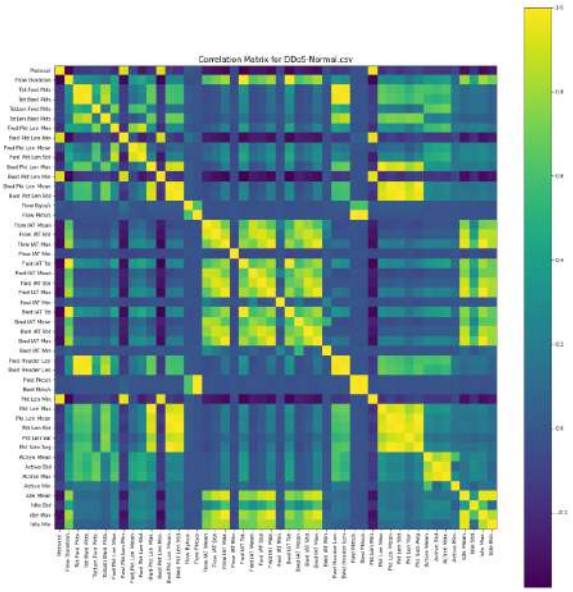

Figure 3: - Correlation matrix of attributes

It shows correlation coefficients between the attributes of the dataset. Each cell in figure 3 represents the correlation between two attributes and the value lies between -1 and 1. It gives the advanced analysis of variables and shows the extent to which variables are related to each other. It is virtualizing their association. If both variables increase and decrease together then it's a positive correlation. If an increase in one causes a decrease in another one then it's a negative correlation. Figure 4 is graph plotted to show the scatter matrix of the same.

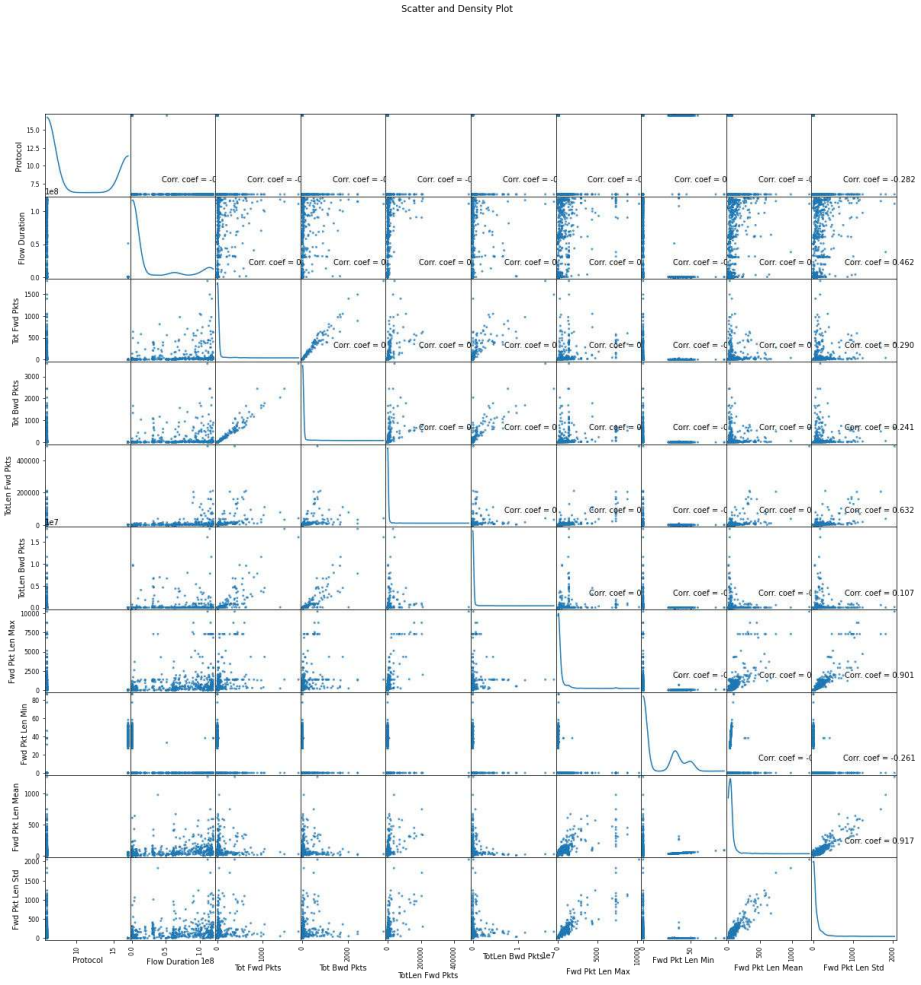

Figure 4: - Scatter matrix of attributes

This paper highlights the effects of using three feature extraction techniques (PCA, ICA, LDA) with three machine learning models Decision Tree, Random Forest, and Support Vector Machine. A publicly available SDN dataset for application layer DDoS attacks in SDN is used for experimentation. The dataset is passed an input to different feature extraction algorithms. The number of components or variance is provided as a parameter, and based upon those, the algorithms reduce the features and provide new dimensions. Thereafter the result and dimensions are passed to three ML models one by one and the accuracy of HTTP flood attack detection is compared for all three models.

For pre-processing the data, standard scaling is performed before doing feature extraction using

pipeline = Pipeline([

('min_max_scaler', MinMaxScaler()),

('std_scaler', StandardScaler())])

5.2 Feature Extraction Process

We have used InSDN dataset [18], which is publicly available for experimentation. This dataset has traces of HTTP flood DDoS attack flow and normal flow. There are 48 features related to SDN network which are collected by using ONOS controller by the researchers.

Using PCA

These features are given as input to the PCA feature extraction algorithm by providing a variance of 0.99 as parameter.

pca = PCA(0.99)

X_train_fit = pca.fit_transform(X_train)

X_test_fit = pca.transform(X_test)

Input of 48 features to PCA extracted 6 features after transformation as shown in figure 5.

```
Original number of features: 48
Reduced number of features: 6
```

	0	1	2	3	4	5
0	-2.061605e+07	-9.837254e+04	-6.150092e+05	-3.688066e+05	-2.513275e+05	-1.996456e+05
1	-2.061608e+07	-9.830160e+04	-6.156644e+05	-3.688212e+05	-2.513363e+05	-1.996230e+05
2	-2.061568e+07	-9.905892e+04	-6.060913e+05	-3.684957e+05	-2.511432e+05	-1.997404e+05
3	-2.056489e+07	-1.046498e+05	-3.790700e+05	-3.833345e+05	-2.481154e+05	-1.746720e+05
4	1.728718e+08	-7.251140e+07	-6.699606e+06	-2.983435e+05	-9.457062e+06	3.363246e+06
...	...	...	...	...	...	...
63627	-2.058337e+07	-1.139834e+05	-3.503096e+05	-3.753590e+05	-2.507520e+05	-1.875545e+05
63628	-1.914110e+07	1.179429e+05	-5.976546e+05	-1.087667e+06	-3.554221e+04	1.488685e+05
63629	-2.061569e+07	-9.921834e+04	-6.004103e+05	-3.685784e+05	-2.515668e+05	-1.990371e+05
63630	-2.061505e+07	-9.739550e+04	-6.324256e+05	-3.635851e+05	-2.495050e+05	-1.997489e+05
63631	-1.950103e+07	1.453296e+05	-6.568532e+05	2.986777e+05	9.257112e+04	2.679284e+05

63632 rows × 6 columns

Figure 5: - Principal Component Analysis

With the above the classifier Decision Tree (DT) is implemented which shows 99.6% accuracy in 0.85 sec. The resultant metrics of DT with PCA if shown below in figure 6(a) below. For the Random Forest (RF), the accuracy with PCA is 98.4% which took 2 sec for execution. The metrics is shown in figure 6(b). And the result of the Support Vector Machine (SVM) shows 38.7% accuracy in 4.3 sec. SVM shows the lowest accuracy in the highest time with PCA. The metrics is shown below in figure 6(c).

	precision	recall	f1-score	support
HTTP-flood	0.99	0.99	0.99	4158
Normal	1.00	1.00	1.00	11750
accuracy			1.00	15908
macro avg	1.00	1.00	1.00	15908
weighted avg	1.00	1.00	1.00	15908

Figure 6(a): - Principal Component Analysis with DT

	precision	recall	f1-score	support
HTTP-flood	0.97	0.97	0.97	4158
Normal	0.99	0.99	0.99	11750
accuracy			0.98	15908
macro avg	0.98	0.98	0.98	15908
weighted avg	0.98	0.98	0.98	15908

Figure 6(b): - Principal Component Analysis with RF

	precision	recall	f1-score	support
HTTP-flood	0.30	1.00	0.46	4158
Normal	1.00	0.17	0.29	11750
accuracy			0.39	15908
macro avg	0.65	0.59	0.38	15908
weighted avg	0.82	0.39	0.34	15908

Figure 6(c): - Principal Component Analysis with RF

Using ICA

We used ICA algorithm for 48 features which extracted 10 columns as the n_components passed to the algorithm was 10. Figure 7 below shows the extracted feature dataframe.

ica = FastICA(n_components=10)

X_train_fit = ica.fit_transform(X_train)

X_test_fit = ica.fit_transform(X_test)

Original number of features: 48
Reduced number of features: 10

	0	1	2	3	4	5	6	7	8	9
0	-0.000657	0.000526	0.000400	-0.000708	0.000265	-0.000590	-0.000367	0.000413	-0.000657	0.000196
1	-0.000657	0.000526	0.000400	-0.000708	0.000265	-0.000590	-0.000367	0.000413	-0.000657	0.000196
2	-0.000657	0.000527	0.000400	-0.000709	0.000262	-0.000586	-0.000367	0.000410	-0.000657	0.000192
3	-0.000653	0.000545	0.000404	-0.000706	0.000186	-0.000507	-0.000365	0.000411	-0.000660	0.000152
4	-0.003124	-0.000447	-0.002871	0.001741	0.000368	-0.001180	0.001511	0.000844	0.022684	-0.002392
...	...	...	...	...	...	...	...	...	...	...
63627	-0.000656	0.000535	0.000402	-0.000701	0.000176	-0.000555	-0.000366	0.000416	-0.000658	0.000198
63628	-0.000541	0.000702	0.000379	-0.000473	0.000276	0.001213	-0.000203	0.000904	-0.000703	0.000357
63629	-0.000657	0.000526	0.000401	-0.000708	0.000260	-0.000593	-0.000366	0.000414	-0.000657	0.000195
63630	-0.000656	0.000529	0.000398	-0.000709	0.000271	-0.000588	-0.000367	0.000405	-0.000657	0.000190
63631	-0.000397	0.001097	0.000356	-0.000217	0.000260	-0.000450	-0.000170	-0.000133	-0.000541	0.000398

63632 rows × 10 columns

Figure 7: - Independent Component Analysis

Then the results of all three classifiers were found using the ICA feature set. We have presented the metrics in figures 8(a), 8(b), and 8(c) below. We found that by implementing ICA the accuracy of the DT classifier is 73.3% in 8.5 sec of time. The accuracy of RF comes to be 73.1 in 2.25 sec. and in the case of SVM the accuracy was 73.8 in 3.4 sec of time. Thus, we can see that three of the classifiers approximately give same result. But SVM takes longer execution time then other models.

	precision	recall	f1-score	support
HTTP-flood	0.00	0.00	0.00	4158
Normal	0.74	1.00	0.85	11750
accuracy			0.74	15908
macro avg	0.37	0.50	0.42	15908
weighted avg	0.55	0.74	0.63	15908

Figure 8(a): - Independent Component Analysis with DT

	precision	recall	f1-score	support
HTTP-flood	0.03	0.00	0.00	4158
Normal	0.74	0.99	0.85	11750
accuracy			0.73	15908
macro avg	0.38	0.50	0.42	15908
weighted avg	0.55	0.73	0.62	15908

Figure 8(b): - Independent Component Analysis with RF

	precision	recall	f1-score	support
HTTP-flood	0.00	0.00	0.00	4158
Normal	0.74	1.00	0.85	11750
accuracy			0.74	15908
macro avg	0.37	0.50	0.42	15908
weighted avg	0.55	0.74	0.63	15908

Figure 8(c): - Independent Component Analysis with SVM

Using LDA

Lastly, we used the LDA algorithm for feature extraction before using ML models. We haven't provided component numbers to the algorithm. Thus, it extracted 1 out of 48 features as result.

lda = LinearDiscriminantAnalysis()

X_train_fit = lda.fit(X_train, Y_train).transform(X_train)

X_test_fit = lda.fit(X_test, Y_test).transform(X_test)

Figure 9 below shows the dataset of extracted features.

```
Original number of features: 48
Reduced number of features: 1
```

	0
0	1.276267
1	1.274144
2	0.658801
3	-2.741739
4	1.343669
...	...
63627	-1.943495
63628	-0.813040
63629	1.172361
63630	-0.678707
63631	-0.844024

63632 rows × 1 columns

Figure 9: - Linear Discriminant Analysis

In figures 10(a), 10(b), and 10(c) the result values of the LDA feature set with ML models are presented. We found the accuracy and execution time of DT, RF, and SVM. The accuracy of DT with LDA is 91.0% executed in 7.1 sec. The RF execution results in an accuracy of 93.4 % in 1.6 sec and SVM gives an accuracy of 93.4% in 1.4

sec. We can compare the results which show that DT gives little less accuracy than RF and SVM. Also, the execution time taken by the RF and SVM models are less as compared to DT.

	precision	recall	f1-score	support
HTTP-flood	0.88	0.76	0.82	4158
Normal	0.92	0.96	0.94	11750
accuracy			0.91	15908
macro avg	0.90	0.86	0.88	15908
weighted avg	0.91	0.91	0.91	15908

Figure 10(a): - Linear Discriminant Analysis with DT

	precision	recall	f1-score	support
HTTP-flood	0.89	0.85	0.87	4158
Normal	0.95	0.96	0.96	11750
accuracy			0.93	15908
macro avg	0.92	0.91	0.91	15908
weighted avg	0.93	0.93	0.93	15908

Figure 10(b): - Linear Discriminant Analysis with DT

	precision	recall	f1-score	support
HTTP-flood	0.93	0.81	0.87	4158
Normal	0.94	0.98	0.96	11750
accuracy			0.93	15908
macro avg	0.93	0.90	0.91	15908
weighted avg	0.93	0.93	0.93	15908

Figure 10(c): - Linear Discriminant Analysis with DT

From the results presents in all above metrices we have compared the results of all three models using PCA, ICA and LDA. The comparison is presented below in table 3.

Table 3: Comparison of results of DT, RF, and SVM using PCA, ICA, and LDA

		Decision tree		**Random Forest**		**Support Vector Machine**	
	Features Extracted	**Accuracy**	**Time Taken**	**Accuracy**	**Time Taken**	**Accuracy**	**Time Taken**
PCA	**6**	99.6	0.85	98.4	2	38.7	4.3
ICA	**10**	73.3	8.5	73.1	2.25	73.8	3.4
LDA	**1**	91	7.1	93.4	1.6	93.4	1.4

The above table shows that maximum accuracy was found 99.6 using the PCA technique with a Decision Tree classifier. Secondly, the accuracy near to that was given by Random Forest as 98.4 using PCA. So, from above we can say that using the PCA technique for DT and RF gives the best results. And, using LDA with all three models gives a result above 90%. We must use the feature extraction technique as per the ML model we have chosen to detect DDoS attack.

6. CONCLUSION

This work has identified the attributes of user behavior for detecting DDoS attack traffic at the application layer of legacy and Software Defined Networking (SDN). We have taken a Software Defined Network dataset "InSDN" to show a set of features of HTTP flood DDoS attack. The features used by various researchers in literature, based on the browsing behavior of the user and attacker are identified. To show the effectiveness of features for the detection mechanism, we have used techniques such as PCA, ICA, and LDA. These techniques reduce the number of features and provide reduced dimensions as a new dataset. We have created the histogram graph and correlation chart of the attributes, followed by the scatter graph as part of data pre-processing. We used the reduced dataset to show the attack prediction through the Decision Tree, Random Forest, and Support Vector Machine algorithm. It predicted a maximum accuracy of 99.6% through experimentation. The decision tree with features reduced through PCA has provided the highest prediction accuracy. The comparison table is also included for all three classifiers with each feature extraction technique. Also, this study aims to highlight the similarities/ dissimilarities through various features of application layer DDoS attacks and normal traffic. In the future, we will use the related features in our further work to detect application layer DDoS attacks in Software Defined Networking environments using MININET emulator.

Declaration

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interest Sarabjeet Kaur, Amanpreet Kaur Sandhu, Abhinav Bhandari declare that they have no conflict of interest

Authors' contributions

All authors of this research paper have directly participated in the planning and study selection process. The research work was conducted by Sarabjeet Kaur under the supervision of Dr. Amanpreet Kaur Sandhu, Associate Professor, University Institute of Computing, Chandigarh University, Gharuan and Dr. Abhinav Bhandari, Assistant Professor, Department of Computer Engineering, Punjabi University, Patiala.

Funding Not Applicable

Data Availability "InSDN" dataset is publicly available dataset

Code Availability Not Applicable

7. REFERENCES

[1] S. Indraneel, V. Praveen Kumar Vuppala, HTTP Flood attack Detection in Application Layer using Machine learning metrics and Bio inspired Bat algorithm, Appl. Comput. Informatics. (2017). https://doi.org/10.1016/j.aci.2017.10.003.

[2] R. Saravanan, S. Shanmuganathan, Y. Palanichamy, Behavior-based detection of application layer distributed denial of service attacks during ash events, Turkish J. Electr. Eng. Comput. Sci. 24 (2016) 510–523. https://doi.org/10.3906/elk-1308-188.

[3] K. Singh, P. Singh, K. Kumar, Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges, Comput. Secur. 65 (2017) 344–372. https://doi.org/10.1016/j.cose.2016.10.005.

[4] K. Singh, P. Singh, K. Kumar, User behavior analytics-based classification of application layer HTTP-GET flood attacks, J. Netw. Comput. Appl. 112 (2018) 97–114. https://doi.org/10.1016/j.jnca.2018.03.030.

[5] K. Hong, Y. Kim, H. Choi, J. Park, SDN-Assisted Slow HTTP DDoS Attack Defense Method, IEEE Commun. Lett. 22 (2018) 688–691. https://doi.org/10.1109/LCOMM.2017.2766636.

[6] S. Daneshgadeh, T. Kemmerich, T. Ahmed, N. Baykal, An Empirical Investigation of DDoS and Flash Event Detection Using Shannon Entropy, KOAD and SVM Combined, 2019 Int. Conf. Comput. Netw. Commun. ICNC 2019. (2019) 658–662. https://doi.org/10.1109/ICCNC.2019.8685632.

[7] S. Bravo, D. Mauricio, Distributed denial of service attack detection in application layer based on user behavior, Webology. 15 (2018) 38–53.

[8] Jaafar, G. A., Abdullah, S. M., & Ismail, S. Review of Recent Detection Methods for HTTP DDoS Attack. Journal of Computer Networks and Communications, (2019), 1–10.doi:10.1155/2019/1283472

[9] C. Benzaid, M. Boukhalfa, T. Taleb, Robust Self-Protection Against Application-Layer (D)DoS Attacks in SDN Environment, IEEE Wirel. Commun. Netw. Conf. WCNC. 2020-May (2020). https://doi.org/10.1109/WCNC45663.2020.9120472.

[10] B. Singh, K. Kumar, A. Bhandari, Simulation study of AL-DDoS attack, Proc. 2015 Int. Conf. Green Comput. Internet Things, ICGCIoT 2015. (2016) 893–898. https://doi.org/10.1109/ICGCIoT.2015.7380589.

[11] H.H. Jazi, H. Gonzalez, N. Stakhanova, A.A. Ghorbani, Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling, Comput. Networks. 121 (2017) 25–36. https://doi.org/10.1016/j.comnet.2017.03.018.

[12] M. Odusami, S. Misra, O. Abayomi-Alli, A. Abayomi-Alli, L. Fernandez-Sanz, A survey and meta-analysis of application-layer distributed denial-of-service attack, Int. J. Commun. Syst. 33 (2020) 1–24. https://doi.org/10.1002/dac.4603.

[13] Bhandari, A., Sangal, A. L., & Kumar, K. (2016) Characterizing flash events and distributed denial-of-service attacks: an empirical investigation. Security and Communication Networks, (2016), n/a–n/a.doi:10.1002/sec.1472.

[14] Yungaicela-Naula, N. M., Vargas-Rosales, C., & Perez-Diaz, J. A. (2021). SDN-based architecture for transport and AL-DDoS attack detection by using machine and deep learning. IEEE Access, 9, 108495–108512. https://doi.org/10.1109/ACCESS.2021.3101650

[15] Park, S., Kim, Y., Choi, H., Kyung, Y., & Park, J. (2021). HTTP DDoS flooding attack mitigation in software-defined networking. IEICE Transactions on Information and Systems, E104D(9), 1496–1499. https://doi.org/10.1587/transinf.2021EDL8022

[16] Wang, Y. C., & Ye, R. X. (2021). Credibility-Based Countermeasure against Slow HTTP DoS Attacks by Using SDN. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021, 890–895. https://doi.org/10.1109/CCWC51732.2021.9375911

[17] Kwak, N., Choi, C.-H., & Choi, J. Y. (2001). Feature Extraction Using ICA. Lecture Notes in Computer Science, 568–573. doi:10.1007/3-540-44668-0_80

[18] Elsayed, M. S., Le-Khac, N.-A., & Jurcut, A. D. (2020). InSDN: A Novel SDN Intrusion Dataset. IEEE Access, 8, 165263–165284. doi:10.1109/access.2020.3022633

[19] Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M.R., & Portmann, M. (2021). Feature Extraction for Machine Learning-based Intrusion Detection in IoT Networks.

[20] A. Praseed and P. S. Thilagam, "DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications," in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 661-685, Firstquarter 2019, doi: 10.1109/COMST.2018.2870658.

[21] Rustam, Furqan & Mushtaq, Muhammad & Hamza, Ameer & Farooq, Shoaib & Jurcut, Anca & Ashraf, Imran. (2022). Denial of Service Attack Classification Using Machine Learning with Multi-Features. Electronics. 11. 3817. 10.3390/electronics11223817.

[22] Ogundokun, R.O., Misra, S., Bajeh, A.O., Okoro, U.O., Ahuja, R. (2022). An Integrated IDS Using ICA-Based Feature Selection and SVM Classification Method. In: Misra, S., Arumugam, C. (eds) Illumination of Artificial Intelligence in Cybersecurity and Forensics. Lecture Notes on Data Engineering and Communications Technologies, vol 109. Springer, Cham. https://doi.org/10.1007/978-3-030-93453-8_11

[23] Jaber, A. (2023). Model for Preventing DDoS Attacks Using a Hypervisor. In: Barolli, L. (eds) Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2022. Lecture Notes in Networks and Systems, vol 571. Springer, Cham. https://doi.org/10.1007/978-3-031-19945-5_7

