Intrusion Detection System based on the SDN Network, Bloom Filter and Machine Learning

Traore Issa[1]
Institute of Mathematics research (IMAR)
Computer Science Laboratory Telecom Networks
Felix Houphouet-Boigny University
08 BP 2035 Abidjan 08, Cote d'Ivoire

Kone Tiemoman[2]
Virtual University of Cote d'Ivoire
Computer Science Laboratory Telecom Networks
28 BP 536 28 Abidjan
Cote d'Ivoire

***Abstract*—The scale and frequency of sophisticated attacks through denial of distributed service (DDoS) are still growing. The urgency is required because with the new emerging paradigms of the Internet of Things (IoT) and Cloud Computing, billions of unsecured connected objects will be available. This document deals with the detection, and correction of DDoS attacks based on real-time behavioral analysis of traffic. This method is based on Software Defined Network (SDN) technologies, Bloom filter and automatic behaviour learning. Indeed, distributed denial of service attacks (DDoS) are difficult to detect in real time. In particular, it concerns the distinction between legitimate and illegitimate packages. Our approach outlines a supervised classification method based on Machine Learning that identifies malicious and normal packets. Thus, we design and implement Defined (IDS) with a great precision. The results of the evaluation suggest that our proposal is timely and detects several abnormal DDoS-based cyber-attack behaviours.**

***Keywords*—Distributed denial of service; intrusion detection software; software defined network; machine learning; synchronize; acknowledgment; bloom filter**

I. Introduction

Over the past decade, DDoS attacks have been a powerful threat to the security of many Internet service providers, and have resulted in economic losses for them. DoS attacks cause a denial of service to legitimate requests by depleting network resources and services. To maximize impact, the attack will be launched from distributed sources, called attacks through denial of distributed service. In most cases, these attacks are launched by botnets. The largest DDoS attack on the latest records occurred in February 2018 as revealed by the Git Hub. The attack came from more than thousand different European Union countries out of tens of thousands of single endpoints. This was the one amplification attack using Memcached technology that peaked at 1.35Tbps. Another major DDoS attack is the Mirai [1] botnet attack that was used in a high volumetric DDoS of about 1.1 Tbps that destroyed a large part of Dyn's database in October 2016. Mirai has successfully ordered nearly 100,000 robots by exploiting the low security of cameras, home routers, digital recorders and printers with default credentials used for their telnet ports.

Many methods are used to block DDoS attacks, including some:

- The signature-based approach: it requires an a priori knowledge of the elements related to the signature of attacks, see SNORT [2]. Signatures are manually built by security experts. The authors of [3] analyze previous attacks to look for a match with incoming traffic to detect intrusions. Signature-based techniques are only effective in detecting the traffic of known DDoS attacks; while new attacks or even slight variations of old attack go unnoticed.
- Anomaly-based detection: the anomaly-based system uses a different method. It treats any network connection that violates the normal profile as an anomaly. The anomaly is revealed if incoming traffic deviates significantly from normal profiles, see [4] and [5]. To detect DDoS attacks, it is first necessary to know the overall normal behaviour of the system traffic and then to find deviations from this behaviour. The anomaly-based technique can detect new attacks. However, it can initiate many false alarms.
- Packet filtering: packets entering and leaving the network protect the network against attacks from any source. This technique uses server firewalls, router based packet filtering [6]. This requires the installation of filter input and output packets on all routers. It is used to filter the spoofed IP address, but approaches to prevent it need a global implementation that is not practical [7].

In this article, we set up IDS capable of detecting anomalies based on Machine Learning techniques. The volume of data to be studied is enormous, so we use SDN technology for efficient data processing. We also used the Bloom filter, which is a probabilistic structure for storing and accessing data efficiently. This document is structured as follows: Section 2 describes some approaches used to solve DDoS attack problems. Section 3 outlines our method of resolution and then Section 4 illustrates the results and discussion. Finally, the conclusion is presented in Section 5.

II. Related Work

An attack through denial of Distributed Service (DDoS) is a flood attack using several controlled sources, called Botnets or Zombies, to disable a service and prevent legitimate users from using it.

A. How a SYN Flood Attack Works

SYN flood attacks work by exploiting the process of establishing a TCP connection. Under normal conditions, the TCP connection has three distinct processes for establishing a connection.

- First, the client sends a SYN packet to the server to establish the connection.
- The server then responds to this initial packet with a SYN / ACK packet, in order to acknowledge receipt of the communication.
- Finally, the client sends back an ACK packet to acknowledge receipt of the packet from the server. After completing this sequence of sending and receiving packets, the TCP connection is open and capable of sending and receiving data.

To create a denial of service, an attacker exploits the fact that after receiving an initial SYN packet, the server responds with one or more SYN / ACK packets and waits for the last step of making contact, see Fig. 1.

- The attacker sends a high volume of SYN packets to the target server, often with spoofed IP addresses.
- The server then responds to each connection request and leaves an open port ready to receive the answer.
- The server waits for the last ACK packet, which never arrives, the attacker continues to send more SYN packets. The arrival of each new SYN packet forces the server to temporarily maintain a new port connection open for a period of time. Once all available ports have been used, the server can no longer operate normally.

By repeatedly sending SYN initial connection request packets, the attacker is able to overwhelm all available ports on a target server computer, resulting in the target device responding to legitimate traffic slowly or not at all.

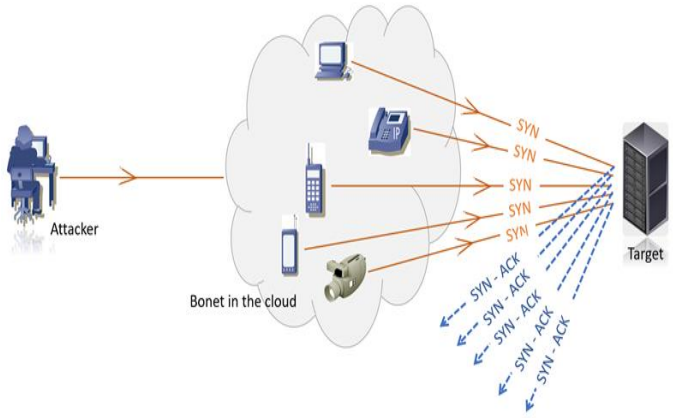

Fig. 1. DDOS Attacks Architecture.

B. Defensive Mechanisms

The DDoS defence mechanisms [7] and [8] are classified into two of the four main categories: source identification, attack detection, reactivity and attack prevention.

- The identification of the attack source uses mechanisms to find the source IP address to block them. The trace back investigation [9] is the most popular mechanism to identify the attacker's source IP address.
- Attack detection detects the DDoS attack when it occurs. Some defence mechanisms are MULTOPS [10] and anomaly-based detection [11].
- Reactivity to an attack aims to reduce or eliminate the effect of the attack [12]. Two main approaches [13] are taken to respond to DDoS attacks and network resource management.
- Attack prevention tries to stop the attack before it occurs. The attack does not reach the target host. Some examples are Ingress/Egress 6 filters on routers, packet filtering on routers [14], and automatic learning to detect anomalies.

The first three methods have proven their effectiveness, but they are reactive, the damage has already been done. The latter approach is proactive. It has proven its usefulness. However, it requires a lot of calculation, large data to store and managed. Our approach is proactive, because the objective is to ensure quality of service without it being blocked by a DDoS attack. Thus, this paper proposes a new IDS network paradigm based on Machine Learning to solve the network control problem. The main contribution of this article can be summarized as follows:

- Use SDN [15], automatic learning and Bloom filter [16] to set up a high-performance network and effective real-time security,
- Provide a DDoS attack detection architecture by leveraging incoming flow monitoring capability to filter traffic and establish legitimate TCP connections.

Implement a proactive IDS capable of automatically making decisions related to several behavioural parameters. These decisions are based on the set of rules predefined by the administrator.

To do this, we ensure on the one hand, the storage of IP packet information in a compact way in the address intended for this purpose, and on the other hand, the calculation of automatic learning on dedicated servers in an architecture combining SDN and Machine Learning.

III. Methodology

A DDoS attack caused by botnets generates a lot of resources; a traditional router can hardly predict the attack. The router performs calculations to route packets, assigns priorities, makes routing decisions, and enforces rules specified by the administrator. Thus, it can only be changed manually by the administrator, which obviously takes time and does not lend itself to rapid context changes. With the SDN, these changes are automated and even programmable.

A. SDN Architecture

The data plan and the control plan are increased tenfold. Thus, the administrator defines the rules in the controller, and they are instantly transmitted in the network equipment.

Fig. 2 illustrates the SDN architecture, which consists of three layers. The lowest layer is the infrastructure layer, also called the data plan. It includes the elements of the transfer network. The responsibilities of the routing plan are mainly data transfer, as well as monitoring of local data transmission, information and statistical collection.

The layer above is the control layer, also called the control plan. He is responsible for the programming and management of the routing plan. To this end, it uses the information provided by the transmission plan and defines the operation and routing of the network. It includes one or more controllers that communicate with the elements of the transmission network through standardized interfaces, known as southbound interfaces.

The application layer contains network applications that can introduce new network functionality through APIs, such as security and management, transfer schemes or control layer support in network configuration. It has an abstract and global view of the network from the controllers and uses this information to provide appropriate advice to the control layer. The interface between the application layer and the control is called the northward interface.

Northbound APIs can be used to facilitate innovation and enable efficient network orchestration and automation to align with the needs of different applications through SDN network programmability. We will use this property of the application layer to implement a TCP flooding attack detection module.

B. Bloom Filter: Data Storage

A Bloom filter is a probabilistic structure that allows the efficient storage of a set of elements [16]. It consists of a vector of m bits and a set of k hash functions. Initially all bits are at 0. To insert an element into a filter, the k hash functions are calculated on it and their results determine the positions of the bits set to 1. To test if an element belongs to a filter, simply calculate its k hash functions on the element and check if all the bits at the corresponding positions are at 1. If not, it is certain that the element is not in the filter.

However, there is still a probability of false positives: it is possible that all the corresponding bits have been set to 1 by other stored elements, and therefore to detect a tested element when it is not in the filter. The probability of false positives as a function of the number of elements stored n and the size of the filter is given by the formula:

For any pair of integers (m, k) :

$$P \approx (1 - e^{-\frac{km}{n}})^k \quad (1)$$

To maintain the same false positive rate with an increasing number of elements, it is necessary to increase the number of bits and hash functions, which results in higher memory consumption and increased computing costs.

The Bloom filter stores in the form of a table of bits that represent the IP addresses considered malicious, see Fig. 3.

Consider $\mathcal{F} = \{ip_1, ip_2, ..., ip_n\}$, the n IP addresses that describe the array of n bits. Initially all bits are at 0.

Let $\mathcal{H} = \{h_1, h_2, ..., h_k\}$, all the independent hash functions stored in p.

For each ip_x on $\mathcal{F}$:

$$h_j(ip_x) = 1 \text{ for } 1 \leqslant j \leqslant k. \quad (2)$$

To check if an attack suspect ip_x address is in $\mathcal{F}$. We check that all $h_j(ip_x) = 1$, otherwise $h(ip_x) = 0$ is not malicious. This process can generate false positives. In other words, it can happen that for an ip_x address we have $h_j(ip_x) = 1$ while it is not malicious.

In our approach, false positives are negligible because the probability of their existence is low. Indeed, let us consider m, the size of the Bloom filter, n the number of hash functions. Let X be a random variable representing all the bits. Thus, the false positive rate can be evaluated by:

$$P(X = 0) = (1 - \frac{1}{m})^{nk} \quad (3)$$

In [17] have shown that this rate is very low because:

$$k = \ln(2)\frac{p}{n} \quad (4)$$

When k=10 and p=20n, the probability of a false positive is 0.0000889. This result justifies the use of the Bloom filter in the detection of DDoS attacks based IDS architecture.

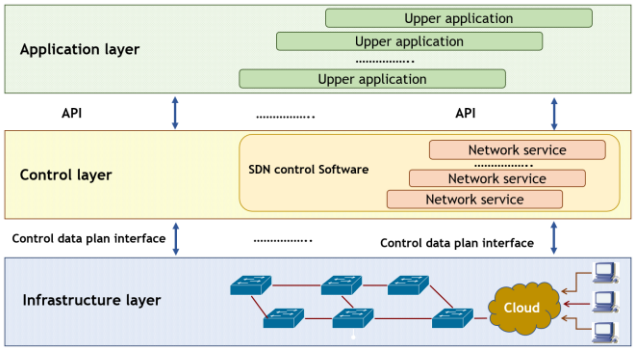

Fig. 2. Architecture SDN.

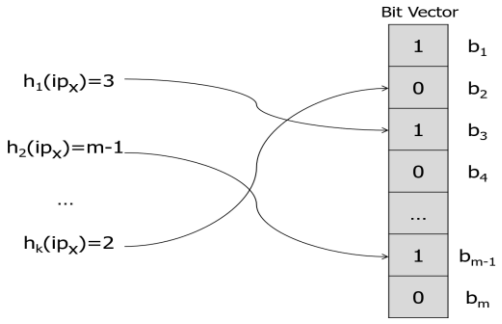

Fig. 3. IP Address Hash Functions.

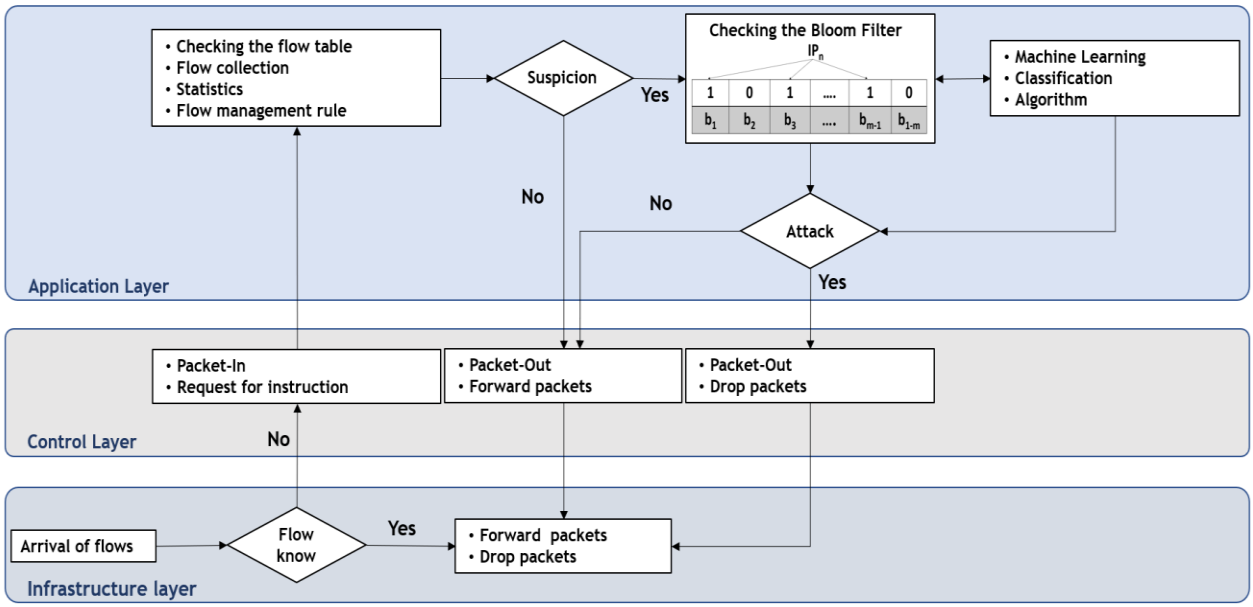

Fig. 4. IDS Architecture for DDOS Attack Detection.

In our approach, we have combined the advantages of the behavioural filter and Machine Learning as shown in Fig. 4.

In our method, network traffic must be collected from switches and then used to build the drive and classification set. The management of packets entering the network is presented in Fig. 4. When a new packet arrives at the switch, if it belongs to an existing flow in the flow table, it updates the flow statistics otherwise, a "Packet-In" message is sent to the Openflow controller. The controller responds with a "Packet-out" message on the attitude to be followed according to the pre-established rules.

Switches in the data plan uses tables to route packets. This is possible by using entries in flow tables and a packet processing process. According to [17] an entry in the flow table consists of seven fields: Match Fields, Priority, Counters, Instructions, Timeouts, Cookie, Flags.

The Counters field allows you to know the total number of packets processed for an entry. Counters can be maintained for each flow table, number of packets or bytes, flow entry, port, queue, duration during which the entry was activated.

C. ADIS: DDoS Attack Detection Module

In this section, we describe the mechanism for detecting and preventing attacks.

1) DDoS defense architecture

a) The data plan: When a new packet arrives at the switch, it checks whether the packet header matches an entry in its flow table.

If it finds an entry, it processes the packet as defined in the corresponding entry. Otherwise, he forwards the packet to the controller in order to receive instructions after a thorough investigation.

b) The control plan: After receiving a new packet, the controller processes, calculates and creates a new flow entry, which it then sends to the switch. The switch receives the message from the controller, adds the new entry to its flow table, and manages the packet as defined in the entry [18]. When the packet is unknown to the controller, the Openflow protocol sends the packet header to the Identity Attack and Storage Detection (ADIS) module in the application plan.

c) The application plan: The ADIS module of the application layer is designed to analyze the SDN network flow tables and collect traffic flows by inspecting the IP header {src_ip, src_port, dst_ip, dst_port, protocol}. Each flow can be represented by a set of statistical characteristics, such as DurationSeconds, packetCount and byteCount, etc. The ADIS module checks if the IP address of the packet is stored in the Bloom filter (attacker database). Failing this, a deep analysis based on the number of packets sent per second by the source to the Openflow switch classifies the category of the source IP address. In the following section, we propose a classification algorithm.

2) Data classification algorithm: In order to detect the DDoS attack, the IDS must be supplied with traffic information related to the following parameters: src_ip, src_port, dst_ip, dst_ip, dst_port, protocol, DurationSeconds, packetCount and byteCount, etc. It uses a Machine Learning (ML) classification model to detect attack activity. In our example, we will use the following models : linear discriminant analysis (LDA), k-nearest neighbors (KNN) and Support vector machine (SVM).

These models can learn the pattern with few training samples and produce an accurate classification by reducing false positives.

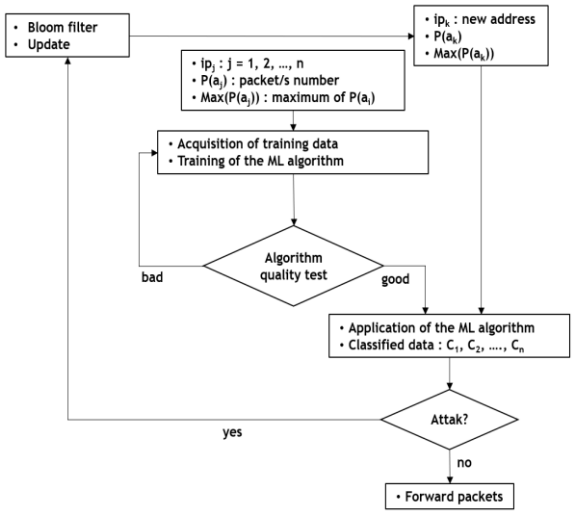

Fig. 5. Classification Algorithm.

Using the recognized model in Fig. 5, the IDS perform category prediction for new unknown traffic samples. The classification results for the test data points would be either normal or attack.

This security approach integrates a cognitive layer above the control layer and thus allows artificial/automated intelligence to be naturally introduced into network management. In this context, IDN (Intelligence Defined Networks) approaches are presented as an evolution of the SDN.

D. Implémentation

In this section, we present our experimental design and describe the experimental results of our methods. Then we compare the models against the error rates as defined above.

1) Simulation environment: To evaluate our approach, we chose the Kali-Linux simulator. The official version of Kali-Linux has several modules to simulate hacking and computer attacks. In our experience, we install hping3 which is a package available by default on Kali-Linux. The packet flow is captured by Wireshark and Tshark for analysis on the I/O graph. This data is collected at the SDN Openflow switches for analysis.

2) Data collection and analysis: Incoming packets are captured by Wireshare, two processes are performed. The first consists in performing an analysis of SYN, SYN-ACK and ACK exchanges. The second based on automatic learning allows a classification of the captured data into clusters classified according to the analysis.

In Section IV, we use R Studio software to classify the data. We use information on initiated connections: source and destination addresses, protocols, connection time, packet size, information on SYN, ACK, sequence numbers, etc.

The purpose is to classify the addresses that have initiated a SYN connection according to ACK responses or not. Indeed, the imbalance of flows can facilitate the detection of DDoS. In a normal packet flow, the number of incoming packets corresponds to the number of outgoing packets over a given period of time. For example, each packet in TCP connection is normally acknowledged. However, during the attack the number of incoming and outgoing packets is unbalanced. The second treatment makes it possible to update the addresses considered malicious in the Bloom filter according to Fig. 3.

IV. Results and Discussions

The experimental results produced the following Fig. 6 shows the curve representing the number of packets sent per second to the server.

Fig. 7 shows a point cloud of the number of packets received per second. We will use this graph to classify the data into three classes. One class of packets is considered normal, another is considered suspicious and the last one is considered malicious.

The Machine Learning method makes it possible to dissociate normal, suspicious and attacking IP addresses. On Fig. 8, Fig. 9 and Fig. 10 below, we can see the LDA, KNN and SVM methods resulting from the relationships between training data and test files. The SVM is the approach that achieves the best results.

In our approach, learning will be performed several times on 75% of the original classification data set. Training performance is given by train: mmce. For each iteration, the formed model will be tested on a subset of training (75%) and a subset of tests (25% of the original data set).

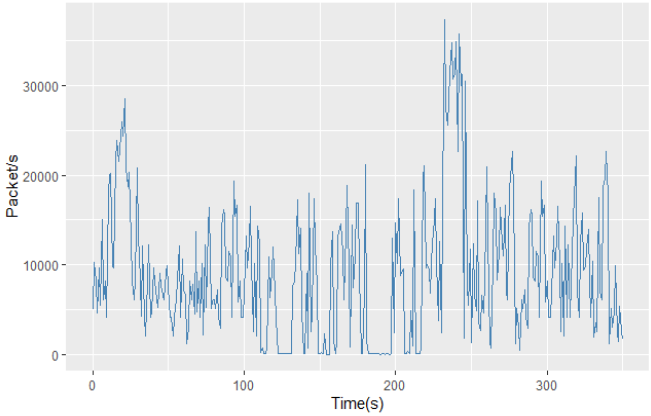

Fig. 6. Number of Source Packets / s.

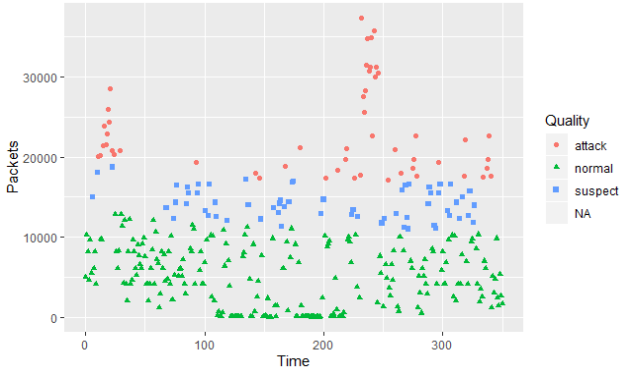

Fig. 7. Point Cloud of Source Packets per Second.

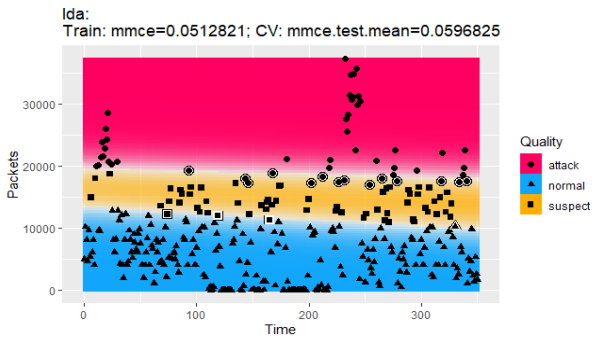

Fig. 8. LDA Classification.

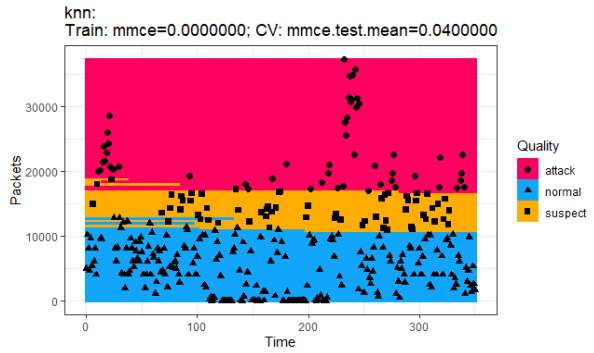

Fig. 9. KNN Classification.

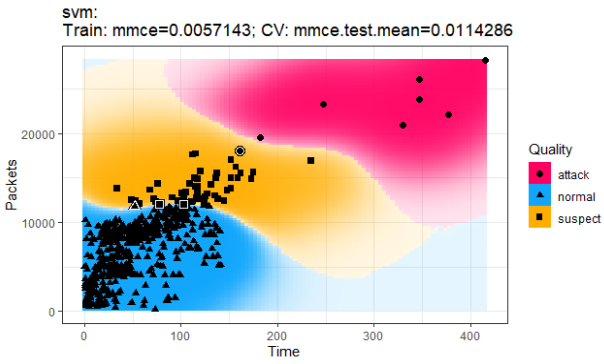

Fig. 10. SVM Classification.

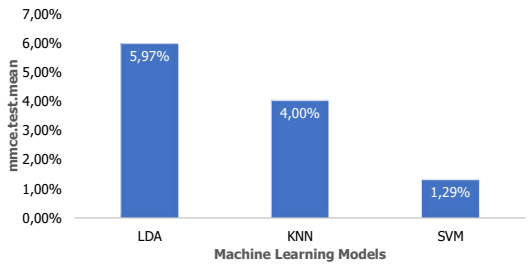

Fig. 11. Average Model Error.

The performance of the model will be measured by the average misclassification error (mmce.test.mean) of each model LDA, KNN and SVM, see Fig. 11.

Thus, the classification methods exposed made it possible to simulate the requests sent to a server. Our method manages to classify the addresses from the packets involved with an acceptable error rate. Indeed, according to Fig. 11, the error rate in classification is 1.29%. Thus, our proposal allows classifying the IP addresses of packets resulting from DDoS attacks and normal packets.

V. Conclusion and Future Work

In the age of large data, with the exponential growth of network traffic, network attacks are becoming more diversified and sophisticated. In this document, we use the SDN architecture and Bloom filter to ensure the computing power of Openflow controllers, storage and data access. The Machine Learning algorithm allowed the IDS to detect and suppress DDoS attack traffic. We focused on analyzing the data in order to avoid false positives as much as possible. Thus, the network application classifier based on the SVM learning model allows the expected objectives to be achieved with greater precision.

As part of our future work, we plan to extend our analysis to Machine Learning, in order to find an appropriate model to ensure a higher accuracy rate and eliminate false alarms.

Acknowledgment

The Publication of this research was supported by the Mathematics Research Institute.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, ‘‘DDoS in the IoT: Mirai and other botnets,’’ IEEE Computer, 50(7), (2017), p80-84. https://doi.org/10.1109/MC.2017.201.

[2] M. Roesch, ‘‘Snort-Lightweight Intrusion Detection for Networks,’’ In: Proceedings of the USENIX Systems Administration Conference (LISA November 1999), pp. 229–238.

[3] V. Paxson, ‘‘Bro: A System for Detecting Network Intruders in Real-Time,’’ International Journal of Computer and Telecommunication Networking 31(24), (1999), pp2435–2463.

[4] U. Dincalp, M. Serdar, O. Sevine, E. Bostanci, I. Askerzade, ‘‘Anomaly Based Distributed Denial of Service Attack Detection and Prevention with Machine Learning,’’ 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018.

[5] P. Ombase, P. Scholar, S. Bagade, N. Kulkarni, A. haisgawali, ‘‘DoS Attack Mitigation Using Rule Based and Anomaly Based Techniques in Software Defined Networking,’’ In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; pp. 469–475.

[6] K. Park, H. Lee, ‘‘On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets,’’ In: Proceedings of the ACM SIGCOMM 2001, Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press, New York (2001), pp 15–26.

[7] T. Peng, C. Leckie, K. Ramamohanarao, ‘‘Protection from distributed denial of service attack using history-based IP filtering,’’ In: Proceedings of IEEE International Conference on Communications (ICC 2003), Anchorage, AL, vol. 1, pp. 482–486.

[8] E. Fenil, P. Mohan Kumar , ‘‘Survey on DDoS defense mechanisms,’’ wiley, wileyonlinelibrary.com/journal/, Décembre 2018.

[9] V. Chidri, V. Balasubramani, S. Sadath Ali, S. Shrikrishna Hegde, P. Sadanand, ‘‘A Survey on Distributed Denial-of-service Attacks and Defense Mechanisms,’’ JETIR1504089 Journal of Emerging Technologies and Innovative Research (JETIR) www.jetir.org, avril 2015.

[10] T. Jog, M. Natu, S. Shelke, '' *Distributed capabilities-based DDoS defense,''* 2015 International Conference on Pervasive Computing (ICPC), janvier 2015, pp. 1–6.

[11] C. Buragohain, M. Kalita Santosh Singh, D. Bhattacharyya, ''Anomaly based DDoS Attack Detection,'' Chaitanya Buragohain, Manash Jyoti Kalita Santosh Singh, Dhruba K.Bhattacharyya, International Journal of Computer Applications (0975 –8887) Volume 123–No.17, August, 2015, pp35-40.

[12] A. Cardigliano, L.Deri et T. Lundstrom, ''*Commoditising DDoS mitigation,*'' septembre 2016, p. 523–5282016 International Wireless Communications and Mobile Computing Conference (IWCMC).

[13] N. Lu, S. Su, M. Jing, and J. Han, ''A router-based packet filtering scheme for defending against dos attacks. China Communications, 11(10, 2014), pp136–146.

[14] R. Koning, B. de Graaff, G. Polevoy, R. Meijer, C. de Laat, P. Grosso, ''Measuring the efficiency of SDN mitigations against attacks on computer infrastructures,'' Future Generation Computer Systems **91**(1), 144{156 (2019)., https://doi.org/10.1016/j.future.2018.08.011, https://doi.org/10.1016/j.future.2018.08.011.

[15] R. Patgiri, S. Nayak, and S. K. Borgohain, "Preventing DDoS using bloom filter: A survey," ICST Transactions on Scalable Information Systems, vol. 5, no. 19, Article ID 155865, 2018.

[16] C. Tseung, K. Chow, and X. Zhang. ''Anti-DDoS technique using self-learning bloom filter,'' In Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on, pages 204–204. IEEE, 2017.

[17] P. Cao, ''Bloom filters-the math,'' University of Wisconsin-Madisson, Madisson (1998). http://pages.cs.wisc.edu/~cao/papers/summary-cache/node8.html.

[18] Open Networking Foundation, ''OpenFlow Switch Specification,'' Version 1.5.1 (Protocol version 0x06), https://www.opennetworking.org /images//openflow-switch-v1.5.1.pdf.