Communication Papers of the Federated Conference on Computer Science and Information Systems pp. 57–63
DOI: 10.15439/2019F87
ISSN 2300-5963 ACSIS, Vol. 20

A Framework for Network Intrusion Detection using Network Programmability and Data Stream Clustering Machine Learning Algorithms

Admilson de Ribamar Lima Ribeiro[1] , Edward David Moreno Ordonez[2] and Anderson Clayton Alves Nascimento[3]
[1,2] *Computing Department, Federal University of Sergipe, UFS*
São Cristóvão, Brazil
email: admilson@ufs.br, edwdavid@gmail.com
[3]*Institute of Technology, University of Washington Tacoma*
Tacoma, Washington
email: andclay@uw.edu

***Abstract*— Several operational security mechanisms have been developed to mitigate malicious activity in the Internet. However, the most these mechanisms require a signature basis and present the inability to predict new malicious activity. Other anomaly-based mechanisms are inefficient due to the possibility of an attacker simulates legitimate traffic, which causes many false alarms. Thus, to overcome that problem, in this paper we present an anomaly-based framework that uses network programmability and machine learning algorithms over continuous data stream. Our approach overcomes the main challenges that occur when develop an anomaly-based system using machine learning techniques. We have done an experimental evaluation to demonstrate the feasibility of the proposed framework. In the experiments, we use a DDoS attack as network intrusion and we show that the technique attains an Accuracy of 98.98%, a Recall of 60%, a Precision of 60% and an FPR of 0.48% for 1% DDoS attack on the real normal traffic. This shows the effectiveness of our technique.**

***Index Terms*--Operational Security, DDoS, Machine Learning, Data Stream.**

I. Introduction

THE HUGE variety of attacks in the Internet combined with the emergence of the new environments such as smart homes has demanded the improvement of forms of defense. This issue is more serious by fact that traditional mechanism of security as cryptography is not adequate to be used due to the real-time nature of these new environments. That is, very strong cryptograph functions can slow down the system. Thus, detection intrusion mechanisms are more adequate to face this issue.

Several studies have analyzed the use of detection intrusion in computer network security [1-7]. However, all those solutions are based in signature, that is, there is a need of a signature basis of the intrusions so that they can be detected. Despite its speed in detecting certain attacks due to the knowledge of the signatures of the main attacks, this category of solutions presents a crucial limitation: they can only detect attacks that are compatible with previously available signatures, not acting in new ways of malicious code.

One way to bypass that limitation, it is addressing the problem through a solution based in anomaly where abnormal traffic can be detected considering the knowledge of normal activities through profiles. Thus, deviations from normality are treated as threats. This technique presents two main problems [8]: a high false alarm due to the possibility of an attacker simulates legitimate activities and normally, it is used synthetical data due to difficult to find real training datasets. Moreover, some studies have employed machine learning techniques in inappropriate manner [9].

Therefore, in this paper, we analyze the challenges that must be overcome to provide an anomaly-based network intrusion detection mechanism and we present a framework that uses network programmability through SDN (Software Defined Networking) and a machine learning algorithm that works with continuous data stream clustering. In addition, the algorithm provides an outlier detection mechanism inside it. In our experiment, we use a DDoS attack as network intrusion to demonstrate the viability of the proposed framework

The remainder of this paper is organized into six sections. In Section II, we present the related works and highlights our contribution. In Section III, we present the main challenges in the development of an anomaly-based mechanism using machine learning algorithms. In Section IV, we outline a set of requirements to improve such mechanisms. In Section V, we describe our framework to meet the requirements presented in the Section V. In Section VI, we present and discuss some results. Finally, in Section VII, we discuss the future research and conclude the paper.

II. Related Work

Our anomaly-based solution mainly comprises three concepts of machine learning algorithms: clustering, outlier detection and data streaming so that we revised the literature considering those three factors.

In [10], the authors present an unsupervised solution based on a modification of outlier detection mechanism of random forest algorithm. The experiments were performed using KDD'99 dataset. The results were like results of previous approaches.

Devarakonda *et al*. [11] explore the possibility to detect outliers using a multi layered framework. This solution is adequate to high dimensional datasets. The experiments are performed using the KDD'99 dataset.

In [12], the authors present SPOT (Stream Projected Outlier Detector) to find out outliers in Unix system. SPOT can process high-dimensional data streams and detect new attacks. This solution use UNM datasets in the experiments.

Da *et al.* [13] present a method to detect DDoS attacks and SYN flood attacks. The method is an improved mining outlier detection using clustering. The experiments were executed in real time in a local network.

In [14], the authors combine several machine learning techniques to obtain an effective intrusion detection mechanism. The techniques were PCA, K-means and SVM. In the experiment was used KDD´99 dataset.

In [15] and [16], the authors present a self-protection architecture for IoT based on artificial neural network algorithms and fuzzy logic. The main DDoS attacks that occur in IoT environment has been investigated such as, selective forward, blackhole, sinkhole and flooding.

However, those studies that use machine learning algorithms are limited by using of static data usually located in datasets. Those algorithms use small datasets of training data that are available in memory. In some environments, such as IoT applications and TCP/IP traffic that generate high-speed data streams, it is impracticable to store all the data in memory and run multiple passes over the training data. Moreover, the learning model can change over time due to the generation of data by non-stationary distributions, for example the occurrence of a new malicious activity in the computer network. This change has an impact on the algorithm accuracy since the training dataset is soon outdated and there are errors in the estimator forecast. In addition, some systems like SDN networking, the switches do not select their ports and paths through static data, but through data streams. Therefore, the best way to analyze the collected information is not through datasets but through transient data flows. Thus, there is necessity of a new approach that considers transient data flows in the Internet. Our contribution in this paper is to show a framework to fill this gap.

III. Challenges in Anomaly-based Intrusion Detection System

Intrusion Detection System (IDS) is an operational mechanism to detect malicious activity in a computer network. These malicious activities or intrusions can affect the normal operation of system resulting in serious problems of security. As the malicious activities are different of normal behavior of system, anomaly detection techniques can be applied in this case [17]. Some techniques commonly used are machine learning models.

However, it is necessary to consider a set of challenges before to implement an anomaly-based mechanism using machine learning algorithms. We group these challenges in three categories: 1) according to the characteristics of network traffic, 2) according to the problems with anomaly detection approaches and 3) according to the application of machine learning techniques in inappropriate manner.

In the category 1, there are some key challenges in computer network domain including huge volume of data, data streaming fashion and high data variability. Nowadays, the data acquisition is automatic instead of manual. Sensor devices and computers collect, process and send information to other computers continually all the time resulting in huge volume of data. In some networks, such as wireless sensor networks and TCP/IP networks, the best manner to collect and analyze the data is through continuous data stream instead of persistent files. Thus, techniques based on small datasets and in batch are not more viable. Moreover, normal traffic in data stream present high variability, becoming difficult to find stable patterns.

In the category 2, anomaly detection approaches present the following problems: high false alarm rate, training data lack and evasion [8]. The detectors normally generate a high false positive rate, and in network domain that is more problematic due to the huge volume of data. A determined rate is more significant than other domains with a volume of data reduced. In relation a training data, it is not difficult to find labeled data for normal behavior, but it is hard to find labeled data for the intrusions. Finally, it possible for an attacker simulates malicious activity as benign activity so that the system can be evaded.

In the category 3, the use of machine learning techniques in anomaly detection presents the following challenges: outlier detection, high cost of errors, semantic gap and difficulty of evaluation [9]. Although there is a necessity in detecting outliers, machine learning algorithms present higher performance in detecting similarities instead of outliers. That is, that technique is more adequate to detect similarities among the data. In classification problems, the error rate is high when compared with other domains, sometimes damaging all the system. Not always, the detection results are interpreted correctly by the network operator, occurring a semantic gap that need to be resolved. Finally, the difficulty of evaluation is a significant challenge due to the lack of real data to work in this domain.

IV. Proposed Overcomings

A. Overcoming the Category 1 Challenge

For facing this challenge, we use a data stream machine learning algorithm. In machine learning system that apply continuous data stream, it is generated a non-stationary and dynamical environment, which the data are achieved in a continued way and with dynamical unknown. The learning process is continuous and evolves over time. Machine learning algorithms can incorporate new information into the decision model, detect and react to changes, using limited computational resources. Changes that evolve over time include: the user's interest, the type of anomaly, the quality of a product, among others [18].

The properties of a computational model in these learning systems are [18]: 1) incrementality; 2) real-time learning; 3)

to be able to process examples in constant time and limited memory; 4) limited access to processed examples; 5) capacity to detect and to adapt the decision model to concept drift.

The properties from 1) to 4) are adequate to apply in the computer network domain because of the huge volume of data, data streaming fashion and high data variability. Due to the huge volume of data, it is adequate that anomaly-based techniques are efficient computationally to handle that challenge. The characteristic of data streaming demand an incremental and online approach. The property 5) is adequate to face the high data variability due to adaptation of model for new instances.

B. Overcoming the Category 2 Challenge

The machine learning technique is not a panacea that can be employed in all anomaly problem successfully. In this type of problem, it is necessary to have a clear view about the objectives to be reached. The easier manner to get this view is maintain the objectives well limited. That is, try keeping the scope narrow to adapt the detector the specifics of problem and reduce the potential of errors of misclassification [9]. In our problem, for keeping the scope narrow, we treat just one type of attack: the DDoS attack. This attack consists in to try impeding that legitimate users access the services of a system, that is, becoming unavailable the resources or services provide for a computer network. This is done through the exhaustion of system resources such as servers, communication channel, etc. Normally, this attack is executed of a distributed (Distributed Denial-of-Service – DDoS) manner where a botnet is created to generate the attack. Thus, an attacker boosts a DDoS attack through a computer network instead of a simple computer. Thus, we adapt the detector to specifics of this attack and reduce the errors of misclassification.

The data availability is a crucial issue in anomaly detection problem. Working with real data is important because shows that system can used in practice. In our case, we overcome this challenge through using real data of an enterprise computer network.

Evasion occurs when an attacker can simulate malicious traffic as normal traffic so that the system is deluded. For reducing this threat, we consider an SDN environment with a small network that can be used to protect a larger network in an enterprise environment. The site presents low risk for explicit targeting for an attacker and, we consider that there will be a few evasion problems.

C. Overcoming the Category 3 Challenge

Machine learning techniques applied to anomaly detection problems are not well succeeded in relation to other domains. According to Sommer and Paxson [9], the difficulties that appears are due to use of machine learning in an inappropriate manner and point out some recommendations, that we follow, to bypass this problem: understanding the threat model, keeping the scope narrow, reducing the costs and treating evaluation issues.

The high rate of false positive is the principal factor to increase the costs in anomaly detection system. To overcome this problem, we follow three recommendations: 1) reduction of scope of system; 2) dealing with the traffic diversity through the machine learning algorithm over continuous data stream; and 3) rigorous examination of the features of network traffic.

In anomaly detection problem, the semantic gap can appear in two manners: in network operator actions and in the type of networks. Network operators can have difficulties in act in face of determined results. Normally, it is hard to identify what occurring in the network. That is, there is an anomalous traffic or there is a false alarm? We face this challenge through employing a mitigation technique of DDoS attacks, releasing the network operator of this problem. In relation to type of networks, academic networks have security policies different of corporate networks. That is, an anomaly detection system for corporate networks is not adequate for academic networks. Here, we are considering a corporate network.

About evaluation issues, we analyzed manually the false positive rate to check out if there is a case related incorrectly. And, we analyzed the true positive rate and true negative rate to verify if the system learned what it must learn.

In relation to the outlier detection, it is better to use an unsupervised technique. Unsupervised techniques of machine learning are more adequate to true classification problems, therefore with anomaly detection is better used to find out variations of known attacks instead of attacks themselves [17]. Thus, we can train the system with a few known attacks along with normal traffic available, thus overcoming this challenge.

V. FRAMEWORK OVERVIEW

Our framework is constituted of three components: SDN architecture, data stream clustering algorithms and mitigation technique. These components were chosen to implement the recommendations presented in the section IV.

A. SDN Architecture

Nowadays, the most of computer networks that belongs the Internet present an architecture where the control plane and the data plane are highly coupled and embedded in the same network devices. The whole structure is decentralized. This fact was very important in the early days of Internet where the main preoccupation was based on its resilience. That architecture is relatively static and very complex occurring management and innovation problems.

To overcome those problems, two proposals have appeared: to support network management, a small number of vendors offer proprietary specialized hardware solutions, operating systems and control programs; and to overcome the lack of new internal functionality, many specialized components have proliferated in today´s networks, such as firewalls, packet filters, packet inspection machines and so

on. These proposals have increased the complexity of management and innovation of the computer networks and their operations.

Due to the limitations of those proposals, a new approach of network architecture has emerged: the SDN architecture. This architecture is based on four pillars [19]: the control and data plane are decoupled; forwarding decisions are flow-based instead of destination-based; control logic is moved to external entity called SDN controller; and the computer network is programmable through software applications. Fig. 1 shows an SDN architecture with all the main components. The controller is responsible to provide abstractions and essential resources for facilitating the programming of forwarding devices. The forwarding devices execute an elemental operations set to forward network packets to hosts and other network elements.

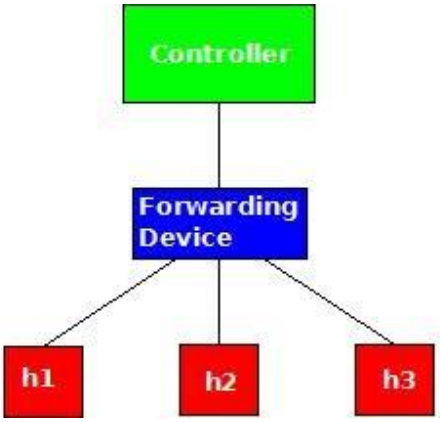

Fig.1 The SDN architecture.

The SDN networks present various advantages in relation to the traditional networks [20]: 1) It is easier to program network applications because now there are several abstractions provided to programming languages available. 2) The integration of different applications becomes more direct, for example it is possible combine a load balancing application with a routing application. 3) the network applications can take actions in any part of computer network. 4) the weak coupling between the control and data plane becomes easier to add new functionality.

We use SDN in our framework because of the easiness of implementing the machine learning algorithm as a network application rather than inserting some hardware device (middlebox) as would be done in traditional networks.

B. *Data Stream Clustering Algorithms*

Researches in data stream clustering algorithms have produced several proposals to provide unsupervised learning. Silva *et al.* [21] describe the main characteristics these algorithms considering some criteria such as window model, outlier detection mechanisms, cluster shape and so on. Considering this study and our problem where an algorithm of machine learning has to present three properties: clustering, outlier detection and data stream. The selected option was the OutlierDenStream algorithm [22].

This algorithm is a modified version of DenStream algorithm [23] and it is adequate to be used in unsupervised environment and in outlier detection. It uses the same concepts than DenStream algorithm: micro-cluster, distances, weight, neighborhood and pruning strategies. These concepts are represented in parameters. Therefore, OutlierDenStream uses a number of parameters (namely, μ, β, λ, and ε). The most of parameters are tuned automatically from OutlierDenStream, except the parameters λ and β that were tuned manually before evaluating the solution. The parameter λ represents the fading factor, that is, it is a weight to eliminate the core cluster before later instances can be added. The parameter β is related to pruning strategies.

The OutlierDenStream run in two phases. In the first phase is necessary to form the clusters and this is done through the DBScan algorithm [24] that builds a buffer dataset. After that, the algorithm maintains the clusters incrementally and when arrives a new sample it labels it as normal or abnormal, that is, the algorithm attempts to cluster normal instances, treating outliers as anomalous traffic.

C. *Mitigation Technique*

For mitigating the DDoS attack, we can employ the characteristics of the flow table of the SDNs. One manner is through the blocking of an entry of the table that supports a link where there is a DDoS attack. It is possible to block that link for a time fixed and after this time the link can be unblocked. Thus, users will not disturb by the activity of security of the computer network.

In order to show how the mitigation technique works, we can consider the scenario shown in Fig. 2.

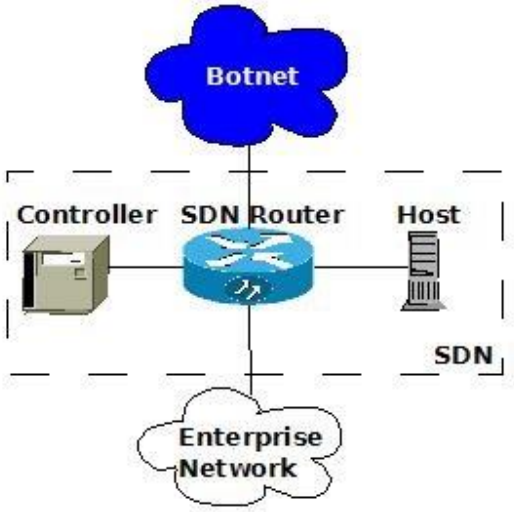

Fig. 2 The SDN Scenario.

The botnet created for an attacker try to attack an enterprise network through an DDoS attack. The enterprise network has access to the Internet through a SDN router that contains a flow table to forward the network packets since authorized by controller. Our data stream clustering algorithm is deployed in the controller because it is responsible to verify all the ingoing traffic. The normal traffic is observed by a certain period of time. Afterwards, the data stream clustering algorithm acts to verify if there is some abnormality in the network. If an DDoS attack is detected, an action is taken in the controller to block the entry in the flow table that supports the suspect traffic. After, a certain time this entry in the flow table is unblocked.

The viability of this technique is possible due to two factors: only one entry in the flow table is blocked and quickly released and also the short duration of DDoS attacks. Thus, it is possible that the detection algorithm can be executed in the controller.

VI. EXPERIMENTAL EVALUATION

In our experiment, we implemented the anomaly-based mechanism using a DDoS attacks as anomalous traffic and we simulated the data streaming through a real dataset of a corporation computer network.

The algorithms were obtained through Python libraries at the 3.5.1 version. The execution of the algorithms to get the best hyperparameters was in a HP desktop, with 4Gb of RAM, Intel i5-3470S (2.9GHz) executing in an Ubuntu 12.04.5 LTS operational system.

A. Data Stream and Processing

The used dataset contains traffic of a corporation network that was attacked for a period of 24 hours [25]. It was selected 76 features as shows the Table I. The normal behaviour contains traffic of five different activities such as checking of e-mail and file transferring among the users of the company. The normal traffic was labelled as “Bening” and the DDoS attack traffic was labelled as “Attack”.

For the experiments, we use four datasets with 1%, 2%, 3% and 4% of DDoS attack traffic. Each dataset was generated of a unique dataset that contains 464976 labelled instances of training and 4% of DDoS attack traffic. Table I shows the description of each dataset.

Table I. Dataset Description

Number of dataset	Number of features	Percentage of DdoS attack traffic	No of labelled instances
1	76	4 %	464976
2	76	3 %	461464
3	76	2 %	456814
4	76	1 %	452163

For evaluating our technique, the following performance metrics were used: Accuracy, Precision, Recall and False Positive Rate. The results of anomaly detection are commonly represented in a confusion matrix composed of TP (True Positives), FN (False Negatives), TN (True Negatives), and FP (False Positives), respectively. The Precision and Recall are defined as: Precision=TP/(TP+FP), Recall=TP/(TP+FN). The Accuracy is defined as: Accuracy = NCD/TI, where NCD is the Number of Correct Detections and TI is the Total of Instances. The False Positive Rate is defined as: FPR = FP/(TN+FP).

The Accuracy metric shows the capacity of success of the technique, the Precision metric measures how well the technique detected abnormal instances, the Recall metric complements the Precision metric for all the instances, and the False Positive Rate (FPR) indicates the percentage of false alarms generated, in this case, normal traffic identified as DDoS attack.

It is common in anomaly detection problem to present the results as ROC (Receiving Operating Characteristics) curve [26]. The ROC curve plots the detection rate, represented by the Recall metric, against the False Positive Rate. Thus, it is possible evaluate the performance of an anomaly detector through relating to two performance metrics.

B. Results and Discussion

After the selection of performance metrics, we use the dataset of an enterprise computer network. The instances are converted in data stream by taking the data input order as the order of streaming. In the data streaming, appears only two types of network traffic: normal traffic and DDoS attack traffic.

For adequate use of the OutlierDenStream algorithm, we must tune, manually, the parameters λ and β using a set of training data. For simulating the data stream, we collect 452163 instances of normal traffic and 1% of DDoS attack traffic. Fig. 3 shows the variation of performance metrics (Recall, Precision, Accuracy and FPR) against the parameter λ. We can observe the best choices are in the interval from 0.015 to value 0.03. The value 0.03 was used in our validation. The parameter β have not presented any variation in the metric performances and was established to zero in all experiments.

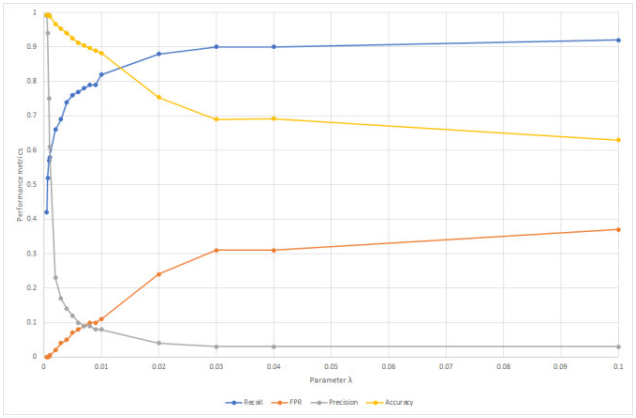

Fig. 3 The performance metrics versus parameter λ. – dataset 4

After 452163 instances of data stream, the OutlierDenStream algorithm achieved an Accuracy of 98.98%, Precision of 60%, Recall of 60% and FPR of 0.48%. It is important to point out the low FPR, because, when a false positive is identified in anomaly problem, a normal traffic can be blocked as a mitigation action. A false positive requires expensive time of a network administrator to examine a problem that did not occur or processing time if the mitigation is made automatically.

Fig. 4 shows the ROC curve for the dataset that contains 1 % of DDoS attack (dataset 4). We can observe that our technique can achieve a high detection rate for a low false positive rate. We can note that the best combination is a 0.9 Recall and a 0.2 FPR.

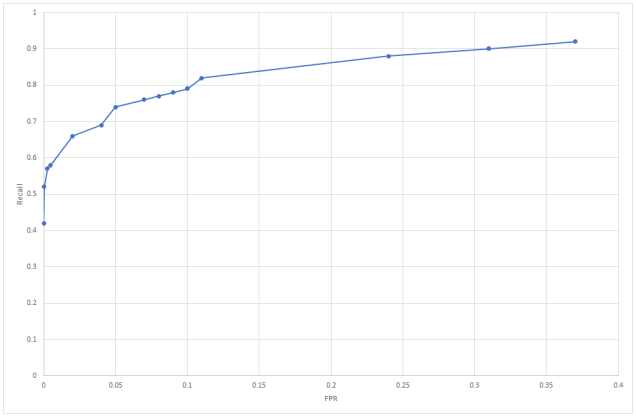

Fig. 4 The ROC Curve – dataset 4

Fig. 5 shows the Precision-Recall plot for the dataset 4. We can observe that the best combination is a 0.6 Precision and a 0.98 Recall. We can see that it is possible to improve the Precision but, in this case, we have a high decrease in Recall.

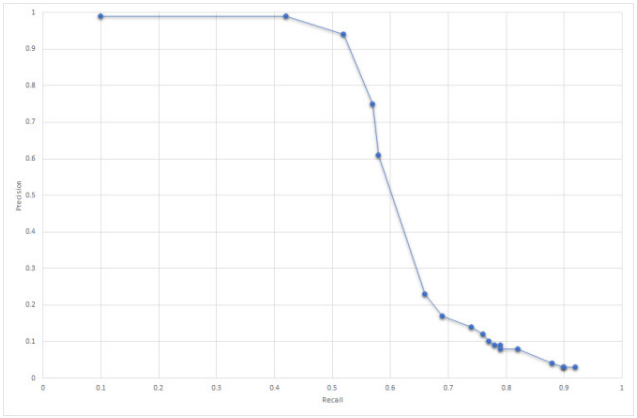

Fig. 5 The Precision-Recall plot – dataset 4

The most of machine learning algorithms present overfitting. Overfitting occurs when there are a lot of errors on instances data that has not been trained, that is, the machine learning algorithms cannot generalize. Basically, there are two manners to prevent overfitting: using the cross-validation technique or split instances in different datasets (training and testing datasets). In our experiments, we use four different datasets and measure the difference in Recall between them. As shown in Fig. 6, the difference between the Recall is less than 2%, that confirm that there is not overfitting on the algorithm used.

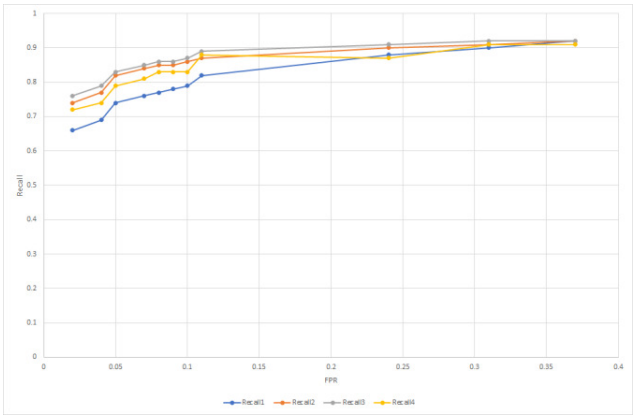

Fig. 6 The recall metric – datasets 1,2,3 and 4

VII. Conclusions and Future Work

In this paper, we have used an SDN architecture, data stream clustering machine learning algorithms and a mitigation technique to design a security framework for detecting and mitigating the DDoS attacks that can occur in the Internet. With this framework will be possible to protect any corporate computer network connected to the Internet.

From experiments, we can observe the effectiveness of our solution through the performance metrics used. On normal traffic, the 1% DDoS attack attains an Accuracy of 98.98%, a Precision of 60%, a Recall of 60% and FPR of 0.48%, while the 4% DDoS attack attains closer values of

the 1% DDoS attack attains an Accuracy of 98.98%, a Precision of 60%, a Recall of 60% and FPR of 0.48%, while the 4% DDoS attack attains closer values of the 1% DDoS attack. We can also note that in our framework it is possible to improve the precision decreasing the recall.

In the future works, we will implement our solution in an online and real environment. Besides, we will implement the mitigation technique of DDoS attacks inside the SDN environment. Also, we will develop new techniques to avoid various attacks that can occur in the Internet using the SDN environment to protect a computer network.

REFERENCES

[1] B. Sun, L. Osborne, Y. Xiao, et al. Intrusion detection techniques in mobile ad hoc and wireless sensor networks. IEEE Wirel Commun 2007; 14: 56–63.

[2] V. Paxson. Bro: a system for detecting network intruders in real-time. Comput Netw 1999; 31(23): 2435–2463.

[3] J.B. Cabrera, L. Lewis, X. Qin et al. Proactive detection of distributed denial of service attacks using MIB traffic variables—a feasibility study. In: Proceedings of the 2001 IEEE/IFIP international symposium on integrated network management, Seattle, WA, 14–18 May 2001, pp.609–622. New York: IEEE.

[4] M. Roesch. Snort: lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on system administration, Seattle, WA, 7–12 November 1999. Berkeley, CA: USENIX Association.

[5] C.M. Cheng, H. Kung and K.S. Tan. Use of spectral analysis in defense against DoS attacks. In: Proceedings of the IEEE global telecommunications conference, 2002 (GLOBECOM' 02), Taipei, Taiwan, 17–21 November 2002, vol. 3, pp.2143–2148. New York: IEEE.

[6] A. Hussain, J. Heidemann and C. Papadopoulos. A framework for classifying denial of service attacks. In: Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications, Karlsruhe, 25–29 August 2003, pp.99–110. New York: ACM.

[7] A. Hussain, J. Heidemann and C. Papadopoulos. Identification of repeated denial of service attacks. In: Proceedings of the 25th IEEE international conference on computer communications (INFOCOM 2006), Barcelona, 23–29 April 2006, pp.1–15. New York: IEEE.

[8] C. Gates and C. Taylor, "Challenging the Anomaly Detection Paradigm: A Provocative Discussion," in Proc: Workshop on New Security Paradigms, 2007.

[9] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection", In Proc. of IEEE Symposium on Security and Privacy, pp. 305-316, 2010.

[10] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *Proc. 2006 IEEE International Conference on Communications (ICC)*, 2006, vol. 5, pp. 2388-2393.

[11] N. Devarakonda, S. Pamidi, V. V. Kumari, and A. Govardhan,"Outliers Detection as Network Intrusion Detection System Using Multi Layered Framework," in *Advances in Computer Science and Information Technology*, Communications in Computer and Information Science Vol. 131, 2011, pp. 101-111.

[12] D. Kershaw, Q. Gao, and H. Wang, "Anomaly-based network intrusion detection using outlier subspace analysis: A case study," in *Advances in Artificial Intelligence*, Lecture Notes in Computer Science Vol. 6657, 2011, pp. 234-239.

[13] W. Da and H. S. Ting, "Distributed intrusion detection based on outlier mining," in *Proc. 2012 International Conference on Communication, Electronics and Automation Engineering (ICCEAE)*, Advances in Intelligent Systems and Computing Vol. 181, 2013, pp. 343-348.

[14] P. Manandhar, and Z. Aung. "Intrusion Detection Based on Outlier Detection Method". Intl' conference on Intelligent Systems, Data Mining and Information Technology (ICIDIT'2014) April 21-22, 2014 Bangkok (Thailand)

[15] R. M. A. C. Mello; A. R. L. Ribeiro; F. M. Almeida, and E. D. Moreno."Mitigating attacks in the Internet of Things with a Self-protecting Architecture". In: AICT 2017 - The 30th Advanced International Conference on elecommunications, 2017, Venice. Proc. of AICT 2017. Paris: IARIA, v.1. p. 1-6, 2017.

[16] F. M. Almeida; A. R. L. Ribeiro; E. D. Moreno; and C. A. E. Montesco. . "Performance Evaluation of an Artificial Neural Network Multilayer Perceptron with Limited Weights for Detecting Denial of Service Attack on Internet of Things". In: In: AICT 2016 - The 12th Advanced International Conference on Telecommunications, Valencia. Proc. of AICT 2016. Paris, France: IARIA XPS Press, v. 1. p. 1-6, 2016.

[17] V. Chandola, A. Banerjee and V. Kumar. "Anomaly Detection: A Survey", ACM Computing Surveys, pp 1-72, 2009.

[18] P. Domingos and G. Hulten. A general method for scaling up machine learning algorithms and its application to clustering. In: Proceedings of the Eighteenth International Conference on Machine Learning, p. 106-113, 2001.

[19] J. A. Wickboldt, W. P. de Jesus, P. H. Isolani, C. B. Both, J. Rochol, and L. Z. Granville, "Software-Defined Networking: Management Requirements and Challenges," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 278–285, Jan 2015.

[20] D. Kreutz, P. E. Veríssimo, S. Azodolmolky, "Software-Defined Networking: A Comprehensive Survey", arXiv preprint arXiv:1406.0440, 2014

[21] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. Carvalho and J. Gama, "Data Stream Clustering: A survey", ACM Computing Surveys, vol 46, Issue 1, October 2013.

[22] "The OutlierDenStream Algorithm", https://github.com/anrputina/OutlierDenStream.

[23] F. Cao, M. Ester, W. Qian and A. Zhou, "Density-based Clustering over an Evolving Data Stream *SIAM Conference Data Mining*, Bethesda, 2006.

[24] M. Ester, H. Kriegel, J. Sander and X. Xu, "A Den-sity-Based Algorithm for Discovering Clusters Spatial Databases with Noise," *International Conference on Knowledge Discovery in Databases and Data Mining* (*KDD*-96), Portland, pp. 226-231, 1996.

[25] A. Ghorbani and A. H. Lashkari, CDMC2018 Dataset: DDoS Attacks Detection for Enterprise Network Security, Canadian Institute for Cybersecurity, University of New Brunswick, http://www.csmining.org/

[26] T. Fawcet, "An introduction to ROC analysis", Pattern Recognition Letters, p. 861-874, 2005.