Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

Secure SDN Traffic based on Machine Learning Classifier

Mahmood K. Mohammed[1], Zaid A. Abod[2], Alharith A. Abdullah[3]
[1,2]Al-Qasim Green University, Hillah-Iraq.
[3]University of Babylon, Hillah-Iraq.
mahmood@uoqasim.edu.iq[1], zaid@uoqasim.edu.iq[2], alharith@itnet.uobabylon.edu.iq[3]

DOI: 10.5281/zenodo.6786157

ABSTRACT

Nowadays, the majority of human activities are carried out utilizing a variety of services or applications that rely on the local and Internet connectivity services provided by private or public networks. With the developments in Machine Learning and Software Defined Networking, traffic classification has become an essential study subject. As a consequence of the segregation of control and data planes, Software Defined Networks have some security flaws. To cope with malicious code in SDN, certain operational security techniques have been devised. In this paper, a machine learning model, supervised, was utilized to identify normal and malicious traffic flows. While, normal traffic were generated using Internet traffic generator, malicious traffic were accomplish by Scapy and Python. The main network features of the OpenFlow flow table such as Packets count, bytes counts, packet rates, byte rate for forward and revers flows, were extracted. The combination of good ML classifier and dataset produced the greatest accuracy rate over 99% in DDoS attack detection, according to the results. Further to the main aim, the presented approach could be utilized to classify different traffic flows with the purpose of balance and priorities the important traffic.

Keywords: SDN, Machine Learning, Networking, Network Security.

Cite as: Mahmood K. Mohammed, Zaid A. Abod, Alharith A. Abdullah. (2022). Secure SDN Traffic based on Machine Learning Classifier. LC International Journal of STEM (ISSN: 2708-7123), 3(1), 118–128. https://doi.org/10.5281/zenodo.6786157

INTRODUCTION

In general, SDN allows you to virtualize networks, make it simpler, and configure and administer it from a single location. It distinguishes the control plane, which determines where packets are routed in routers and switches, from the data plane, that forwards traffic to its destination (Ropke & Holz, 2015; Scott-Hayward et al., 2016). Controlling network flows with SDN is possible thanks to a centralized control program running on a server or VM. This controller establishes set of rules to manage and handle network traffic (Polat et al., 2020). After that, network forwarding devices are programmed with the rules. The router and switches, in a sense, become "slaves" to this application-driven controller.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

SDN is one of the virtualization's applications. In recent years, it has become a widely utilized and well-known network architecture. SDN is used to separate the data plane from the control network. Control, data, and application planes are all part of the SDN architecture (Thakare & Pund, 2021). The data plane is made up of equipment like routers and switches. The data plane was both controlled and programmed by the control plane. The control plane is in charge of the devices that are employed in the data plane for transmission. The controller, the network's brain, is likewise located in this plane. The controller generates a set of rules that are used by network equipment for packet transmission. Through a controller, the Application layer communicates with various devices on the network's infrastructure.

Manageability, scale, and enhanced performance are just a few of the benefits of Software Defined Networking (Polat et al., 2020). SDN, on the other hand, has its own set of security issues, particularly if the controller is vulnerable to DDoS attacks. Although Software Defined Networking provides a simple and straightforward method for network management, it also introduces a new security concern. Denial-of-Service assaults, Man-in-the-Middle attacks, and other types of attacks are among the risks. The most widespread and well-known attack is DDoS. The entire network fails as a result of this effect. Therefore, detecting this attack is critical for a network's security.

Despite the buzz about SDN, security concerns have just lately been addressed. There is a vast range of viewpoints on this topic. Some feel that the security issues raised by SDN are solvable, and that SDN can even provide security advantages; others say that Pandora's box has been opened, and that SDN-enabled networks will become extremely hard, if not impossible, to effectively protect (Ropke & Holz, 2015; Scott-Hayward et al., 2016).

Through the communication route between the controller and the data plane, the controller is vulnerable to DDoS attacks. If the flow input in the flow table does not match the packets arriving at the OpenFlow switch, the packets are placed in the flow buffer. In this case, the controller's sources are rendered inoperable, and the network is rendered useless. The bandwidth of the communication line between the controller and the OpenFlow switch that is exposed to attack traffic is reduced. The data plane is vulnerable to DDoS attacks because of the flow table in network devices. Packets from unknown sources are sent to the switch in DDoS assaults. For these arriving packets, the controller creates a rule and directs them to the switch's flow table. The switch flow table's capacity is reached. The flow table cannot be updated with new rules, hence packets cannot be forwarded. Only the flow entry from the controller is used by OpenFlow switches to manage packets.

Network security and cyberattacks have improved significantly during the last decade, yet the growth of network attacks has far surpassed the protection mechanisms. In this period of rapidly evolving network threats, wiser and much more efficient approaches to maintaining networks and data secure must be created (Sultana et al., 2021; Zaman et al., 2020). Machine learning has the capacity to classify SDN traffic flows effectively and efficiently. ML's advantages include its ability to handle high-dimensional data and map classes with extremely complicated properties. Machine learning-based traffic classification is a technique that is growing quickly. The process of recognizing and connecting packet flows to traffic classes is known as traffic classification, and it is based on information collected from the traffic as input. The purpose of traffic classification is to improve network resource management, network security, and service quality (Crotti et al., 2006; Ng et al., 2015). SDN might cause security issues if the controller is vulnerable to DDoS attacks (Liyanage et al., 2017). Machine learning-based models were utilized to identify DDoS assaults in SDN.

Presently, many researchers used the ML to classify SDN traffic depending on publicly available dataset or live SDN traffic flow. In (Bakker et al., 2019) authors reported experience in set up network traffic

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

classifiers in a real SDN Network. Using publicly accessible datasets, they construct a standard reference for each classifier's performance in relations of detection, accuracy and precision percentage. An analogous testing conducted on a real Software Defined Network reveals that the classifiers accomplish much worse than the reference standard, 11 percent worse accuracy, precision is lower by up to 30 percent, and a detection percentage of less than 15%. They contend that communications between the switch and the controller have a major impact on the conduct of machine learning algorithms in a live network that should be taken into consideration in a real-world utilization. They claimed that the performance of their classifiers is less efficiently comparing to the one applied on offline datasets. It is crucial to highlight that the poor traffic categorization is due to the classifiers' inability to detect fraudulent flows in the dataset. Each classifier correctly identified normal traffic. Observing away from the accuracy of the classifiers, research indicates that greater consideration should be devoted to the networking settings in which machine learning is employed, because not sufficient attention is given to the influence of machine learning algorithms by the network environment they operate in.

The researchers in (Malik et al., 2020) have implemented a new deep learning model, called Deep-SDN. Their suggested model can identify network traffic application kinds with high accuracy and speed, making it suitable for identifying online traffic. The experimental results demonstrated Deep-efficacy SDN's in recognizing traffic application kinds. The proposed study in (Abubakar & Pranggono, 2017) describes intrusion detection for SDN based on machine learning. The flow-based IDS model is constructed on a signature-based IDS architecture to identify anomaly-based attacks in the SDN environment. Pattern Recognition is employed in this article since it outperforms other types of neural network models in terms of accuracy. Researchers in (Le & Tran, 2020) suggested three Deep Learning models and compared them to typical machine learning techniques. In comparison to established methodologies, the experimental results demonstrate that the proposed methodology with high accuracy has good potential for further improvement.

SOFTWARE-DEFINED NETWORKING

The goal of software-defined networking is to separate the control and data layers (see figure 1). The main benefit of this separation is that it makes network management and control easier, as well as allowing both layers to evolve independently (Farhady et al., 2015; Kareem & Jasim, 2022b). The most essential aspect of SDN is the control plane's centralization, which makes the network more adaptive, flexible and programmed. SDN is a networking approach that lets open protocols to operate network switches and routers using software controls and abstract infrastructure according to application and network service necessities (Xie et al., 2019). This allows network administration and control to be implemented using software. The term software denotes that network devices may be programmed, but it does not imply that software is in charge of everything. This method aids network and infrastructure management, control plane modularity, cost-effectiveness, generic data planes, and flexible networks, making virtualization simple to build as needed (Yurekten & Demirci, 2021).

The software defined networking results in a customizable network infrastructure. This automatically qualifies for software control and network flow, with network devices controlled by software applications. Instead of manually rebuilding the network infrastructure, a network engineer now is empowered and should be capable to reprogram it (von Rechenberg et al., 2021). This led to performance enhancement duo to the separation of the two layers.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

MACHINE LEARNING (ML)

Machine learning, in combination with big data technology and high-performance computers, has opened up new avenues for unraveling, quantifying, and understanding data-intensive processes in SDN traffic flows. Machine Learning is defined as a scientific field that, along with other things, allows machines to learn without the necessity to be exactingly programmed (Gupta & Grover, 2021; Kareem & Jasim, 2022c; Rahul et al., 2017).

ML techniques often incorporate a learning process with the aim of learning to achieve a function from experience depending on training data. Data in machine learning is composed of cases. An individual example is generally described by a collection of characteristics, sometimes known as features or variables (Aslam et al., 2022). The type of features could be one of the following binary, nominal, ordinal or numeric. The performance of the ML model in a given work is measured using a performance metric that increases with experience. To calculate the success of ML models and algorithms, many statistical and mathematical models are utilized. When the learning process is complete, the trained model may be used to perform classification, prediction, or clustering on new samples (testing data) using the acquired knowledge during the training phase.

Machine learning tasks are allocated into two collections based on the learning data of the learning system: supervised and unsupervised learning (Le & Tran, 2020). Data is delivered with sample inputs and outputs, with the goal of evolving a universal principle that links inputs (raw data) to outputs (results), in supervised learning. On the other hand, unsupervised learning, does not differentiate across training and test sets since the data is unlabeled. The learner analyzes input data in order to expose interesting patterns (Le & Tran, 2020).

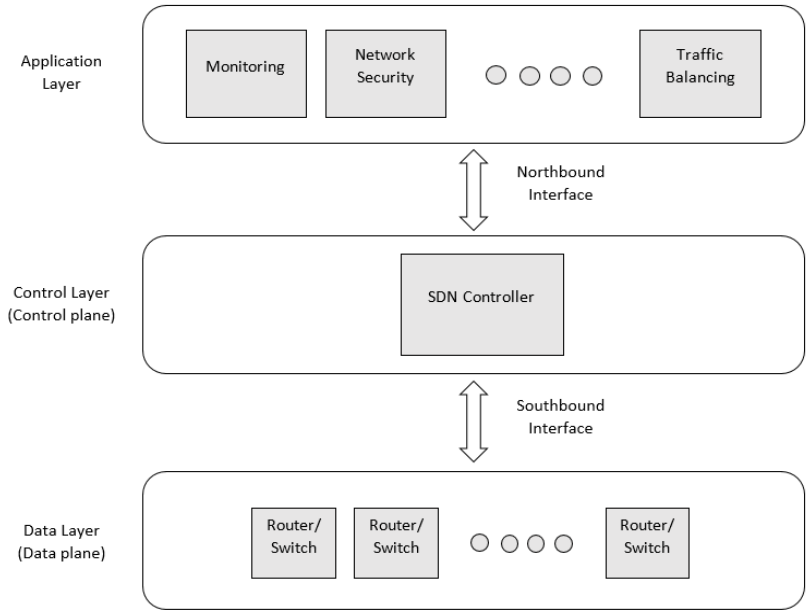

Figure 1: SDN architecture.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

ATTACKS ON SDN

SDN architecture's centralized nature presents additional vulnerabilities, making the network exposed to a variety of security attacks. All three layers of SDN, actually, are inherently vulnerable to many forms of attacks (Gupta & Grover, 2021). Some of these threats are particular to SDNs, resulting from the separation of control and data plane functions. These attacks might take place either in the SDN controller or on communication routes connecting control and data plane devices. Furthermore, there are certain threats that are similar to SDN standards and traditional networks, such as cyberattacks on application layer or data plane layers (Nisar et al., 2020). Although some threats are common and have a slight to moderate impact on traditional networks, the effect of these threats is magnified in SDN. In this part, we discuss brief attacks clustering that have a significant influence on several aspects of SDN. We divide the primary SDN network attacks into four categories, which are as follows (Elsayed et al., 2020):

Data Plane Layer Attacks

The intruder may attack network components directly. To launch various attacks, the attacker could get unauthorized entry to unprotected hosts in the SDN network (Kareem & Jasim, 2022a). Furthermore, the attacker might overwhelm the nodes by generating malicious traffic on a hosting computer or associated switch. The primary purpose of these attacks is to drain the controller assets or the flow table-space of OF-Switch. Furthermore, the attacker can inflict network resource harm by establishing a bogus switch in the SDN network to divert network traffic or steal data (Wang & Li, 2021). Additionally, the attacker can divert network traffic for stealing purposes, modify the OpenFlow switch's flow entry rules to redirect genuine traffic, overwhelm the controller, or slow down network traffic. On the other hand, physically compromising the hardware switches of a traditional network and changing its forwarding tables is far more difficult.

Control Plane Communication Attacks

The controller in an SDN network may manage data plane devices via communication links. Theoretically every device has its individual channel with the controller, however practically, all of these channels are connected by the same physical connection. Using spoofed sources to launch a flooding attack might cause channel connections to become congested (Behal & Kumar, 2017). As a result, disrupting communication between controller and data plane components could disconnect the controller from the rest of the network. Additionally, the attacker could conduct a man-in-the-middle attack, sniff vital data, or obtain complete control of the controller plane by attacking the connection between the controller and the OpenFlow switches (Khairi et al., 2021).

Control plane layer attacks

The controller is the central brain of the entire network in the SDN architecture. Getting access to or shutting down the controller might cause the entire system to go down (Ubaid et al., 2017). Furthermore, the controller is subject to the same security flaws as the operating system it runs (Xie et al., 2019). In rare circumstances, the attacker may utilize a fake controller and route network traffic according to malicious configuration. Furthermore, if the attacker successfully uses the weak Northbound API, he may take control of the entire network and set his own policies (Sherwood et al., n.d.).

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

Application layer attacks

The difficulty of getting information about the SDN network is depending on the belongingness of the SDN applications (either from the same provider as the controller or from a third-party). The attacker could easily lunch varieties of attacks on the second kind of the SDN applications due to the open environment (Behal & Kumar, 2017). The attacker could even easier target the controller and the application in case he is the third-party application.

PROPOSED FRAMEWORK

As stated, we proposed a traffic classification based on machine learning models for the SDN traffic flows. There are five steps in the suggested framework as illustrated in figure 2. Firstly building the SDN network topology (more in the next section). Then normal and attack traffic flows were simulated. The attack was carried out on the data plane layer (the first category of the primary attacks against the SDN networks as stated above). After that, the training data were collected depending on the output data from Ryu controller. Fourthly, supervised machine learning models were trained and tested. Finally, utilizing the ML model from the previous step to classify SDN traffic flows in real time.

After successfully configuring and deploying the SDN topology (first step), specific tools were utilized to simulate network traffics. AS known, in machine learning, data is the most important factor to produce efficient ML model. Set of data were collected and used to extract the attributes as illustrated in table 1. Using five virtual machines (Virtual Box) to create simple network topology as illustrated in figure 3. The network is involves one SDN controller, one OpenFlow switch and three Linux host one of them represent the attacker from inside the network. On the Controller Virtual machine, the Ryu controller

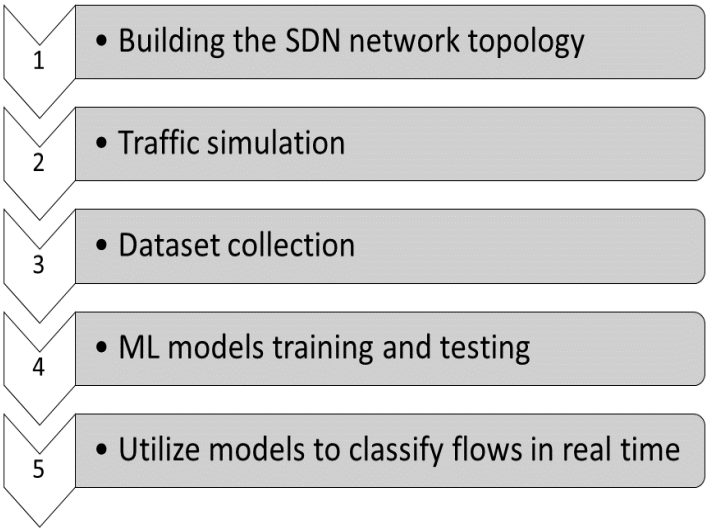

Figure 2: Steps of accomplish the suggested framework

was installed. With the purpose of route hosts' traffics through the OpenFlow Switch virtual machine rather than through the native switching mechanism Virtual Box, an overlay network was set up. The network's hosts utilized Open vSwitch with two interfaces. While, one interface was internal, the second

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

used VXLAN tunneling to connect to the OpenFlow Switch virtual machine. The OVS Switch was directly linked to the Controller VM using underlay IP. Once the network is configured and correctly established the controller should be alert of packets passing via the OVS switch among network elements. The simple switch monitor script of the Ryu controller was edited to show some data that was utilized as input for the classifier script. This data include time, datapath, in-port, eth_src, eth_dst, out-port, total_packets and total_bytes. Then a flow object was formed with a set of attributes see table 1.

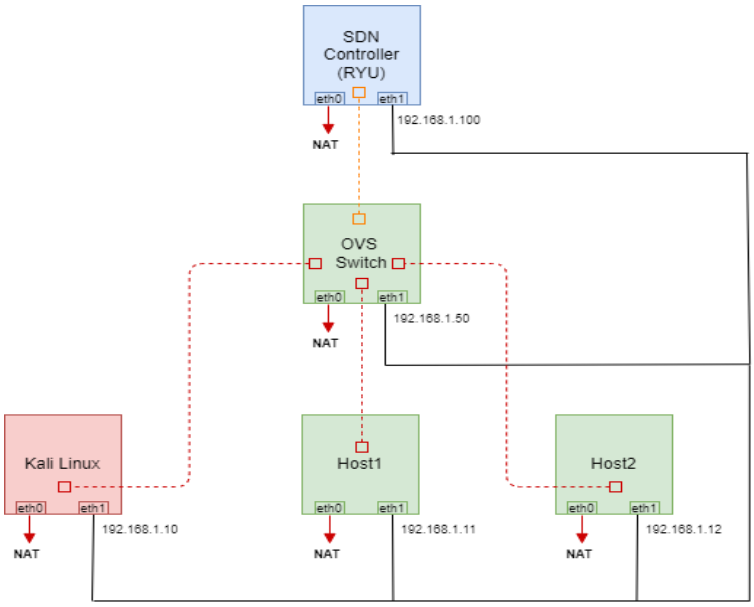

Figure 3: SDN Network simulation environment

EXPERIMENTAL RESULTS AND EVALUATION

A supervised Logistic Regression model was employed as the Machine Learning method. It is employed in the prediction of categorical target variables. The outcome is usually a binary value, but when there are many objectives, it chooses the one with the highest likelihood of occurring. Logistic Regression behaved incredibly well in our program, with an accuracy of above 99 percent (figures 4 and 5 illustrates the decision boundaries and Confusion Matrix for the model respectively).

From the decision boundaries, we clearly notice that the classifier is performing extremely excellent. It does two functions. In the first one, it classify the normal traffic from the malicious one. The second function (or use case) is to classify (identify) the type of normal traffic flows (DNS, PING or Voice). This is beneficial in balancing and priorities the SDN traffic flow.

Figure 5 illustrates A Confusion Matrix, which shows there is essentially no failure in Logistic Regression. It could assist identify where the proposed model is cease to work properly to determine the aim properly. On the y-axis are expected labels, while on the x-axis are genuine labels. The main purpose of the matrix is to display if a model has a tendency to forecast one traffic type with others traffic classes. In our case, the model failed 5 times to identify DDOS traffic from more than 600 instances. This rate is insignificant and could be neglected.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

Table 1: The extracted attributes from the generated traffics

Feature	Description
time_start	The UTC clock value at the time the flow is first detected
datapath	The switch ID to identify the switch in Ryu
in-port	The port receiving the incoming traffic
eth_src	The source MAC address of the flow
eth_dst	The destination MAC address of the flow
out-port	The port sending the outgoing traffic
forward_packets	The total number of packets seen in the forward direction
forward_bytes	The total number of Bytes seen in the forward direction
forward_delta_packets	The number of packets seen since the last forward flow detection
forward_delta_bytes	The number of Bytes seen since the last forward flow detection
forward_inst_pps	The instantaneous packets per second in the forward direction
forward_avg_pps	The average packets per second in the forward direction
forward_inst_bps	The instantaneous Bytes per second in the forward direction
forward_avg_bps	The average Bytes per second in the forward direction
forward_status	The status (active/inactive) of the forward flow
forward_last_time	The UTC clock value of the last time forward flow was detected
reverse_packets	The total number of packets seen in the reverse direction
reverse _bytes	The total number of Bytes seen in the reverse direction
reverse _delta_packets	The number of packets seen since the last reverse flow detection
reverse _delta_bytes	The number of Bytes seen since the last reverse flow detection
reverse _inst_pps	The instantaneous packets per second in the reverse direction
reverse _avg_pps	The average packets per second in the reverse direction
reverse _inst_bps	The instantaneous Bytes per second in the reverse direction
reverse _avg_bps	The average Bytes per second in the reverse direction
reverse _status	The status (active/inactive) of the reverse flow
reverse _last_time	The UTC clock value of the last time reverse flow was detected

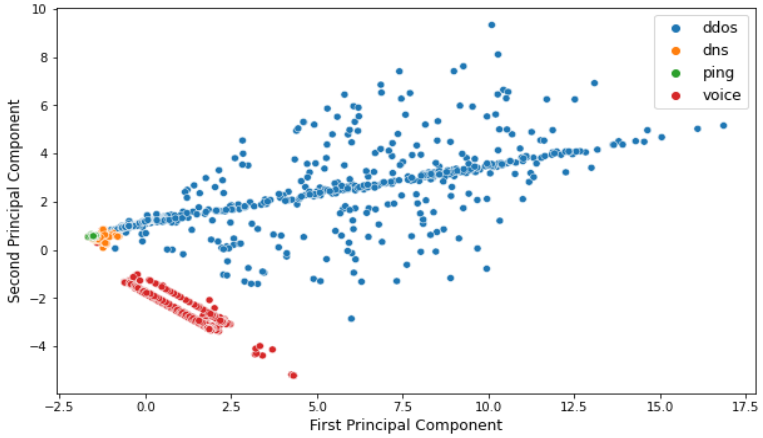

Figure 3: Decision Boundaries

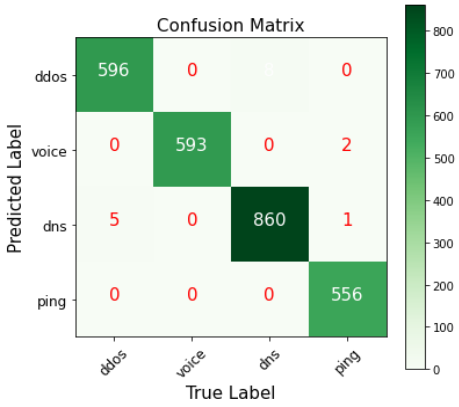

Figure 3: Confusion Matrix

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

CONCLUSION

In this paper, we presented SDN framework that utilized ML classifier to detect DDoS attacks and could be easily leverage to protect the network elements against variety of security threats. With the purpose of provide a proper security for the SDN network elements, we simulate normal and malicious network traffic. While, an entire SDN architecture was build, the DDoS attack simulation took place in one SDN layer (data plane layer). After that a specific purpose dataset was collected. Then, a machine learning model was build, trained and tested using the collected dataset. This approach could be easily applied to protect other SDN layers not only against the DDoS attack but also to detect other types of cybersecurity threats.

REFERENCES

REFERENCES

Abubakar, A., & Pranggono, B. (2017). Machine learning based intrusion detection system for software defined networks. *Proceedings - 2017 7th International Conference on Emerging Security Technologies, EST 2017*. https://doi.org/10.1109/EST.2017.8090413

Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S. A., Elaziz, M. A., Al-Qaness, M. A. A., & Jilani, S. F. (2022). Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT†. *Sensors*, *22*(7). https://doi.org/10.3390/s22072697

Bakker, J., Ng, B., Seah, W. K. G., & Pekar, A. (2019). Traffic classification with machine learning in a live network. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management, IM 2019*.

Behal, S., & Kumar, K. (2017). Characterization and comparison of DDoS attack tools and traffic generators - a review. *International Journal of Network Security*, *19*(3), 383–393. https://doi.org/10.6633/IJNS.201703.19(3).07

Crotti, M., Gringoli, F., Pelosato, P., & Salgarelli, L. (2006). A statistical approach to IP-level classification of network traffic. *IEEE International Conference on Communications*, *1*. https://doi.org/10.1109/ICC.2006.254723

Elsayed, M. S., Le-Khac, N. A., & Jurcut, A. D. (2020). InSDN: A novel SDN intrusion dataset. *IEEE Access*, *8*, 165263–165284. https://doi.org/10.1109/ACCESS.2020.3022633

Farhady, H., Lee, H., & Nakao, A. (2015). Software-Defined Networking: A survey. In *Computer Networks* (Vol. 81, pp. 79–95). Elsevier B.V. https://doi.org/10.1016/j.comnet.2015.02.014

Gupta, S., & Grover, D. (2021). A Comprehensive Review on Detection of DDoS Attacks using ML in SDN Environment. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 1158–1163. https://doi.org/10.1109/ICAIS50930.2021.9395987

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | **Email:** editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

Kareem, M. I., & Jasim, M. N. (2022a). DDOS Attack Detection Using Lightweight Partial Decision Tree algorithm. *2022 International Conference on Computer Science and Software Engineering (CSASE)*, 362–367. https://doi.org/10.1109/CSASE51777.2022.9759824

Kareem, M. I., & Jasim, M. N. (2022b). *The Current Trends of DDoS Detection in SDN Environment*. 29–34. https://doi.org/10.1109/it-ela52201.2021.9773744

Kareem, M. I., & Jasim, M. N. (2022c). Fast and accurate classifying model for denial-of-service attacks by using machine learning. *Bulletin of Electrical Engineering and Informatics*, *11*(3), 1742–1751. https://doi.org/10.11591/eei.v11i3.3688

Khairi, M. H. H., Ariffin, S. H. S., Latiff, N. M. A. A., Yusof, K. M., Hassan, M. K., Al-Dhief, F. T., Hamdan, M., Khan, S., & Hamzah, M. (2021). Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms. *IEEE Access*, *9*, 76024–76037. https://doi.org/10.1109/ACCESS.2021.3081629

Le, D. H., & Tran, H. A. (2020). A novel Machine Learning-based Network Intrusion Detection System for Software-Defined Network. *Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020*, 25–30. https://doi.org/10.1109/NICS51282.2020.9335863

Liyanage, M., Braeken, A., Jurcut, A. D., Ylianttila, M., & Gurtov, A. (2017). Secure communication channel architecture for Software Defined Mobile Networks. *Computer Networks*, *114*, 32–50. https://doi.org/10.1016/j.comnet.2017.01.007

Malik, A., de Frein, R., Al-Zeyadi, M., & Andreu-Perez, J. (2020). Intelligent SDN Traffic Classification Using Deep Learning: Deep-SDN. *2020 2nd International Conference on Computer Communication and the Internet, ICCCI 2020*. https://doi.org/10.1109/ICCCI49374.2020.9145971

Ng, B., Hayes, M., & Seah, W. K. G. (2015). Developing a traffic classification platform for enterprise networks with SDN: Experiences & lessons learned. *Proceedings of 2015 14th IFIP Networking Conference, IFIP Networking 2015*. https://doi.org/10.1109/IFIPNetworking.2015.7145322

Nisar, K., Jimson, E. R., Hijazi, M. H. A., Welch, I., Hassan, R., Aman, A. H. M., Sodhro, A. H., Pirbhulal, S., & Khan, S. (2020). A survey on the architecture, application, and security of software defined networking: Challenges and open issues. In *Internet of Things (Netherlands)* (Vol. 12). Elsevier B.V. https://doi.org/10.1016/j.iot.2020.100289

Polat, H., Polat, O., & Cetin, A. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability (Switzerland)*, *12*(3). https://doi.org/10.3390/su12031035

Rahul, R. K., Anjali, T., Menon, V. K., & Soman, K. P. (2017). Deep Learning for Network Flow Analysis and Malware Classification. *Communications in Computer and Information Science*, *746*, 226–235. https://doi.org/10.1007/978-981-10-6898-0_19

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)

Logical Creations Education Research Institute
LC INTERNATIONAL JOURNAL OF STEM
E-ISSN: 2708-7123
Web: www.lcjstem.com | Email: editor@lcjstem.com
Volume-03 | Issue-01 | March-2022

Ropke, C., & Holz, T. (2015). Retaining control over SDN network services. *Proceedings - International Conference on Networked Systems, NetSys 2015*. https://doi.org/10.1109/NetSys.2015.7089082

Scott-Hayward, S., Natarajan, S., & Sezer, S. (2016). A survey of security in software defined networks. In *IEEE Communications Surveys and Tutorials* (Vol. 18, Issue 1). https://doi.org/10.1109/COMST.2015.2453114

Sherwood, R., Foster, N., Association for Computing Machinery. Special Interest Group on Data Communications, Association for Computing Machinery, ACM Digital Library., & ACM SIGCOMM Conference (2013 : Hong Kong, C. (n.d.). *HotSDN'13 : proceedings of the 2013 ACM SIGCOMM Workshop on Hot topics in Software Defined Networking : August 16, 2013, Hong Kong, China*.

Sultana, R., Grover, J., & Tripathi, M. (2021). Security of SDN-based vehicular ad hoc networks: State-of-the-art and challenges. In *Vehicular Communications* (Vol. 27). Elsevier Inc. https://doi.org/10.1016/j.vehcom.2020.100284

Thakare, S., & Pund, M. A. (2021). *Software Defined Network: Comprehensive Study* (pp. 603–611). https://doi.org/10.1007/978-981-33-6307-6_61

Ubaid, F., Amin, R., Ubaid, F. bin, & Iqbal, M. M. (2017). Mitigating Address Spoofing Attacks in Hybrid SDN. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 8, Issue 4). www.ijacsa.thesai.org

von Rechenberg, M., Rettore, P. H. L., Lopes, R. R. F., & Sevenich, P. (2021, May 4). Software-Defined Networking Applied in Tactical Networks: Problems, Solutions and Open Issues. *2021 International Conference on Military Communication and Information Systems, ICMCIS 2021*. https://doi.org/10.1109/ICMCIS52405.2021.9486399

Wang, H., & Li, W. (2021). DDosTC: A transformer-based network attack detection hybrid mechanism in SDN. *Sensors*, *21*(15). https://doi.org/10.3390/s21155047

Xie, J., Richard Yu, F., Huang, T., Xie, R., Liu, J., Wang, C., & Liu, Y. (2019). A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. In *IEEE Communications Surveys and Tutorials* (Vol. 21, Issue 1, pp. 393–430). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2018.2866942

Yurekten, O., & Demirci, M. (2021). SDN-based cyber defense: A survey. *Future Generation Computer Systems*, *115*, 126–149. https://doi.org/10.1016/j.future.2020.09.006

Zaman, S., Shamim Kaiser, M., Khan, R. T., & Mahmud, M. (2020). Towards SDN and Blockchain based IoT Countermeasures: A Survey; Towards SDN and Blockchain based IoT Countermeasures: A Survey. *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*. https://doi.org/10.1109/STI50764.2020.9350392/20/$31.00©20XX

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0)