iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** III **Month of publication:** March 2023

DOI: https://doi.org/10.22214/ijraset.2023.49565

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue III Mar 2023- Available at www.ijraset.com

Using Machine Learning in Software Defined Networks to Recognize and Avoid DDOS Attacks

Naman Sonthalia[1], E.V. Avinash Reddy[2], Harsh Pagaria[3], G. Vani Jayasri[4]
[1, 2, 3, 4] *Department of Computer Science and Engineering, GITAM*

Abstract: ***In recent years, Distributed Denial of Service (DDoS) attacks have become a significant concern for network administrators and internet service providers. DDoS attacks are designed to overwhelm a target network with a flood of traffic, causing it to crash or become unavailable. These attacks can have severe consequences, including financial losses, reputational damage, and disruption of critical services.***
In response to this growing threat, researchers have been exploring various techniques to detect and mitigate DDoS attacks. One promising approach is to use machine learning (ML) algorithms in software-defined networks (SDN). SDN provides a centralized control plane that enables fine-grained control over network traffic, making it an ideal platform for deploying ML-based DDoS detection and mitigation techniques. This paper discusses the use of ML in SDN to recognize and avoid DDoS attacks, providing an overview of the key concepts, techniques, and challenges involved in this emerging field. Six ML algorithms were assessed, viz. Logistic Regression (LR), Naive Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The inspection showed that KNN, DT and RF is the best appearing ML algorithm. Results showed that our model is able to recognize attacks precisely and rapidly.
Keywords: Software Defined Networks, Attack Recognition, Attack Avoidance, Machine Learning, Feature Extender, Generate Traffic, Launch Attack, Prevent Attack.

I. INTRODUCTION

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices. From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination. It's critical to act soon after discovering a DDoS attack since it provides you the chance to avert significant disruption. The server can start to crash on waiting too long, and a full recovery might take hours. The hardest part about avoiding a DDoS attack is that often it's virtually impossible to do so without impacting legitimate traffic. This is because attackers go to great lengths to masquerade fake traffic as real. To prevent this, we have devised a machine learning-based model that can automatically recognize and avoid assaults in Software Defined Networks.

Software Defined Networking (SDN) is a way to manage computer networks. It's like having a remote control for your network that allows you to program how data flows through the network. This makes it easier to manage the network and helps it run more efficiently. Instead of needing to make changes to individual devices, an SDN controller can make changes to the network as a whole. It's kind of like having a traffic controller for the internet! SDN can also help keep the network secure by controlling access to different parts of the network. In short, SDN makes networks easier to manage, more efficient, and more secure.

Detecting a DDoS attack in an SDN can be challenging, but there are several methods that can be used to identify such an attack. One common method is to monitor the network traffic for any unusual spikes or patterns that indicate a sudden increase in traffic. SDN controllers can be configured to collect and analyze real-time data on network traffic and flow statistics. If a large number of packets are being sent from the same source or if the traffic patterns are irregular, this could indicate a DDoS attack. Another method is to monitor the network performance metrics, such as CPU usage, memory usage, and bandwidth utilization. A sudden increase in these metrics could indicate that the network is being overwhelmed by the attack traffic. Finally, analyzing security logs can help detect any unusual or suspicious activity that could be indicative of a DDoS attack. By employing these methods, network administrators can detect a DDoS attack in its early stages and take appropriate measures to mitigate its impact on the network.

Preventing DDoS attacks in an SDN involves a combination of proactive measures and reactive responses. One of the most effective ways to prevent a DDoS attack is to deploy a robust network security solution, such as firewalls and intrusion detection systems. These solutions can detect and block malicious traffic before it reaches the network. Additionally, network administrators should

implement network segmentation to isolate critical assets from the rest of the network. This can help prevent attackers from accessing sensitive data even if they manage to breach the perimeter. Furthermore, network administrators can leverage traffic engineering techniques to distribute traffic across different paths, ensuring that no single path is overwhelmed by the attack traffic. Finally, network administrators should have a well-defined incident response plan that outlines the steps to be taken in the event of a DDoS attack. By following these steps, network administrators can significantly reduce the risk of a DDoS attack and minimize its impact if it does occur.

The objective of our paper is to introduce a new SDN model that utilizes Machine Learning (ML) to detect and mitigate attacks in SDN networks. The proposed model works by collecting traffic flow entries from switches at regular intervals, extracting the basic flow features and extending them. The extended features include the average flow packet size, the number of flows to the same host as the current flow in the last 5 seconds, and the number of flows to the same host and port as the current flow in the last 5 seconds. Using these extended features, along with the native counters such as packet count and byte count, a detection module employs a binary classification ML algorithm to classify each flow as normal or anomaly. When an attack is detected, the source of the attack is immediately blocked. To determine the best classification ML algorithm to be used in the detection module, we evaluated six algorithms, including Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN), Decision Tree (DT), Naïve Bayes (NB), and Random Forest (RF).

II. RELATED WORK

1) This paper proposed a model which is able to detect and mitigate DDoS attacks automatically in SDN networks using ML. All of the switches' traffic flow entries are periodically collected by the model, which then extracts the native flow features and expands them by incorporating new features. A detection module uses five features to categorize each flow as normal or anomalous. When an attack is discovered, its source is prevented. Six ML algorithms were assessed with regard to the classification ML method utilized in the detection module, including LR, NB, KNN, SVM, DT, and RF. The outcomes of the experiment demonstrated that RF is the best classifier for the generated network. Without losing typical performance, the model was able to swiftly and effectively identify and block attacks.

2) This work proposes a method of DDoS attack detection based on deep belief network feature extraction and LSTM model. This technique uses deep learning to extract IP packet attributes, builds an LSTM traffic prediction model, and then recognizes DDoS attacks using the built-in LSTM model. Technology for detecting DDoS attacks is appropriate for this system. The model can effectively forecast the pattern of typical network traffic, spot irregularities brought on by DDoS attacks, and be used to develop more DDoS attack detection techniques in the future.

3) In this work, the authors proposed a model which analyses the correlation information of flows in data centers. It offers a reliable method of detecting DDoS attacks based on CKNN (K-nearest neighbors traffic categorization with correlation analysis).

4) This paper proposes a new model to detect DDoS attacks in SDN based on SVM. Firstly, a trained Support Vector Machine (SVM) approach is used by the model to extract numerous important features from the packet-in messages and measure the distribution of each feature using entropy. Studies reveal that this technique is highly effective at both real-time DDoS mitigation and security event detection.

5) This paper is about the exploring the attacks on the SDN network Topology using Poisoning. It also talks about the mitigation methods against Network Topology Poisoning Attacks. And evaluate present security extension TopoGuard.

6) This paper talks about the advantages of the SDN network over the traditional networks with main focus on security aspects of SDN networks. It talks about the possible threats and counter measures.

7) This paper is about analyzing the security challenges present in SDN and propose a comprehensive security architecture is proposed to overcome these challenges.

8) This paper aims at studying SDN accompanied with OpenFlow protocol from the perspective of intrusion and Distributed Denial of Service (DDoS) attacks and suggest machine learning based techniques for mitigation of such attacks.

9) The idea is to secure the interaction layer between control plane and data plane. This is solved by having a security enhanced version of the OpenFlow FloodLight Controller. The end result is to have an authentication system, role-based authorization and other such security features

10) Papers discusses about the issues that exist because of the things that make up SDN and the ways to secure SDN from those attacks. The main areas of concern: - Network programmability. - Control logic centralization.

11) The goal of this paper is to propose a higher standard for intuitive abstraction for programming the SDN called frenetic project. it mainly concentrates on network monitoring traffic, compiling the forwarding packet for policy and real run time policy update and changes.

12) This paper is about susceptibility of SDN to DDoS attacks. Two types of DDoS attacks are emulated using Mininet and the results are shared. It also proposes a security measure to overcome these attacks.

13) This paper shows a new attack to fingerprint SDN networks and further launch efficient resource consumption attacks. This attack demonstrates that SDN brings new security issues that may not be ignored.

The framework has a limitation where the flow features extracted by default are often inadequate to effectively differentiate between normal and attack flows. Additionally, the duration feature used in a previous approach is not appropriate because it represents the duration since the flow entry was installed, not the actual duration of the flow. The use of packet-based detection also leads to high processing and memory overhead. However, these limitations are overcome in this paper by utilizing flow-based detection with the help of ML while also expanding the set of flow features.

III. SDN RECOGNITION AND AVOIDANCE MODEL

To accomplish our objective of constructing a model for identifying and mitigating attacks in SDN networks, we created a model on the application layer. Our proposed mechanism is made up of four primary modules: the Flow Collector module, the Feature Extender module, the Anomaly Detection module, and the Anomaly Mitigation module. The Flow Collector module is in charge of periodically gathering traffic flow data from all switch flow tables. The Feature Extender module is then responsible for generating new characteristics for each flow entry based on this data. The Anomaly Detection module receives this information and employs machine learning to identify and classify flows as normal or anomalous. If a flow is classified as an anomaly, it is forwarded to the Mitigation module, which blocks the source of the attack. If a flow is classified as normal, no action is taken. A diagram of the fundamental elements of our framework is presented in Figure 1, and we will now describe each of these components in greater depth.

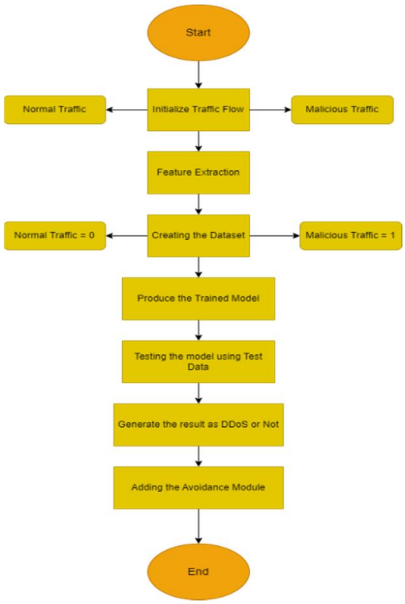


Fig. 1. The fundamental elements of the framework.

A. Traffic Flow Collector Module

At first, the Flow Collector communicates with the controller to obtain traffic data. Once the traffic information request is initiated, the controller utilizes the OpenFlow protocol to retrieve information from flow tables. A message called flow-stats request is sent to each switch that is linked to the controller, which requests flow statistics. Each switch then responds with a flow-stats reply message that contains all flow entries in every flow table, including descriptions of the flow and associated counters. The controller collects traffic data from all switches and returns it to the Flow Collector component. Upon receipt of flow information, the Flow Collector parses the data and eliminates any irrelevant information.

The pertinent data that is preserved includes the flow identifier (flow-id) and the flow counters. The controller collected flow entries from OpenFlow switches at predetermined time intervals. The definition of this time interval is crucial as it impacts the detection and mitigation of attacks. If the intervals are too infrequent, there will be a delay in detecting an attack, and less time will be available for mitigation. Conversely, if the intervals are too short, it will increase the overhead of the detection mechanism. The optimal interval is estimated and dependent on various environmental factors, such as communication delay, network topology, and traffic. In our case, we set the polling interval to 5 seconds to obtain a detailed view of network traffic.

B. Feature Extender Module

The Feature Extender calculates new flow features using the native counters obtained for the current time interval. The reason for generating these extended flow features is that OpenFlow provides a limited number of flow features to describe the traffic profile. Although traffic classification is possible using the native counters, using extended flow features can enhance the performance of classification schemes. Moreover, classifying traffic using only native counters may not detect certain anomalous traffic profiles, hence more detailed traffic discriminators are necessary. Therefore, three new flow features are derived, namely the average flow packet size (pkt_size), the number of flows to the same host as the current flow within the last 5 seconds (same_host), and the number of flows to the same host and port as the current flow within the last 5 seconds (same_host_port).

C. Anomaly Recognition Module

Intrusion detection can be classified in various ways, based on the source of data to be analyzed or the technique used to detect abnormal events. Flow-based or packet-based detection can be used, depending on the data to be analyzed, while signature-based or anomaly-based detection can be used depending on the detection technique. In our case, flow-based detection was preferred over packet-based detection because it is more efficient in terms of processing and memory overhead, making it more suitable for high-speed networks. Anomaly-based intrusion detection was selected as the detection technique, specifically using machine learning (ML). This decision was made because malicious traffic can take various forms and attackers can frequently change their attack patterns. Therefore, it is necessary to learn from experience, which can be accomplished with the help of ML.

The task of detecting attacks can be viewed as a classification problem, where each flow is classified as either normal or anomalous. To accomplish this, we employed an offline machine learning approach. This involves first constructing a model using a classification algorithm, and then using this model to classify network flows. The process of using machine learning for attack detection is depicted in Figure 2. It's worth noting that there are various classification algorithms that can be utilized to build the detection model. In our implementation, we selected the most suitable algorithm among six algorithms: LR, NB, KNN, SVM, DT, and RF, based on certain evaluation metrics.

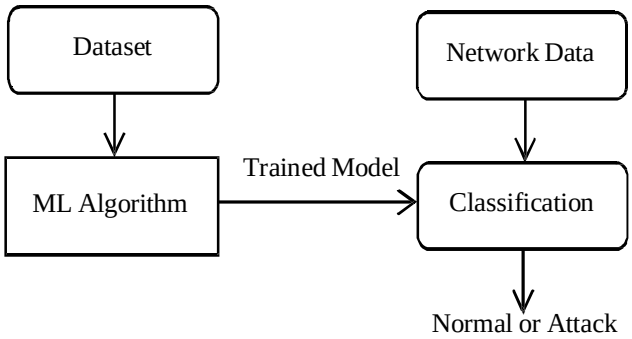


Fig. 2. ML process in SDN.

To use classification learning for attack detection, it is necessary to have a labeled dataset that represents network traffic. Additionally, selecting the appropriate features to train the model is crucial for achieving optimal ML performance. In our approach, we selected five features - pkt_count, byte_count, pkt_size, same_host, and same_host_port - to construct our model.

Once the ML model is constructed, the Detection module receives flows every 5 seconds from the Feature Extender module. This module utilizes the 5 selected features to identify whether an anomaly has occurred for a specific flow-id by utilizing the built model. Normal flows do not need any further action, while malicious flows are passed to the Anomaly Mitigation module for further processing.

D. Anomaly Mitigation Module

The task of the Anomaly Mitigation module is to take measures to prevent network disruption or performance degradation when a malicious flow is detected by the anomaly detection module. The objective of our framework is to hinder the origin of the detected attack. It is important to note that blocking the IP addresses of attackers is not always effective in mitigating attacks because of IP spoofing. In our case, we block the attacker's Ethernet address as a proof-of-concept.

IV. SIMULATION AND RESULTS

Once we have designed our attack detection and mitigation method, it becomes crucial to test and assess its effectiveness. To accomplish this, we have developed a prototype system to demonstrate its practical implementation. The evaluation of the framework comprises assessing the performance of different ML algorithms, comparing our model with the basic model, and evaluating the self-healing model as a whole.

A. Simulation Profile

Creating a dataset is a crucial task in implementing classification ML algorithms. In our case, we had to generate a dataset through simulation as we could not find a suitable public dataset for our approach. Despite researching several publicly available datasets, such as packet-based datasets, datasets without normal traffic data, and unlabeled datasets, they were found to be limited in various ways. The datasets that did not have these limitations contained only a small number of features. Hence, we had to simulate our own dataset.

To evaluate the effectiveness of our proposed method, we set up an SDN topology using the Mininet emulator. To do this, we utilized two Ubuntu virtual machines running on VMWare Workstation. One VM was used for the Floodlight controller, while the other was used for Mininet. Our network consisted of a tree-type configuration with a depth of two, featuring one controller, four OpenFlow v1.3 switches, and nine hosts. Fig. 3 displays the layout of this network.

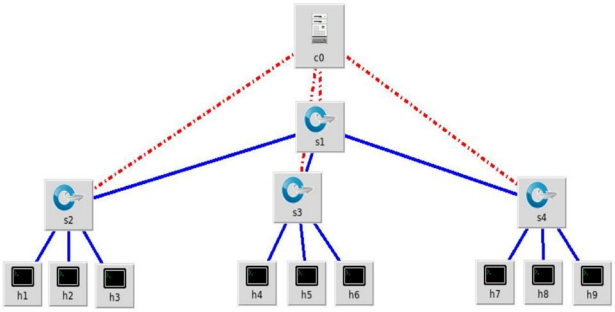


Fig. 3. Network Topology Simulation.

Once the SDN network topology was established, we utilized two programs in Scapy to generate both normal and attack traffic. To be specific, we initiated the normal traffic from five random hosts to three distinct destinations, whereas the attack traffic was created from the leftover host to one of the specified destinations.

B. Intrusion Detection Performance Evaluation

In problems involving binary classification, the confusion matrix is represented by four parameters, which are true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These parameters are defined as follows:

TP represents the number of anomalous flows that are correctly classified as anomalous;

FN represents the number of anomalous flows that are incorrectly classified as normal;

FP represents the number of normal flows that are incorrectly classified as anomalous;

TN represents the number of normal flows that are correctly classified as normal.

To evaluate our anomaly detection approach, we used several metrics including accuracy to determine the performance.

Accuracy refers to the proportion of correctly classified flows in both normal and anomaly classes out of the total number of classified flows. It is expressed as a percentage.

Training time refers to the amount of time taken by the classifier to train the model. Testing time, on the other hand, refers to the time taken by the classifier to test the model.

Table I. Performance Of Different Ml Models

SVM	83.58%
LR	66.02%
KNN	100.00%
DT	100.00%
NB	71.41%
RF	100.00%

In the table above, we see 100% results in some cases, which is practically not possible and these above values came because of limitations of the dataset discussed in the future work.

We observed that KNN, DT, and RF had the highest values for accuracy. Additionally, all the algorithms had low training and testing times, with values less than 1 second. Although training time was not a significant factor for our case, as the model was built before online classification, we aimed for low testing time to reduce the processing overhead of online classification. Therefore, we selected KNN, DT, and RF to construct the training model for our detection module since they outperformed the other algorithms.

V. CONCLUSION AND FUTURE WORK

The research presented in this project makes it easier and more effective to recognize DDoS attacks. It even illustrates the effectiveness of various machine learning techniques as a comparison between six ML algorithms is done viz. Logistic Regression (LR), Naive Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF).

For effective analysis, self-generated dataset was used. It was decided to use KNN, DT, and RF since they produced good results. It's necessary to evaluate both machine learning and mathematical models against actual threats.

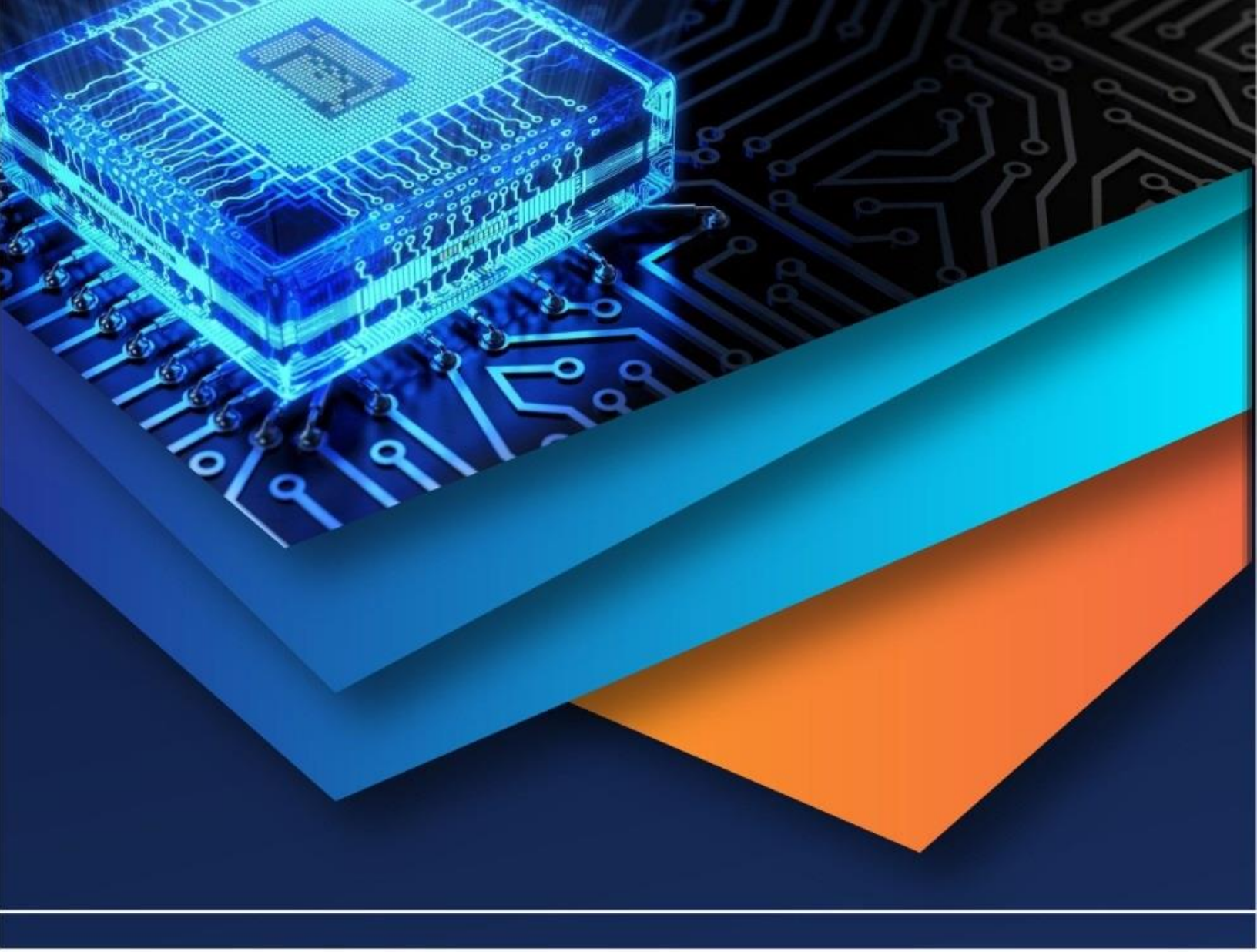
The proposed model only has data from one dataset, which is one of its limitations. As a result, a distributed dataset can be investigated to offer suggestions for future improvements.

The presented work is limited to the network layer that is, it only considers Internet Protocol as a feature for detection of attack. In future, this research can be extended for the inclusion of feature from other layers as well, like MAC address in the data link layer.

In addition, a load balancing module can be implemented on a pool of controller to further distribute the risk and efficiently handle the traffic to the controller.

REFERENCES

[1] Dong Li,Chang Yu, Qizhao Zhou and Junqing Yu .”Using SVM to Detect DDoS Attacks in SDN Network.” 2018 IOP Conf. Ser.: Mater. Sci. Eng. 466 012003,2018 .
[2] Yijie Li, Boyi Liu, Shang Zhai and Mingrui Chen ,”DDoS attack detection method based on feature extraction of deep belief networks.”,IOP Conference Series: Earth and Environmental Science, Volume 252, Issue 3,2019.
[3] Peng Xiao,Wenyu Qu,Heng Qi ,Zhiyang Li.”Detecting DDoS attacks against data centers with correlation analysis.”,Computer Communications 67,2015.
[4] Fatima Khashab, Joanna Moubarak, Antoine Feghali , and Carole Bassil.”DDoS Attack Detection and Mitigation in SDN using Machine Learning”,IEEE 7th International Conference on Network Softwarization (NetSoft),2021.
[5] Sungmin Hong, Lei Xu, Haopei Wang, Guofei Gu "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures", 7 Feb 2015.
[6] Software-Defined Networking Security: Pros and Cons, IEEE Communications Magazine — Communications Standards Supplement, June 2015 0163-6804/15/$25.00 © 2015 IEEE
[7] A comprehensive security architecture for SDN, Zhiyuan Hu; Mingwen Wang; Xueqiang Yan; Yueming Yin; Zhigang Luo, DOI: 10.1109/ICIN.2015.7073803 Publisher: IEEE
[8] Handling Intrusion and DDoS Attacks in Software Defined Networks Using Machine Learning Techniques, Javed Ashraf; Seemab Latif, DOI: 10.1109/NSEC.2014.6998241 Publisher: IEEE
[9] "Securing the Software-Defined Network Control Layer" , Phillip Porras, Steven Cheung, Martin Fong, Keith Skinner, and Vinod Yegneswara
[10] Towards Secure and Dependable SDN, Diego Kreutz , Fernando M.V. Ramos , Paulo Verissimo
[11] Languages for Software-Defined Networks, Nate Foster; Arjun Guha; Mark Reitblatt; Alec Story, 201
[12] Denial-of-Service Attacks in OpenFlow SDN Networks, Rajat Kandoi; Markku Antikainen, 2015
[13] Role-Based Multiple Controllers for Load Balancing and Security in SDN, Dharmendra Chourishi; Ali Miri; Mihailo Milić; Salam Ismaeel, IEEE, 2015



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY