

Secure flow authentication and CNN-based detection for DDoS mitigation in SDN

Monika Dandotiya[‡]

Ajay Khunteta[†]

Kriti Sankhla[§]

Department of Computer Science & Engineering

Poornima University

Jaipur 303905

Rajasthan

India

Manoj Kumar Sharma^{*}

Aditya Narayan Hati[@]

Department of Computer and Communication Engineering

Manipal University Jaipur

Jaipur 303007

Rajasthan

India

Abstract

By separating control plane from data plane, Software-Defined Networking has revolutionized network management by facilitating centralized control and increased network adaptability. DDoS in particular, are security flaws introduced by SDN that have the potential to seriously impair network services. This study presents a novel two-phase method for using CNN to increase the security of SDN settings. Establishing a secure authentication method that stops unauthorized nodes from accessing network services is the first step in reducing the risk of unauthorized access. The identification and mitigation of DDoS attacks launched by verified nodes is the focus of the second phase. For real-time traffic analysis, a CNN-based detection system is developed, trained on multiple datasets, including CICDDoS2019, and integrated into the SDN controller. Comprehensive tests utilizing simulation tools show how effective the suggested methodology is at improving

[‡] E-mail: dandotiyamonika@gmail.com

[†] E-mail: Ajay.khunteta@poornima.edu.in

[§] E-mail: kriti.sankhla@gmail.com

^{*} E-mail: manojkumar.sharma@jaipur.manipal.edu (Corresponding Author)

[@] E-mail: aditya.narayan@jaipur.manipal.edu

detection accuracy, lowering false positive rates, and preserving network performance. This study provides a flexible, effective, and reliable approach to enhance DDoS detection and mitigation in SDN systems.

Subject Classification: 68M25, 68P25.

Keywords: SDN, DDoS, Attacks detection, Cyber security, Classification, Deep learning, Convolutional neural network (CNN).

1. Introduction

Due to the fast growth of internet technology over the past several years, upgrading to new network functions has proven difficult due to the old network's dispersed control and management approach [1]. This has made the implementation of network innovation difficult [2]. To improve the flexibility and centralization of network control, it also makes use of the control layer's centralised management mode. SDN has been a popular option for cloud computing, wireless local area networks, network virtualization, and cloud data centers. The security issues associated with SDN have become more prevalent, despite the fact that its flexible network architecture offers convenience for network management. OpenFlow is a protocol that precise the communication parameters between SDN controllers and SDN switches in SDN. Attackers generate a large number of bogus IP packets by exploiting this working mode's vulnerability. The fake packets can't identify the corresponding flow rules in the switch's flow table, so the switch creates a bunch of packet_in messages and sends them to the controller. There are presently numerous techniques for detecting and preventing DDoS attacks in SDN [3][4][5]. There are currently insufficient studies on DDoS defense for SDN controllers, though. To begin with, the majority of detection methods in use today are periodic. Determining an appropriate detection period is challenging, though. The response time will be prolonged if the selected period is very long. Second, DDoS attacks will generate a lot of packet_in messages quickly, which will drain the controller's resources and make it hard for the detection equipment to function. As a result, the mitigation controller's timely unclogging of packet_in messages have to be resolved. Thirdly, because the flow of a DDoS assault is remarkably similar to that of a normal burst flow [6], detecting DDoS attacks is getting harder and harder. Fourth, traditional methods of DDoS attacks detection and mitigation is based on a single controller and not directly applicable to a situation involving many controllers. It is difficult to put in place efficient defenses

against widespread distributed denial-of-service assaults in the multi-domain controller situation [7]. Numerous ML and deep learning methodologies have been implemented to identify DDoS attacks; however, deep learning (DL) demonstrated superior performance when contrasted with machine learning techniques. Consequently, the deep learning method is employed in this paper for the purpose of classification. A Deep Learning (DL) technique is currently employed in a variety of computer science disciplines. One example is the widespread use of LSTM in speech recognition and natural language processing (NLP). Machine Learning (ML) models utilise a limited number of packet features; however, the accuracy and outcome results undergo a decline as the number of packet features increases. A deep learning algorithm is capable to create high-quality results by analysing a large number of feature values. The DL model is also capable of generating new features and scanning the data to identify the correlated feature. Additionally, DDoS attacks are regarded as one of the most perilous attacks that impede performance of SDNs. The control plan of the routers is held by attackers, which makes it challenging for the SDN to manage the packets and transfers control to the attackers. The goal of this research is to create a detection method that can identify and mitigate DDoS attacks in real-time using CNNs' capabilities. The proposed methodology is divided into two phases: the initial phase entails the establishment of an authentication mechanism to prevent unauthorised nodes from accessing the network, while the subsequent phase concentrates on the detection and mitigation of DDoS attacks initiated by authenticated nodes. The primary contributions of the proposed approach are the classification of DDoS attack traffic and legitimate traffic, which is achieved by managing a greater number of features from DDoS attacks. This is done to improve the detection of attacks in SDN through the use of a deep learning framework. Our contributions to this paper are summarised below.

- There are three main parts to how we use CNNs in DL to find DDoS attacks in SDN. First, we come up with a safe and useful way to check users that make sure only real nodes can use network services.
- We examine this mechanism to assess its impact on throughput, reliability, and network overhead to identify the optimal method for mitigating performance degradation. After that, we make a detection system that uses CNNs to find DDoS attacks. This means creating a

complete attack database for training the model, which lets the CNN tell the difference between good and bad traffic.

- The SDN controller uses the trained model to help find and fix problems as they happen. Finally, we use simulation tools to run full tests that check the speed of authentication, throughput, dependability, and network overhead. Comparing our results to those of other methods shows how our approach has strengths and weaknesses that could be fixed in the future.

2. Literature Review

Over the past few years, SDN has been often described as the “future of networking.” The main cause of this technology’s explosive growth and widespread success is the separation of networking devices’ control plane and data plane. Configuring each networking device is no longer necessary thanks to the control plane, which allows for the centralized administration of the entire network and switches. The switching devices that make up the data plane allow network traffic to be forwarded in compliance with the rules set by the applications running above the programmable controller. This significant development in networking has made it easier to speed up service delivery and given more flexibility and agility when it comes to provisioning virtual and physical network devices from a central place. [8]. There have been numerous strategies for intrusion detection systems proposed by scientists in recent years, but there are scarcely any procedures for anomaly detection. The intricacy and rapidity of contemporary malevolent behaviour, as well as the scale of contemporary networks, present a challenge to these strategies in the development of a diverse, adaptable, and uncomplicated approach to aberrant behaviour detection. Various intrusion detection techniques [9] [10]. In 2015, we can detect DDoS attacks by integrating 2 IDS methods, as we have progressed to the next research phase [11]. The detection accuracy has been enhanced by these studies; however, the computational complexity and resource consumption have increased. Then, a study was conducted to identify and select appropriate traits that reduce the detection time and simplify the process [12]. Another methodology [13] forecasted DDoS attacks using a variety of time series techniques by using the anomaly scores to anticipate the behavior of the traffic attributes. Based on the scores, the framework classified traffic as either normal or an assault. A novel technique for identifying DDoS attacks utilizing Lattice Structural access rates is presented in this study [14,15]. In [16], a framework for detecting DDoS

attacks in SDN was constructed using a Spark tool. Out of all the methods, the ML method has the highest accuracy, scoring 91.6%. For real-time deployment, DT was thus selected. The well-known LSTM model was used in another study [17] to find undesirable flows in the advancement of distributed SDN-based edge computing. The study by [18] developed a three-model software-defined networking-based classification system, using a gated RNN “as their primary model [19].

2.1 Research Gap

Despite the enormous advancements in the field, there are still a number of research gaps in software-defined networking (SDN) intrusion detection systems (IDS). Many techniques prioritize intrusion detection, while anomaly detection receives little attention. These current tactics are regularly challenged by the complexity and speed of modern malicious behavior as well as the size of modern networks. Conventional approaches, such as knowledge-based techniques, statistical analysis, machine learning, and mining, often produce significant false positive rates when used separately. While combining different intrusion detection techniques has improved detection accuracy, investigations up to 2015 have shown that this has also led to increased computing complexity and resource consumption. By focusing on feature selection and time series techniques for the prediction of DDoS attacks, further research has attempted to streamline and expedite the detection process. But in dynamic SDN settings, these methods often don't give real-time apps the level of reliability they need. Numerous investigations have examined deep learning models. But these models usually need a lot of processing power and complicated ways to choose features, which makes them not good for finding DDoS attacks in SDN in real time. We need more flexible and varied ways to find things because focusing on specific datasets like NSL-KDD, CICIDS2017, and CICDDoS2019 may not work in every network situation. This paper employs CNNs to rectify the deficiencies in existing DDoS detection methodologies. It's hard to find patterns in network traffic data, but CNNs are great at it. CNNs are great for real-time detection in the world of SDN, which is always changing, because they can quickly adapt to new and changing attack patterns. On the other hand, older methods mostly use features that have already been set. We want to use CNNs to help us find DDoS attacks on SDNs in real time as quickly, accurately, and easily as possible. Not only does our method fill in the gaps in the research we found, but it also gives a more complete and

adaptable solution to the problems that DDoS attacks cause in SDN systems.

3. Research Methodology

This part talks about the work that has been proposed to find DDoS attacks in SDN. The work makes the field better by creating a deep learning-based model for classification and improving detection accuracy. This section also talks about a flowchart that was suggested.

3.1 Problem Statement

By separating the data plane from the control plane, Software-Defined Networking (SDN) makes it easier to manage networks. This gives you more choices and makes it easier to manage everything from one place. This design does make the system more open to attacks, especially Distributed Denial of Service (DDoS) attacks, which can bring down network services and slow down performance. There are many different ways to find anomalies and intrusion detection systems (IDS), but the ones that are already out there often have higher false positive rates, are harder to use, and don't adapt well to new attack patterns. Right now, we need a DDoS detection system that is fast, dependable, and works in real time. It should also work well in the world of SDN, which is always changing. This research aims to address these issues by developing a detection system utilizing a CNN that employs automated feature extraction and adaptability to enhance the accuracy and efficacy of DDoS detection in SDN.

3.2 Proposed Methodology

This paper has a safe way to log in that keeps unauthorized nodes from using network services. This first step is very important because it makes sure that only nodes that have been checked can do things on the network. This makes it less likely that someone will break in or attack. The authentication process was designed to be quick and safe at the same time, so that high security standards can be met while still being quick. We looked into how this mechanism affects the network's speed, how well it finds real nodes, and how much more work it makes the network do. We are now focusing on finding and stopping DDoS attacks that come from authenticated nodes. We need to make a complicated attack detection system that uses CNNs right now. We began by putting together a full

attack database using several datasets, including CICDDoS2019, which has patterns of both good and bad traffic. It was necessary to preprocess the data to make sure the CNN model worked as well as it could and was always the same. This step involved making the data more consistent and getting rid of any features that weren't needed. We have developed an architecture that is specifically designed for network traffic analysis in the CNN model. This entailed the optimisation of the number of layers, filter diameters, and activation functions to achieve a balance between computational efficiency and accuracy. In order to identify the most pertinent features from the original dataset, we implemented feature selection technique, including filter-based, wrapper-based, and embedded-based approaches. This has led to a reduced feature set that improves the model's performance without compromising its capacity to detect DDoS attacks. The preprocessed datasets were employed to train the CNN model after it was developed. With the integration of the trained model into the SDN controller, it is now possible to monitor and analyse network traffic in real time. A real-time feature extraction mechanism has been devised to input network traffic data into the CNN model, enabling the immediate detection. We employed simulation tools to establish a controlled SDN environment in order to verify the efficacy of our methodology. This environment generated both normal and attack traffic, which allowed for a comprehensive assessment of the CNN-based detection system. The results were also compared to existing DDoS detection methods to show how much better they are in terms of real-time use, computational efficiency, and accuracy of detection. Based on the results of the evaluation,

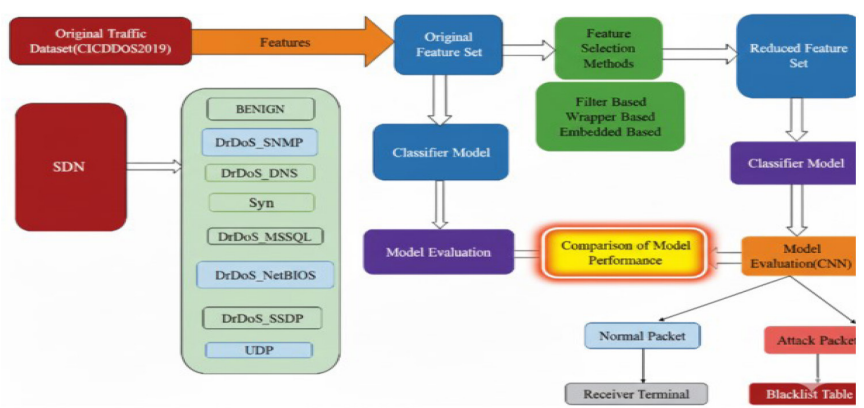


Figure 1
Illustration of the Proposed Methodology

we looked at areas where we could make things better. These included improving the architecture of the CNN model and making the process of extracting features more efficient. Ongoing improvements and the ability to adapt to new attack patterns have come from iterative refinements. In short, we talked about our findings and how well the CNN-based detection system worked to make DDoS detection better in SDN. We talked about possible future research paths, such as looking into other deep learning methods or making the model bigger so it can handle more types of network attacks. We used this thorough research method to make a DDoS detection system that is strong, flexible, and works well. This fixes the problems with the current methods and makes SDN environments safer.

Figure 1 shows the flowchart for the suggested method. When thinking about different ways to authenticate, keep in mind things like network overhead, the speed (or complexity) of the process, and throughput, all while making sure it is reliable. Think about the results in light of other choices.

3.2.1 Data Gathering

The first thing to do is get data from the network. There is both real and fake traffic on the network in the CICDDoS2019 dataset. To find DDoS attacks effectively, the CNN model needs to be trained and tested.

3.2.2 Data Preprocessing

It is important for getting useful and correct results in the field of machine learning. Data preprocessing makes data better by getting rid of noise, fixing inconsistencies, and filling in missing pieces. The preprocessing stage is made up of the steps that follow.

3.2.3 Ensure Consistency

To make sure that the data is consistent, it is pre-processed. This means getting rid of any extra or unimportant features that don't help find DDoS attacks, which clean up the dataset.

- **Normalization:** Normalize the data to ensure that all features are of the same scale. Normalization facilitates the optimal performance.
- **Feature Selection.** The main goal of feature selection is to keep the most useful dimensions in a dataset while getting rid of the rest. Feature selection makes the model work better, lowers the risk of overfitting, and makes it easier to calculate. The CNN model is more accurate and reliable at finding

things when you make sure it only learns the most important patterns from the data. There are a lot of ways to choose features, and each one has its own good and bad points.

- **Filter-based Methods:** Statistical metrics, like how well each feature correlates with the target variable (like a DDoS attack or normal traffic), are used to figure out how important each feature is.
- **Wrapper-based Methods:** These methods test and train the model over and over again on different sets of features. Two examples are recursive feature elimination (RFE) and forward/backward selection. Embedded-based approaches include feature selection in the process of training the model. Two examples are lasso regression and feature importance based on decision trees.

3.2.4 Design and Development of the CNN Model

Create an Attack Database. By building the attack database, you can create a comprehensive database that lists all of the different known patterns of DDoS attacks. This information serves as the basis for training the CNN model to correctly identify and categorize different types of attacks.

3.2.5 Design of the CNN Architecture

We have put into practice CNN architecture specifically created for network traffic analysis. In a range of computer vision and image processing tasks, CNNs a particular kind of neural network have shown remarkable performance. Among CNN's more intriguing application areas include object identification, video processing, speech recognition, natural language processing, and picture classification and segmentation.

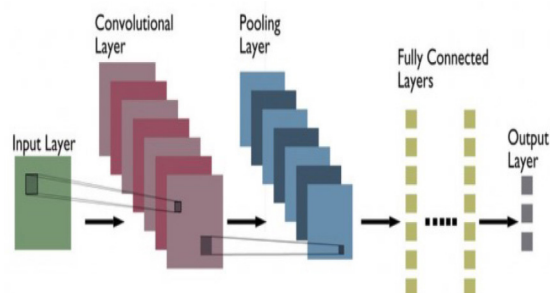


Figure 2
Convolutional Neural Network

Deep CNN's high learning capacity is a result of its numerous feature extraction stages, which may independently learn representations from data. CNNs are a group of deep neural networks that are most frequently employed for processing visual information in the field of deep learning. CNN models are state-of-the-art for image processing applications as shown in figure 2 such as object detection, segmentation, and classification. Understanding a few well-known CNN designs can help us stay ahead of the competition in this cutthroat and cutthroat industry, whether we're trying to break into the image processing space or improve the prediction accuracy of our own CNN models. One type of deep learning algorithm used to analyze data in a grid-like format is CNN.

In the architecture illustrated above in figure 2, the subsequent definitions pertain to distinct layers:

Convolutional Layer: When applied to an input picture, a group of filters known as kernels make up convolutional layers. To build more sophisticated models that can extract more intricate information from images, convolutional layers can be overlaid. Pooling also facilitates the reduction of the number of parameters and speeds up the training process. The two primary forms of pooling are average pooling and maximum pooling. The maximum value is extracted from each feature map using max pooling, whereas the average value is extracted using average pooling. Convolutional layers are frequently followed by pooling layers, which lower the quantity of input before it is sent into a fully linked layer.

4. Evaluation and Testing

This section presents the evaluation and testing of the study with different subsections i.e., simulation tools, performance metrics, and experimental results.

4.1 *Simulation Tools*

Simulation tools are implemented to establish a controlled SDN environment in order to comprehensively assess the performance of a CNN-based DDoS detection system. These tools simulate real-world network conditions by generating both normal and malicious traffic. Python's simulation configuration is significantly influenced by its extensive libraries, including Mininet for network simulation and Scikit-learn or TensorFlow for machine learning. Mininet facilitates the simulation of a fully functional SDN environment, allowing for the testing of the

interaction between network devices and the SDN controller in both normal and attack scenarios. The generation of network traffic, injection of DDoS attack patterns, and real-time monitoring of the system’s responses are all scripted using Python. The CNN model’s overall robustness, false positive rate and detection accuracy are all evaluated in this controlled simulation environment to guarantee dependable performance prior to deployment in actual network scenarios.

4.2 Performance Metrics

Comprehensive assessments are conducted using simulation tools to validate the effectiveness of the proposed methodology. The evaluation underscores several essential criteria:

- **Authentication Process Speed:** The process is guaranteed to be both secure and rapid by measuring the time required to authenticate nodes.
- **Throughput:** After authentication, the data transfer rate is evaluated to guarantee that the method does not substantially impede network performance and maintains a high throughput.
- **Reliability:** The authentication mechanism’s consistency and accuracy are assessed to guarantee that it consistently identifies legitimate nodes and prevents unauthorised access.
- **Network Overhead:** The authentication and detection processes introduce additional data processing and transmission, which is determined with the objective of minimising overhead.

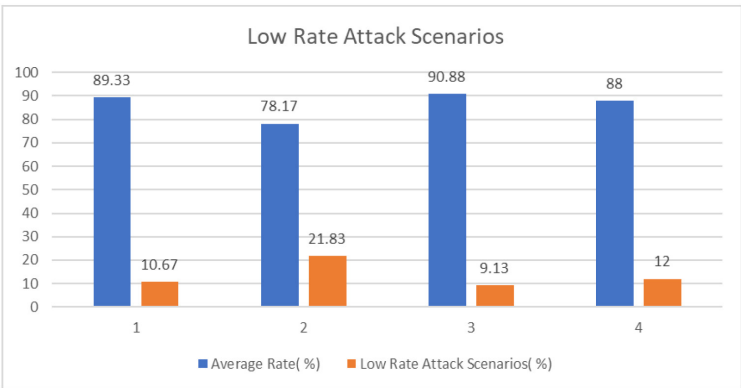


Figure 3
The low-rate SDN-based DDoS attack scenarios

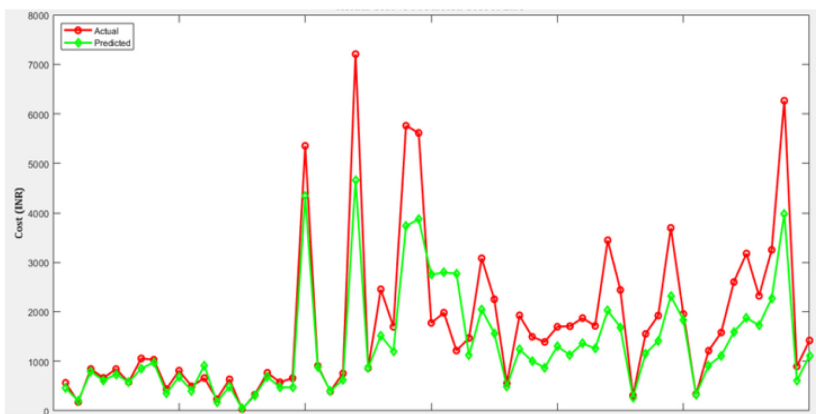


Figure 4
Comparison of Observed and Predicted Network Traffic During SDN-Based DDoS Attacks

4.3 Experimental Results

This SDN-based DDoS detection technique ought to be more appropriate for real-world business applications, such intricate process interactions or a multitude of potential solutions that call for extended simulations, a bigger population of optimization candidates, and more optimization iterations.

The graphic above shows the Low Rate Attack Scenarios of SDN-based DDoS. The difference in DDoS attack traffic rates in each scenario allows for the fluctuation between the average rate and low-rate assault conditions, as illustrated in figure 3.

Figure 4 illustrates a comparative analysis between the actual network traffic observed during an SDN-based DDoS attack and the expected

Flow ID	Src IP	Dst IP	Src Port	Dst Port	Protocol	Timestamp	Flow Duration (m)	Tot Fwd Pkts	Tot Bwd Pkts	Flow Pkt Len Mean	Flow Pkt Len Max	Label
b7c000000	192.168.1.132	10.0.0.162	59343	8080	TCP	2017-08-01 00:01	40767	1486	791	589.1	169.8	BENIGN
b7c000001	192.168.1.151	10.0.0.205	60174	53	TCP	2017-08-01 00:06	9702	1751	216	453.3	544.3	BENIGN
b7c000002	192.168.1.54	10.0.0.59	1213	8080	TCP	2017-08-01 00:08	45768	1276	763	78.9	949.6	UDP_DDoS
b7c000003	192.168.1.3	10.0.0.23	20142	80	UDP	2017-08-01 00:09	6883	1017	187	326.4	945.7	BENIGN
b7c000004	192.168.1.89	10.0.0.237	36797	443	UDP	2017-08-01 00:12	5685	1726	379	1013.2	844.2	TCP_DDoS
b7c000005	192.168.1.237	10.0.0.90	54834	123	UDP	2017-08-01 00:12	171	296	492	1118.8	413.6	BENIGN
b7c000006	192.168.1.159	10.0.0.50	2923	80	TCP	2017-08-01 00:14	37075	1749	1064	872.0	803.3	HTTP_DDoS
b7c000007	192.168.1.39	10.0.0.220	57910	123	TCP	2017-08-01 00:16	26567	720	1180	1301.2	560.8	TCP_DDoS
b7c000008	192.168.1.131	10.0.0.60	2291	5201	TCP	2017-08-01 00:20	33773	1773	14	932.4	1361.5	UDP_DDoS
b7c000009	192.168.1.193	10.0.0.171	32575	443	UDP	2017-08-01 00:21	32616	338	64	1286.0	1194.5	UDP_DDoS
b7c00000a	192.168.1.189	10.0.0.221	51704	443	TCP	2017-08-01 00:23	11544	879	520	1199.1	1021.6	BENIGN
b7c00000b	192.168.1.22	10.0.0.189	12418	80	TCP	2017-08-01 00:25	29137	1077	1367	662.3	360.9	UDP_DDoS

Figure 5
Assessment of CICDDoS2017 Dataset Model

Table 1
DDoS attack detection techniques with the best ratios

Studies	Datasets	Detection Rate (%)
[20,21]	TUIDS CAIDA, & DARPA	98.32
[22,23]	CAIDA 2007,DARPA 2009	96.21
[24-26]	KDD Cup	95.12
[27-30]	Knowledge Discovery	93.14

traffic predicted by the system model. The expected traffic values are estimated using historical traffic patterns and behavioral features extracted from previous DDoS incidents. The results indicate that the prediction model can reliably forecast traffic trends and detect significant anomalies associated with DDoS activities in a Software-Defined Networking environment.

The outcomes of the model evaluation for the CICDDoS2017 Dataset are shown in Figure 5. The detection approach demonstrated better performance than competitors in recognizing high-rate DDoS assaults in all simulated situations. This advantage was explained by the high traffic rate in addition to the method's ability to gather more statistical data about freshly arriving network traffic. The above graphic displays the duration of the actual value in comparison to the expected value. The network's DDoS attack rate is influenced by the quantity of attack sources and how they alter the rate at which they transmit attacks. One host attack or multiple host assaults (many attackers) have the ability to start a distributed denial of service (DDoS) attack.

Using various datasets, various reports have stated on the detection rates of various IDS. These rates are summarized in Table 1. The results demonstrate the model's good detection performance across many datasets, with the first study achieving the highest detection rate in comparison to the others. Two of studies employ the KDD Cup 1999 dataset, and despite slight variations in detection rates, the results consistently indicate the value of assessing IDS.

5. Conclusion and Future Scope

The next step was to use CNN to make a better way to find DDoS attacks. Adding this model to the SDN controller made it possible to watch and analyze network traffic in real time. This made it easier to quickly find and sort through possible threats. In a controlled SDN setting, extensive

testing demonstrated that the CNN-based detection system exhibited superior accuracy, reduced false positives, and enhanced computational efficiency compared to alternative methods. The system changed all the time, which helped it deal with new types of attacks. The results show that the proposed method is very good at finding and stopping DDoS attacks in SDN environments. The favourable results of this study indicate multiple opportunities for subsequent research. The CNN model architecture and the feature extraction method could be improved to make detection more accurate and useful. There may be ways to make DDoS detection even better by looking into other deep learning techniques, such as hybrid models that mix Convolutional Neural Networks (CNNs) with other algorithms or Recurrent Neural Networks (RNNs). The detection system would be safer in general if it could find other types of network attacks. Finally, we could learn a lot about how well the system works and how strong it is by putting it through big field tests and using it in real SDN situations. Because security is always changing, the DDoS detection and mitigation system will need to be watched and changed all the time to stay useful.

References

- [1] V. G. Nguyen, A. Brunstrom, K. J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1567–1602 (2017), doi: 10.1109/COMST.2017.2690823.
- [2] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 897–912 (2019), doi: 10.1109/TIFS.2018.2866244.
- [3] M. Dandotiya and R. R. S. Makwana, "DDoS attack detection and mitigation in SDN environment: A deep learning perspective," in *Proc. IEEE Int. Conf. on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior, India, pp. 1–6 (2024), doi: 10.1109/IATMSI60426.2024.10502843.
- [4] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, Art. no. 102942 (2021), doi: 10.1016/j.jnca.2020.102942.

- [5] Y. Cao, H. Jiang, Y. Deng, J. Wu, P. Zhou, and W. Luo, "Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3571–3585 (Nov.–Dec. 2022), doi: 10.1109/TDSC.2021.3083075.
- [6] M. Dandotiya and R. R. S. Makwana, "Improving network security with hybrid model for DDoS attack detection," in Proc. IEEE Int. Conf. on Blockchain and Distributed Systems Security (ICBDS), Pune, India, pp. 1–8 (2024), doi: 10.1109/ICBDS61829.2024.10837036.
- [7] M. Dandotiya, A. Khunteta, and R. R. S. Makwana, "Enhancing SDN security: Mitigating DDoS attacks with robust authentication and Shapley analysis," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 1, pp. 249–265 (2025), doi: 10.47974/JDMSC-2219.
- [8] V. K. Singh, "DDoS attack detection and mitigation using statistical and machine learning methods in SDN," M.S. thesis, College of Ireland, Dublin, Ireland (2020).
- [9] M. Dandotiya, P. Rahi, A. Khunteta, A. Anushya, and S. S. Ahmad, "SAFE: A secure authenticated and integrated framework for e-learning," in Proc. 4th Int. Conf. on Information Management & Machine Intelligence (ICIMMI), New York, NY, USA: ACM, Art. no. 89, pp. 1–9 (2023), doi: 10.1145/3590837.3590926.
- [10] M. Dandotiya and R. R. S. Makwana, "Secured DDoS attack detection in SDN using TS-RBDM with MDPP-Streebog based user authentication," *Transactions on Emerging Telecommunications Technologies*, vol. 36, Art. no. e70052 (2025), doi: 10.1002/ett.70052.
- [11] M. A. Ribeiro, M. S. P. Fonseca, and J. de Santi, "Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks," *Computers & Security*, vol. 124, Art. no. 102974 (2023), doi: 10.1016/j.cose.2022.102974.
- [12] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26 (2019), doi: 10.2478/cait-2019-0001.
- [13] R. F. Fouladi, O. Ermiş, and E. Anarim, "A DDoS attack detection and defense scheme using time-series analysis for SDN," *Journal of Information Security and Applications*, vol. 50, Art. no. 102402 (2020), doi: 10.1016/j.jisa.2019.102402.

- [14] K. Karthick, G. Kiruthiga, P. Ms Saraswathi, B. Dhiyanesh, and R. Radha, "Subset scaling recursive feature collection based DDoS detection using behavioural based ideal neural network for cloud security," *Int. J. Cloud Applications and Computing*, vol. 12, no. 1, pp. 1–16 (2022).
- [15] H. Alubaidan, R. Alzaher, M. AlQhatani, and R. Mohammed, "DDoS detection in software-defined network using machine learning," *International Journal of Cybernetics and Informatics*, vol. 12, no. 3, pp. 23–36 (2023).
- [16] S. S. Samaan and H. A. Jeiad, "Feature-based real-time DDoS detection in SDN using machine learning and Spark," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1089–1098 (2023), doi: 10.11591/eei.v12i2.4531.
- [17] H. Zhou, Y. Zheng, X. Jia, and J. Shu, "Collaborative prediction and detection of DDoS attacks in edge computing using distributed SDN," *Computer Networks*, vol. 227, Art. no. 109708 (2023), doi:10.1016/j.comnet.2023.109708.
- [18] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in SDN-based networks using deep recurrent neural networks," in *Advanced Sciences and Technologies for Security Applications*, M. Alazab and M. Tang, Eds., Springer, pp. 175–195 (2019), doi:10.1007/978-3-030-00024-5_9.
- [19] T. Khempetch and P. Wuttidittachotti, "DDoS attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 358–367 (2021), doi: 10.11591/ijai.v10.i2.pp358-367.
- [20] M. Al-Jarah, M. Al-Shurman, and B. Al-Duwairi, "Attention-based deep learning approach for detecting IoT botnet-based DDoS attacks," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 6, pp. 1785–1815 (2024), doi: 10.47974/JDMSC-1729.
- [21] T. Peng, C. Leckie, and K. Ramamohanarao, "Proactively detecting distributed denial of service attacks using source IP address monitoring," in *Lecture Notes in Computer Science*, vol. 3224, Springer, pp. 177–189 (2004), doi: 10.1007/978-3-540-24693-0_63.
- [22] G. Mittal, S. Kumar, S. Kumar, S. Narayan, and Y. K. Gupta, "Secure public key encryption and key encapsulation mechanism based on group ring learning with errors problem," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 7, pp. 2797–2810 (2025), doi:10.47974/JDMSC-2131.

- [23] M. Dandotiya, A. S. Dandotiya, N. Dandotiya, and A. Sahu, "A secure detection framework for ARP, DHCP, and DoS attacks on Kali Linux," *International Journal for Research in Applied Science & Engineering Technology*, vol. 10, no. 7, pp. 3044–3053 (Jul. 2022), doi:10.22214/ijraset.2022.42176.
- [24] S. V. Joshi, N. Wagh, J. R. R. Kumar, D. Dongre, and A. P. Baligar, "Mitigating DDoS attacks using an intrusion detection and prevention system based on two-player Bayesian game theory," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2-B, pp. 809–820 (2024), doi:10.47974/JDMSC-1957.
- [25] O. Wardak, S. Kumar, and D. Sinha, "Encryption and decryption scheme through signed Cayley graph," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 7, pp. 2811–2825 (2025), doi: 10.47974/JDMSC-2201.
- [26] Z. S. Alsham, E. Bahçekapılı, and A. Ayaz, "Trends in IoT applications in smart campuses: A topic modeling approach," *COLLNET Journal of Scientometrics and Information Management*, vol. 19, no. 1, pp. 21–40 (2025), doi: 10.47974/CJSIM-2024-017.
- [27] G. Mustafa and M. T. Afzal, "Formulating rules to identify key researchers in computer science: A quantitative approach," *COLLNET Journal of Scientometrics and Information Management*, vol. 19, no. 1, pp. 91–122 (2025), doi: 10.47974/CJSIM-2024-11007.
- [28] S. Gautam, S. S. Gaur, and S. Masood, "A comparative study of certificate and certificate-less cryptographic algorithms and their energy consumption analysis in wireless sensor networks," *TARU Journal of Computer Dynamics & Technological Advances*, vol. 1, no. 2, pp. 85–100 (2019), doi: 10.47974/2019.TJCDTA.004.
- [29] B. Farzana, R. Rafiya, A. Rahman, I. Siddiq, and Sameena, "Frontiers in fintech risk management: A collaborative research platform for a secure future," *Journal of Information and Optimization Sciences*, vol. 46, no. 8, pp. 2423–2433 (2025), doi: 10.47974/JIOS-2057.
- [30] K. Sharma, S. Bhatt, and M. Khari, "Empirical study on network exploitation tools and vulnerability detection tools," *Journal of Information and Optimization Sciences*, vol. 46, no. 5, pp. 1439–1458 (2025), doi: 10.47974/JIOS-1508.

Received September, 2025