Tanzania Journal of Engineering and Technology 2022, **41**(**3**):51-63
OPEN ACCESS articles distributed under Creative Commons Attribution Licence [CC BY-ND]
Websites: https://ajol.org/tjet; https://tjet.udsm.ac.tz

ISSN 1821-536X (**print**); ISSN 2619-8789 (**electronic**)
https://doi.org/10.52339/tjet.v41i3.844

Special Issue – Sixth Tanzania Annual ICT (TAIC) Conference, 26 – 28, October 2022, Zanzibar, TANZANIA

Isolation of DDoS Attacks and Flash Events in Internet Traffic Using Deep Learning Techniques

Carl E. Mihanjo[1] **and Alex F. Mongi**[2†]

[1]Information and Communication Technology Unit, Ministry of Minerals, Dodoma, Tanzania

[2]Department of Electronics and Telecommunication Engineering, University of Dodoma, Dodoma, Tanzania

[†]Corresponding author: alex.mongi@udom.ac.tz; https://orcid.org/0000-0002-2466-7497

ABSTRACT

The adoption of network function visualization (NFV) and software-defined radio (SDN) has created a tremendous increase in Internet traffic due to flexibility brought in the network layer. An increase in traffic flowing through the network poses a security threat that becomes tricky to detect and hence selects an appropriate mitigation strategy. Under such a scenario occurrence of the distributed denial of service (DDoS) and flash events (FEs) affect the target servers and interrupt services. Isolating the attacks is the first step before selecting an appropriate mitigation technique. However, detecting and isolating the DDoS attacks from FEs when happening simultaneously is a challenge that has attracted the attention of many researchers. This study proposes a deep learning framework to detect the FEs and DDoS attacks occurring simultaneously in the network and isolates one from the other. This step is crucial in designing appropriate mechanisms to enhance network resilience against such cyber threats. The experiments indicate that the proposed model possesses a high accuracy level in detecting and isolating DDoS attacks and FEs in networked systems.

ARTICLE INFO

First submitted: June 29, 2022

Revised: Sep. 29, 2022

Presented: Oct. 26 - 28, 2022

Accepted: Nov. 15, 2022

Published: Nov. 30, 2022

Keywords: *DDoS attacks, flash events, network, deep learning*

INTRODUCTION

The exponential increase of multimedia content and cloud-based applications need flexible and reliable communication networks to dynamically allocate resources as the need arises. As per the Visual Network Index (VNI) report of 2020, the global Internet Protocol (IP) traffic will increase threefold in 2022 reaching an annual rate of 1.5 Zettabytes per year or 122 Exabytes per month (Cisco 2020). Projected growth requires a programmable architecture capable of offering high bandwidth and secured networks. This can be achieved by introducing intelligence in network management systems with the help of technologies such as Software-Defined Networks (SDN) and Network Function Virtualization (NVF). The mentioned technologies promote the convergence of services, applications, and networks over IP which is one of the features of Next-Generation Networks (NGN). SDN allows users to define data rules that control the flow of packets and the utilization of network resources such as storage, computing, and bandwidth (Nadeau & Gray, 2013). It consists of three parts which are application, control, and data planes. The application plane abstracts the underlying network to applications using the northbound Application Programming Interface (API). The abstraction level includes parameters such as packet delay, network throughput, and system availability to cover a wider view of the network (Guy, 2015). Normally, the applications request a connection between end nodes and the network for the services to transfer instructions of the parameters to the SDN controller. The parameters are configured in the data plane for effective packet or traffic forwarding. Often, the data plane is composed of physical and/or virtual switches that are responsible for forwarding the packets to the SDN controllers. The connection is made by using the API between the control plane and the data plane (southbound API). The control plane is a centralized controller that coordinates the forwarding of packets. It is realized by portable software which can be installed on commodity servers (Hu, 2014). In practice, SDN is supported by the NFV that realizes the functions of various network nodes. NFV is specialized software deployed on a host machine to offer the functionalities of traditional network infrastructure. The host machine can be a normal or virtualized server. It significantly reduces capital and operating expenditures (Veeraraghavan et al., 2017). Traditional network deployment is expensive due to expensive proprietary equipment and a lack of flexible scale-up options. For any overloaded node, the operators have to add proprietary equipment even if the option is not commercially viable.

The core network equipment encapsulates the lower layer of transport protocols for communication between nodes and centralized gateways such as the Packet Data Network Gateways (PGWs) of 4G evolved packet core (EPC) to deliver user data traffic (Ying, 2018). The encapsulation process is also known as tunneling. The NFV potentially reduces the cost by virtualizing EPC nodes over a Cloud platform. The functions of core network nodes such as gateway, firewall, and layer three switches can be installed and run as normal instances in a standard server. This technology promotes the innovation of various services and applications in IP networks such as electronic commerce, multimedia streaming, and social media networks.

Several studies indicate that IP networks experience attacks that aim to disrupt services (Gray & D. Nadeu, 2019; Zhang, Zhang, & Yu, 2018). According to the CISCO report of 2020, 23% of DDoS attacks recorded consumed more than 1Gbps bandwidth which is sufficient to take down most online-based organizations (Cisco 2020). Similarly, FEs result in a denial of service to the frequently used online services by legitimate users (Behal, Kumar, & Sachdeva, 2018). FEs have the same effects as DDoS on the services but require different mitigation strategies (Sahoo, Tiwary, & Sahoo, 2018). For that reason, it is important to classify the attacks before selecting appropriate mitigation strategies as discussed by Gupta & Dahiya, (2021).

This paper, therefore, proposes a deep learning framework to detect and isolate FEs from DDoS. The paper is organized into six sections that present the Introduction, Literature Review, Methods and Materials, Model Development, Results and Discussions, and Conclusion.

LITERATURE REVIEW

Overview of Denial-of-Service Attacks

The Denial of Service (DoS) is an attack that mostly affects networked services such as web applications. The attacks attempt to stave off real users from accessing servers and/or services of interest due to the overloading of computing resources such as CPU, bandwidth, buffers, and memory (Rajeev & Mangey, 2021). DoS attacks prevent authorized users to access resources or delay time-critical operations (ITU-T Rec X.800, 1991). DoS attacks usually involve a few attackers on the targeted node or computer.

The DoS attacks that are coordinated to target one node from multiple sources are called Distributed Denial of Service (DDoS). It is further explained in terms of the mode and target of attack. The mode of attack can either be a high-rate flooding or a semantic (non-flooding) attack. Normally, attackers that use high-rate flooding aim to consume critical computational physical resources of the target to overload the system. It may be a manual/human, semi-automatic, or automatic coordinated attack on the target. Semantic attacks aim at exhausting logical resources such as operating systems, communication protocols, or applications hosted by the node (Behal, Kumar, & Sachdeva, 2017).

The target of attacks can be described as attacks targeting networks, attacks targeting cellular telecommunication networks, attacks targeting operating systems, and attacks targeting applications (Raghavan & Dawson, 2011). With network attacks, the services such as web and cloud computing that rely on the networks for their operations are normally affected. The networks are normally attacked when malicious programs alter the Transmission Control Protocol (TCP) sequences such as reset messages, acknowledgment (ACK) segments, or re-transmission-time-out (RTO) features of TCP flows. The modified parameters create abnormal behavior that disrupts the network layer functions and hence service inaccessibility. Normally, networks experience DDoS attacks after the authentication system is compromised. Some of the effects of the attacks on networks might be false location update requests, camping on the false base station (BS), de-registering user false requests, false initiation of push-service, and internet router identity (Gupta & Dahiya, 2021). The overall impact of an attack is to confuse systems that result in service inaccessibility.

Apart from that, attackers may even go further and affect the algorithms and data structures of the operating system. The effect may be visible in the applications that run on top of the operating system. Furthermore, oversized Internet Control Message Protocol (ICMP) fragments cause buffer overflow which makes a system stall or reboot and interferes with service delivery (Guy, 2015). Also, the effect of an attack on the application layer may be vivid in application protocols such as Web/HTTP, and FTP. Most of the time the firewall does not block the ports operating these protocols as a result attackers exploit the vulnerability by sending millions of requests that overload the server (Sachdeva & Kumar, 2014).

Generally, attackers exploit vulnerabilities of systems by installing malicious software on massive numbers of machines (zombie computers) controlled remotely. DDoS attacks are easily launched by an attacker by commanding the zombie computers to access the same target at the same time to overwhelm system resources (Maciel et al., 2018). This may happen in manual fashions, semi-automatic, or automatic attacks. The manual attack happens after an attacker identifies the vulnerability of the target computer connected to the Internet and installs malicious codes for executing commands to initiate the attack. This type of DDoS requires the intervention of an attacker step by step. In the case of semi-automatic DDoS attacks, an attacker cluster the zombie computers into master and slave mode. The master computers are the vulnerable computers where the attacker will install malicious software to control the army of slave computers for receiving commands to inflict target nodes (Praseed & Santhi Thilagam, 2019). In automatic attacks, the process is fully controlled by the malicious program, one command is launched and all steps are executed without the regular intervention of an attacker (Agrawal & Tapaswi, 2019).

Overview of flash events

Flash Events (FEs) are surges of traffic in a communication system/network that overwhelms provisioned resources when simultaneous legitimate users access the services. FEs happen during crowded events such as the Olympic Games, national online elections, and world cup sports where many people are interested in getting updates on results. In such scenarios, the target websites may be overwhelmed by the number of requests generated by legitimate users. For instance, in 1998 on the 90th day of the FIFA World Cup, a crowd of people tried to access the server simultaneously and created FEs as reported by Saravanan, Shanmuganathan, & Palanichamy (2016).

Moreover, a study done by Bhatia (2016) explains that FEs may be categorized into predictable and unpredictable attacks. The predicted classification depends on the availability of prior information on the trend utilization of network resources over some time. The trends can be used to predict the attacks and therefore the system can be optimized to reduce the effects of the attack. Unlike the previous one, unpredicted FE attacks are a sudden surge of requests to the target due to unforeseen demand. According to Jung, Krishnamurthy, & Rabinovich (2002), FE is characterized by observing three (3) key issues which are traffic patterns, client and file reference characteristics. The traffic pattern is a good indicator of server performance on the overall traffic volume received. Normally, a server is set to allow a certain optimal performance depending on the available resource and provide a certain tolerance over which service degradation occurs after exhausting all resources. Similarly, unusual traffic flow is a good indication of the FEs occurrence and prompts an appropriate strategy to reduce its effect.

Likewise, client characteristics reveal whether the traffic originates from an attacker who intentionally attempts to overwhelm the server or from legitimate clients who simultaneously access the server. The occurrence of FEs is indicated by the clients who are normally distributed over the network topology. The intended attack would appear to come from certain IP addresses that are systematically exploiting the server. Lastly, the behavior of reference characteristics can suggest whether the requests come from legitimate clients or not and can easily determine the flash crowd.

Intelligent and dynamic features of the networks are crucial in dealing with flash events. The features are necessary for characterizing FEs because of limited parametric differences between DDoS and FE attacks which are difficult in isolation. (Behal et al., 2017).

Related work

As the number of attacks increases year after year, the security of networked systems raises concerns for many service providers and network operators. Furthermore, by basing on the nature of the attack, it is evident that strategies to combat DDoS attacks differ from those for FEs. Li et al., (2018) suggest the application of deep learning (DL) to potentially improve the detection accuracy of DDoS attacks at the range of 98-99%. Similar efforts were reported by McDermott, Majdani, and Petrovski (2018) who demonstrated the strength of DL by deploying the stated accuracy range after employing deep learning based on bi-directional long short-term memory based on recurrent neural networks (BLSTM-RNN).

A study by Daneshgadeh et al (2019) proposed a model that uses Shannon entropy and kernel online anomaly detection (KOAD) algorithms to detect anomalies in network traffic. Further, the model adopted the mahalanobis distance metric working with machine learning to distinguish the occurrence of DDoS from FE. The proposed study managed to reduce false alarms and improves the detection rate on high-rate and low-rate DDoS. The study further integrated a machine learning method to distinguish FE from DDoS attacks. Nevertheless, the authors did not consider a scenario where DDoS attacks and FEs occur simultaneously.

Sun et al., (2019) proposed a method that used KNN to detect DDoS and FEs based on the flow characteristics of the network traffic. The authors focused on protocol type and entropy of source/destination. The proposed model reduced false alarms and improves the detection rate. However, this study considers only a few features of flow characteristics. The Shannon entropy and Kullberg-Leibler divergence metrics to distinguish HR-DDoS and FE in SDN network traffic were proposed by Sahoo et al., (2018). The proposed metrics reduce false alarms. However, the study focuses only on information metrics to detect HR-DDoS and FE. This technique is limited to accuracy depending on the level of information metric collected. Imamverdiyev and Abdullayeva, (2018) propose an application for deep learning based on a Gaussian-Bernoulli type restricted Boltzmann Machine (RBM) to detect DDoS attacks. However, the study focuses only on the identification of DDoS attacks, and FEs were not considered.

In a fog environment, Priyadarshini and Barik (2019) obtained a detection accuracy of 98.88% using deep learning that was based on long short-term memory networks. The model uses 128 input nodes, 3 hidden layers, and one dense layer to achieve the mentioned accuracy. Similarly, a study by Chen et al. (2019) used hybrid techniques that combined unsupervised and supervised machine learning to isolate DDoS attacks from FEs in network traffic. The authors reported high accuracy in the detection of the attacks. Furthermore, Garg et al., (2019) proposed a model using deep learning techniques to detect anomaly traffic flow of social media in SDN. The model achieved 99% detection accuracy of DDoS attacks. In an attempt to simultaneously detect DDoS and FEs in traffic, Tinubu et al, (2022) developed a model to detect DDoS attacks and manage the FEs. It was based on a multi-layer perceptron classifier.

The model was successful in averting web-based service interruption though could not detect both differentiate DDoS attacks from FEs. From the literature, different scholars attempted to propose a solution to detect DDoS attacks or FEs in IP networks. Nevertheless, there are very limited studies that focus on the simultaneous detection of DDoS attacks and FEs in the IP networks which is a common scenario in a real environment, as hackers always hide behind the FEs.

METHODS AND MATERIALS

Research setting and simulation setup

The study was conducted at the Computer Laboratory of the University of Dodoma, College of Informatics and Virtual Education (CIVE). The environment was created by using an HP-Proliant server with 40 cores CPU and Ubuntu 16.04 operating system, Cisco DHCP, router, and switch because training deep learning and machine learning algorithms requires a computer with sufficient processing capacity. A simulation strategy was employed to deploy a communication network. The setup allowed user computers to access applications and services from the target server while generating DDoS attacks and FE. The DDoS attacks were generated by the Scapy tool library in Python language while FIFA world cup 98 datasets were adopted to simulate the FEs. The traffic of combined effects was then recorded and visualized using Wireshark software. The experiment was done in two steps to generate data for model training and validation. Table 1 summarizes the tools used in the experiment.

Table 1: Tool used in the experiment

S/N	Item	Purpose
1	Python language	Developing model
2	Wireshark	Observing and analyzing network traffic
3	The scapy software	Generating DDoS attacks

S/N	Item	Purpose
4	Fifa 98 dataset for FEs	Injecting flash events in the network
5	Computer	Providing a running environment for the software
6	Target server	Emulating the attacks

Experiment Procedure

The study was conducted by following the following steps; - First, the combined DDoS and FEs traffic data were generated and analyzed using a Wireshark. The generated data were collected and categorized into three groups named training, test, and validation data. Because of the labeled dataset that was opted for in this study, the supervised learning approach was selected to detect DDoS attacks and FE. Thereafter, the classification and regression algorithms were applied to the datasets.

Emulation of flash event

FE was generated based on FIFA world cup 98 using the python scapy tool. The aim was to generate a similar pattern that happened on the FIFA world cup 98 datasets. According to Daneshgadeh et al., (2019) the highest FEs occurred on the 66th day around 23:30 and 23:46 taking the last 16 minutes of the game match between Argentina and England. The study replicated this scenario using a simulator by replacing IP addresses with code IDs to retain privacy.

The Python language was used to prepare the data where the total number of requests for the dataset was 2,712,425. A class C network that was used had IP addresses between 192.168.1.0 and 192.168.73.0, further the script traffic generator in the range of 0 to 3100 was used in the program. In this study, the FE traffic generated has a tolerance between +/- 5% as shown in Figure 1.

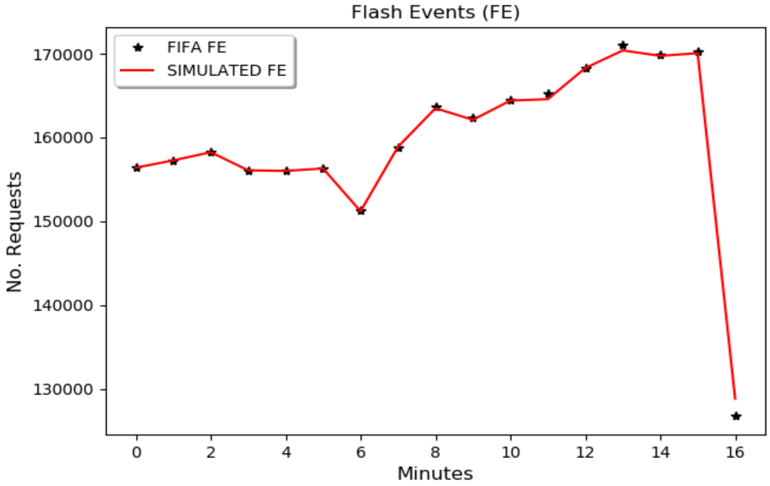

Figure 1: FE Generator.

Emulation of DDoS Attacks

The study focussed on the TCP SYN flood (SYN flood) volume-based network layer to simulate DDoS attacks. The TCP SYN is when an attacker exploits the normal TCP three-way handshake. The attacker interferes with the acknowledgment reply from the end node, hence leaving the server waiting for a while. Consequently, the server will deny other clients as many connections are open and waiting for an acknowledgment. This scenario exhausts network bandwidth, CPU, and other server resources. The execution process of the script took 16 minutes for both scenarios. The simulated DDoS attacks are described in Figure 2.

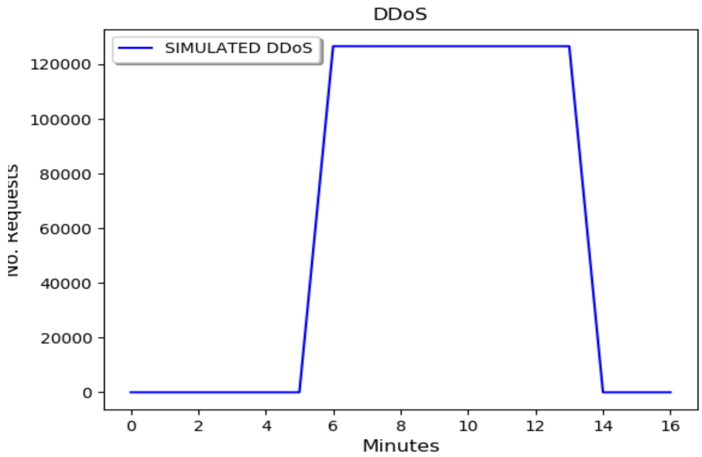

Figure 2: DDoS Network traffic generator.

MODEL DEVELOPMENT

The deep learning supervised models were trained using datasets generated from the emulated network environment. Data manipulation and normalization for the model input part were done by labeling the datasets and extracting features to decide on the number of hidden layers as well as the activation function. The softmax was used as an output activation function to detect the occurrence of DDoS attacks or FEs. It works on the principle of probability by analyzing the possibility of the occurrence of all events and picking the highest one as the accurate output. The designed model was trained and optimized for several epochs. In this process, the model was trained until good accuracy was achieved while avoiding a situation where the model started to generalize or remember the data. This was achieved by maintaining the validation error and normal error below the divergence point during the training phase. The validation was done by using a separate dataset that is blind to the model. The performance metrics used to validate the model are accuracy score, error, and false alarm. The model was deployed in a real network for further evaluation. The model learned various features and patterns of network traffic to accurately detect and distinguish DDoS attacks from FE. Table 2, describes the parameters selected in training and testing the model, which process is explained by a flowchart indicated in Figure 3.

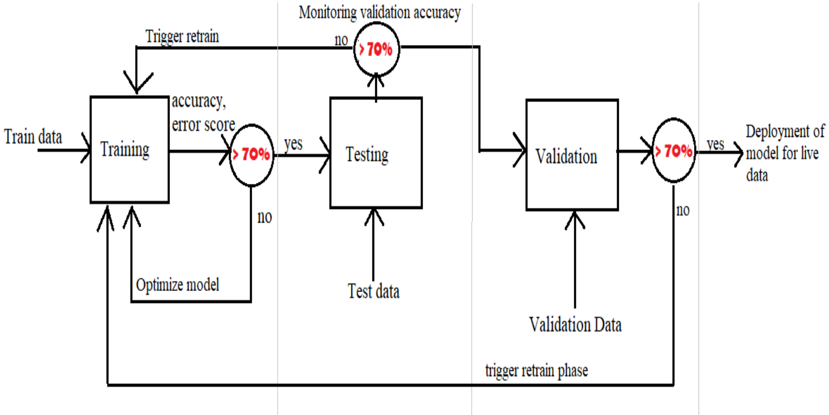

Figure 3: Training, Testing, and Validation phase flow chart

Table 2: Model parameter selection

SN	Parameter	Number/type	Reason
1.	Input	FE/DDoS generated dataset	Number of features
2.	Hidden Layer	1-6	Processing time, reducing over and underfitting
3.	Activation function	Sigmoid, relu, tanh, softmax	Concept of combining activations functions
4.	Output	3	The outlet for logic output either DDoS or FE

Simultaneous Generation of FE and DDoS Attacks

Simulation of DDoS and FEs scenarios was carried out simultaneously, a process that took 16 minutes. The graph in Figure 4 shows the number of generated requests in FE and DDoS attacks. The second dataset was generated by combining both DDoS and FEs together. It provides an alternative dataset to researchers investigating scenarios where FE and DDoS attacks happen at the same time as depicted in Figure 5.

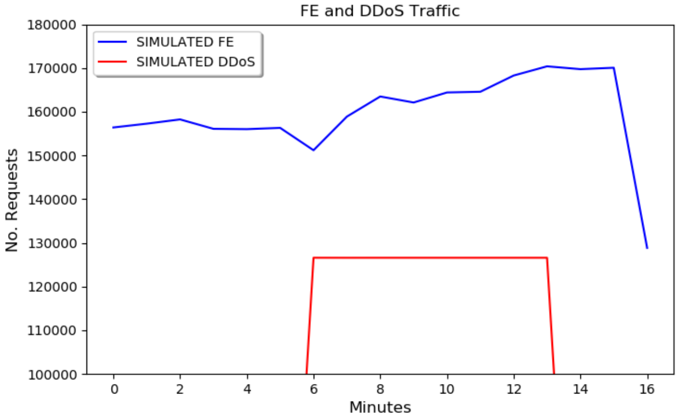

Figure 4: Separated FE and DDoS attacks.

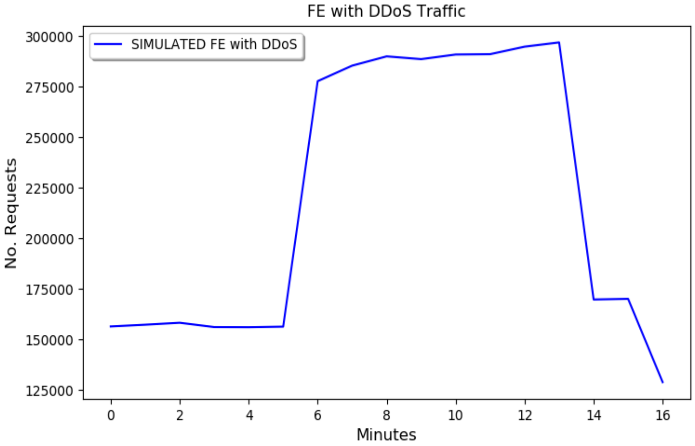

Figure 5: Combined FE and DDoS attacks

RESULTS AND DISCUSSIONS

Model Testing and Validation

The data for testing the model were obtained from the log file of the server hosting the University website. The selected log file was picked by considering the size because the larger the size the larger the number of requests and traffic. The model detected FE and DDoS attacks based on different deployments of the deep layers. This scenario suggests that deep layers have an impact on the output of the model, and consequently, the classification of the attack. The learning rates tested are 0.1 and 0.01. Because of the limitation of space, this paper presents the graphs of the model trained with a learning rate of 0.01. At the three hidden layers, the model accuracy was above 99% and the model stabilized at 25 seconds. The performance remained almost the same with two hidden layers, however, the model attained the saturation stage at a duration above 25 seconds. With one-hidden layer, the model attained saturation after 50 seconds while maintaining detection accuracy. Overall, the results suggest that the proposed model can simultaneously detect DDoS attacks and FEs in traffic at an accuracy of 99 % with a false alarm below 1%. Figures 6-8 summarise the performance of the model at different hidden layers.

Comparison with other Approaches

The isolation process of DDoS from FEs in IP traffic is a critical step in the restoration of services in an attacked network. Several attempts have been made to detect the attacks. However, the approaches focused either on the detection of DDoS attacks or FE in the flowing traffic. Studies by Liu et al., (2018), Imamverdiyev et al (2018), and Chen et al., (2019) proposed deep learning techniques to detect DDoS attacks. The contributions were significant nevertheless the techniques could not identify FEs. The other studies as reported by Daneshgadeh et al., (2019), Sun et al., (2019), and Tinubu et al (2022) proposed detection models that used the KOAD algorithm, KNN, and A Multi-Layer Perceptron (MLP) classifier respectively. Of the reviewed techniques, the authors attempted to develop models that could isolate the occurrence of either DDoS and/or FE whenever subjected to a specific attack. The models were trained with either DDoS or FE patterns and tested against the event under trial. The real network traffic experiences both DDoS and FEs at a time. In this work, a deep learning technique was used to create a model that can detect both DDoS attacks and FEs at the same time. A summary of the comparison of DDoS attacks and FEs isolation methods is shown in Table 3.

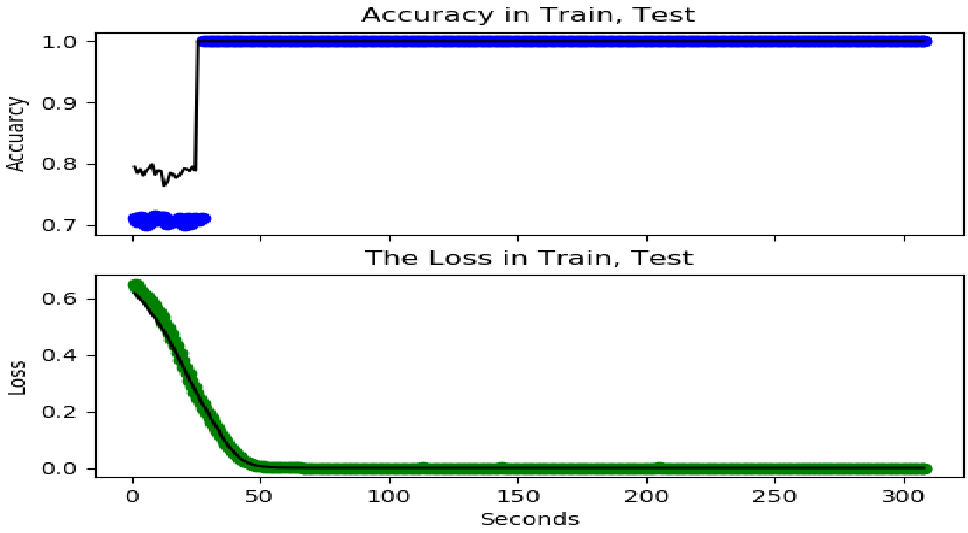

Figure 6: 3-Hidden layer model performance at learning rate 0.01.

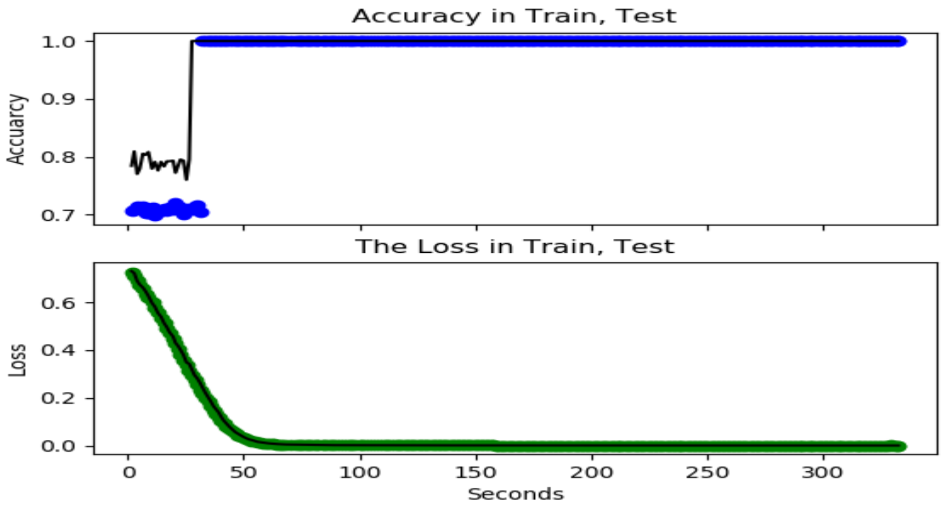

Figure 7: 2-Hidden layers model performance at learning rate 0.01.

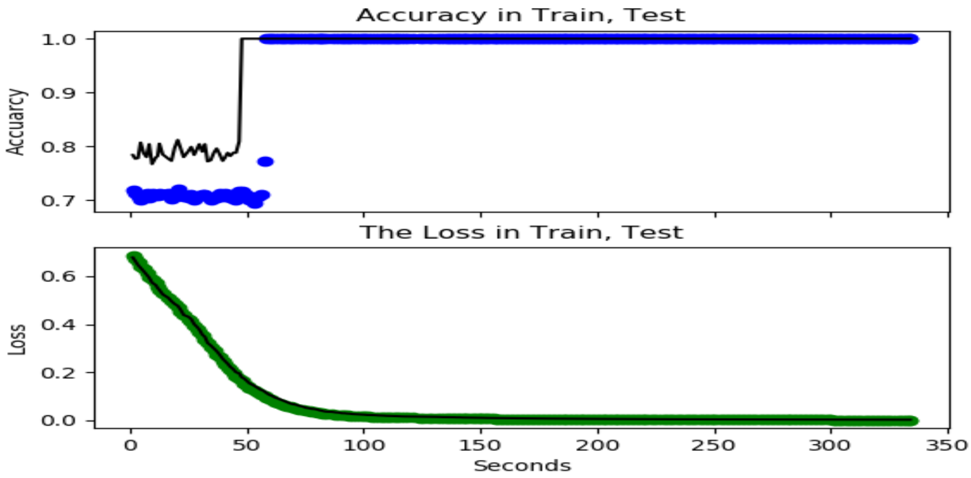

Figure 8: 1-Hidden layer model performance at learning rate 0.01.

Table 3: Comparison of DDoS attack and FE isolation methods

S/N	Reviewed works	Model technique	The type of attack detected
1	Li et al., (2018)	DDoS detection model using deep learning in SDN	DDoS
2	Imamverdiyev et al., (2018)	Deep learning Method for DDoS Attack Detection	DDoS
3	Daneshgadeh et al (2019)	Kernel Online Anomaly Detection (KOAD) algorithms	DDoS or FE
4	Chen et al., (2019)	Hybrid Unsupervised and Supervised machine learning	DDoS only
5	Sun et al., (2019)	KNN to detect DDoS	DDoS or FE
6	Tinubu et al (2022)	A Multi-Layer Perceptron (MLP) classifier	DDoS or FE
7	Proposed method	Deep Learning	DDoS and FEs

CONCLUSION

This study proposes a deep learning model to detect and isolate DDoS attacks from FEs occurring simultaneously in network traffic. The model attained a high accuracy at 99% and a false alarm as low as 1% at the learning rate of 0.01. The deep learning model was investigated at three different layers and the one with 3–hidden layers gave the best results in terms of learning rate and short time for a model to attain its stable state (<25 seconds). This may come with a computational cost because as the number of data to train the model increases, more computing resources will be required, which may again affect the model convergence rate. Therefore, a proper balance must be observed, between the number of hidden layers, computing resources, and volume of data required to train the model.

REFERENCES

Agrawal, N., and Tapaswi, S. (2019). Defense Mechanisms against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges. *IEEE Communications Surveys and Tutorials*, **21**(4): 3769–3795. https://doi.org/10.1109/COMST.2019.2934468

Allmer, J. (2014). *miRNomics: MicroRNA Biology and Computational Analysis*. *1107*, 333. https://doi.org/10.1007/978-1-62703-748-8

Behal, S., Kumar, K., & Sachdeva, M. (2017). Discriminating flash events from DDoS attacks: A comprehensive review. *International Journal of Network Security*, **19**(5): 734–741. https://doi.org/10.6633/IJNS.201709.19(5).11

Behal, S., Kumar, K., and Sachdeva, M. (2018). D-FACE: An anomaly-based distributed approach for early detection of DDoS attacks and flash events. *Journal of Network and Computer Applications*, **111**: 49–63. https://doi.org/10.1016/j.jnca.2018.03.024

Bhatia, S. (2016). Ensemble-based model for DDoS attack detection and flash event separation. *FTC 2016 - Proceedings of Future Technologies Conference*, (December 2016), 958–967. https://doi.org/10.1109/FTC.2016.7821720

Chen, X., Li, B., Proietti, R., Zhu, Z., and Yoo, S. J. B. (2019). Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks. *Journal of Lightwave Technology*, **37**(7): 1742–1749. https://doi.org/10.1109/JLT.2019.2902487

Cisco 2020. (2020). CISO Benchmark Report: Cisco Annual Internet Report (2018-2023). In *Computer Fraud & Security*. https://doi.org/10.1016/S1361-3723(20)30026-9

Daneshgadeh, S., Ahmed, T., Kemmerich, T., and Baykal, N. (2019). Detection of DDoS Attacks and Flash Events Using Shannon Entropy, KOAD, and Mahalanobis Distance. *Proceedings of the 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2019*, 222–229. https://doi.org/10.1109/ICIN.2019.8685891

Daneshgadeh, S., Kemmerich, T., Ahmed, T., and Baykal, N. (2019). An Empirical Investigation of DDoS and Flash Event Detection Using Shannon Entropy, KOAD, and SVM Combined. *2019 International Conference on Computing, Networking, and Communications, ICNC 2019*, 658–662. https://doi.org/10.1109/ICCNC.2019.8685632

Garg, S., Kaur, K., Kumar, N., and Rodrigues, J. J. P. C. (2019). Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective. *IEEE Transactions on Multimedia*, **21**(3): 566–578. https://doi.org/10.1109/TMM.2019.2893549

Gray, K., and D. Nadeu, T. (2019). Network Function Virtualization. In *Signals and Communication Technology*. Elsevier. https://doi.org/10.1007/978-3-030-01647-0_5

Gupta, B., and Dahiya, A. (2021). *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges, and Countermeasures*. Taylor and Francis Group, LLC.

Guy, P. (2015). *Software Networks: Virtualization, SDN, 5G, and Security*. IEEE Wiley Press.

Hu, F. E. I. (2014). Network Innovation through OpenFlow and SDN. In *Network Innovation through OpenFlow and SDN*. https://doi.org/10.1201/b16521

Imamverdiyev, Y., and Abdullayeva, F. (2018). Deep Learning Method for Denial of Service Attack Detection Based on Restricted Boltzmann Machine. *Big Data*, *6*(2), 159–169. https://doi.org/10.1089/big.2018.0023

ITU-T Rec X.800. (1991). *Data Communication Networks: Open System Interconnect (OSI); Security, Structure, and Applications*, p. 48.

Jung, J., Krishnamurthy, B., and Rabinovich, M. (2002). Flash Crowds and Distributed Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, 293–304. https://doi.org/10.1201/9781420070217.ch22

Kumar, A., and Panda, S. P. (2019). A Survey: How Python Pitches in IT-World. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives, and Prospects, COMITCon 2019*, 248–251. https://doi.org/10.1109/COMITCon.2019.8862251

Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., and Gong, L. (2018). Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *International Journal of Communication Systems*, **31**(5): 1–15. https://doi.org/10.1002/dac.3497

Maciel, R., Araujo, J., Dantas, J., Melo, C., Guedes, E., and Maciel, P. (2018). Impact of a DDoS Attack on Computer Systems : An Approach Based on an Attack Tree Model. *Annual IEEE International Systems Conference (SysCon)*. https://doi.org/10.1109/SYSCON.2018.8369611

McDermott, C. D., Majdani, F., and Petrovski, A. V. (2018). Botnet Detection in the Internet of Things using Deep Learning Approaches. *Proceedings of the International Joint Conference on Neural Networks, 2018-July*, 1–8. https://doi.org/10.1109/IJCNN.2018.8489489

Nadeau, T. D., and Gray, K. (2013). *T.D. Nadeau and Ken Gray, "Centralized and distributed Control and Data Plane" in Software Defined Networks, Sebastopol: O'REILLY, 2013*. Retrieved from http://oreilly.com/catalog/errata.csp?isbn=9781449342302

Praseed, A., and Santhi Thilagam, P. (2019). DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Communications Surveys and Tutorials*, **21**(1): 661–685. https://doi.org/10.1109/COMST.2018.2870658

Priyadarshini, R., and Barik, R. K. (2019). A deep learning-based intelligent framework to mitigate DDoS attacks in a fog environment. *Journal of King Saud University - Computer and*

Information Sciences. https://doi.org/10.1016/j.jksuci.2019.04.010

Raghavan, S.., and Dawson, E. (2011). *An Investigation into the Detection and Mitigation of Denial of Service (DoS) Attacks: Critical Information Infrastructure Protection* (Vol. 59). Springer.

Rajeev, S., and Mangey, R. (2021). *Distributed Denial of Service Attacks Concepts, Mathematical and Cryptographic Solutions* (Vol. 6). CPI books GmbH.

Sachdeva, M., and Kumar, K. (2014). A Traffic Cluster Entropy-Based Approach to Distinguish DDoS Attacks from Flash Event Using DETER Testbed. *ISRN Communications and Networking, Volume 201*(Article ID 259831), 15 pages.

Sahoo, K. S., Tiwary, M., and Sahoo, B. (2018). Detection of high-rate DDoS attacks from flash events using information metrics in software-defined networks. *2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018, 2018-Janua*, 421–424. https://doi.org/10.1109/COMSNETS.2018.8328233

Saravanan, R., Shanmuganathan, S., and Palanichamy, Y. (2016). Behavior-based detection of application layer distributed denial of service attacks during ash events. *Turkish Journal of Electrical Engineering and Computer Sciences*, **24**(2): 510–523. https://doi.org/10.3906/elk-1308-188

Sun, G., Jiang, W., Gu, Y., Ren, D., and Li, H. (2019). DDoS Attacks and Flash Event Detection Based on Flow Characteristics in SDN. *Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance*. https://doi.org/10.1109/AVSS.2018.8639103

Tinubu, O. C., Sodiya, A. S., Ojesanmi, O. A., Adeleke, E. O., and Timehin, A. A. (2022). An Intelligent Model for DDoS Attack Detection and Flash Event Management. *International Journal of Distributed Artificial Intelligence (IJDAI)*, **14**(1): 1-15.

Veeraraghavan, M., Sato, T., Buchanan, M., Rahimi, R., Okamoto, S., and Yamanaka, N. (2017). Network function virtualization: A survey. *IEICE Transactions on Communications*, *E100B*(11), 1978–1991. https://doi.org/10.1587/transcom.2016NNI0001

Ying, Z. (2018). *Network Function Virtualization: Concepts and Applicability in 5G*. IEEE Wiley Press (ISBN: 978-1-119-39060-2)

Yu, J., and Zhou, X. (2020a). One-Dimensional Residual Convolutional Autoencoder Based Feature Learning for Gearbox Fault Diagnosis. *IEEE Transactions on Industrial Informatics*, **16**(10): 6347–6358. https://doi.org/10.1109/TII.2020.2966326

Yu, J., and Zhou, X. (2020b). One - Dimension Residual Convolutional Auto - Encoder - Based Feature Learning for Gearbox Fault Diagnosis. *IEEE Transactions on Industrial Informatics*, *PP*(c), 1. https://doi.org/10.1109/TII.2020.2966326

Zhang, B., Zhang, T., and Yu, Z. (2018). DDoS detection and prevention based on artificial intelligence techniques. *2017 3rd IEEE International Conference on Computer and Communications, ICCC 2017, 2018-Janua*, 1276–1280. https://doi.org/10.1109/CompComm.2017.8322748