International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

DeepAD: Propose a new DNN activation function to improve DDoS attack detection in SDN

Shahad A. Hussien[1] and Alharith A. Abdullah [2]

[1,2] College of Information Technology, University of Babylon, Babil, Iraq

Email:alharith@uobabylon.edu.iq
E-mail address: alharith@uobabylon.edu.iq, shahad.alshamary@uobabylon.edu.iq

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Software-defined networks (SDNs) play a fundamental role in the core infrastructure of 5G networks. Therefore, the concern for SDN security has become critical, and DDoS attacks are one of the most significant threats to SDN as the attacker focuses on a single point, which is the controller, and thus leads to the failure of the entire network. In this study, we present Deep Attack Detection (DeepAD), a novel approach to detect DDoS attacks using a deep neural network with a novel activation function. The proposed activation function, SETAF, is based on the features of exponential and sigmoid functions as well as dynamic thresholding and thus adapts to traffic changes and is able to distinguish different DDoS attack patterns. The DeepAD model is implemented and tested on the CICIDS2017 dataset. In addition, the proposed model using SETAF activation function is compared with the standard sigmoid activation function and the loss function ratio is less than 0.01 with an accuracy of 0.99. On the other hand, the proposed DeepAD model was implemented on an SDN environment using Mininet emulator and POX controller, and the experimental results proved the effectiveness of the DeepAD approach in significantly improving the accuracy and speed of DDoS detection.

Keywords: DeepAD, DNN, CICIDS2017, SETAF, DDoS, SDN.

1. INTRODUCTION

Software-defined networks (SDNs) are closely linked to 5G networks to provide various services and manage them more easily; Due to the presence of a central layer responsible for managing and controlling other network devices [1]. The SDN network architecture consists of three layers: the data plane, the control plane, and the application plane as shown in Figure 1.

The data plane includes network devices such as switches and routers that support the OpenFlow protocol, while the control plane includes one or more control units and is connected to the data plane via the southbound interface and is responsible for managing the data plane through the OpenFlow protocol, where it works to build the routing table in OpenFlow Switch [2]. As for the application layer, it consists of a group of services that operate on the network, such as firewall application, load balancing, quality of service, etc., and the application layer is linked to the control layer through the Northbound Interface.

However, SDN security is the core of 5G infrastructure security [1]. There are many security challenges facing SDN architecture, but DDoS attack is the most common attack on SDN networks, and it greatly affects service delivery, causing financial damage as well as targeting the reputation of the network in providing services [2]. DDoS attack focuses the attack on the control layer, thus disabling the controller from network management and harming ordinary users and the services provided. Although many intrusion detection techniques are available, they fail to accurately detect DDoS attacks due to the dynamic nature of network traffic [3].

Machine learning methods are widely used as an essential part of network security, especially in detecting attacks, due to their ability to identify complex patterns and adapt to multiple scenarios [4], [5].

E-mail:author's email

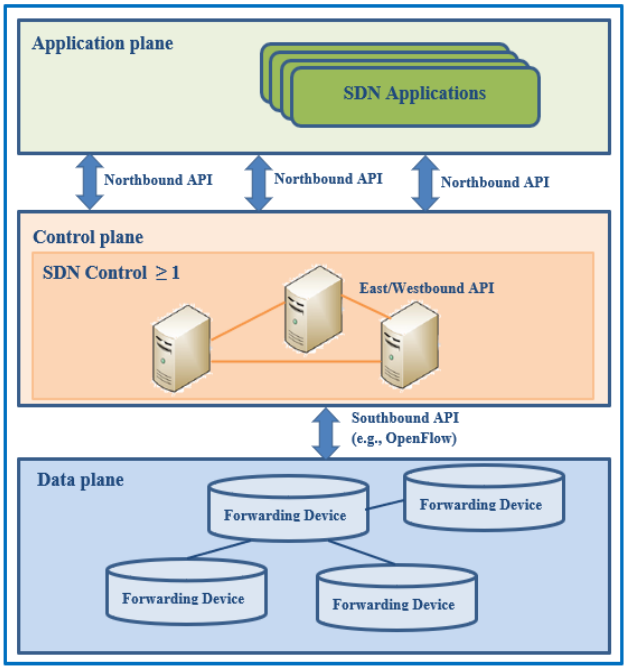

Figure 1. SDN architecture.

Artificial neural networks represent one of the most important modern trends and are based on deep learning, in which a group of interconnected nodes is used as a network, which is formed in the form of layers similar to the human brain [6]. Deep neural networks (DNNs) use a set of tools and functions, including the activation function used within the layers, including the output layer, which fundamentally affects the performance of DNNs [7,8].

In this study, we propose deep attack detection (DeepAD), which is an approach that uses a new activation function in the output layer of the DNN to detect DDoS attacks in SDNs more accurately and dynamically.

The sigmoid exponential threshold activation function (SETAF) is the function proposed in DeepAD. It is specifically proposed to improve the performance and efficiency of DDoS attack detection and is based on the characteristics of DDoS attacks, such as window traffic volume and traffic height; moreover, it is proportional to the dynamic change of network traffic. Therefore, through the features of activation function, DeepAD can more accurately and effectively distinguish between normal traffic and DDoS attacks.

The remainder of this paper is organized as follows: In Section 2, the main literature on emerging technologies that use deep learning to identify DDoS attacks and advanced network security is reviewed. In Section 3, the proposed approach and the proposed activation function SETAF are explained. In Section 4, detailed information about the dataset used during training and the mechanism for implementing the proposal on an SDN environment are provided. In Section 5, the DeepAD evaluation criteria are described, and the SETAF activation function is compared with standard functions. In Section 6, concluding remarks along with recommendations for further research are presented.

2. Related Works

In the past few years, many mechanisms have been used to manage SDN DDoS attacks, which is characteristic of the new period. Polat et al. (2020) [9] created an SDN DDoS detection system, which analyses and filters traffic and selects features in a control plane. Different machine learning methods, such as ANN, KNN, SVM and naive Bayes, were used for detection of DDoS attacks. However, how to overcome the delicate borderline between accuracy and the technical constraint of any computational system remains unclear.

Ujjan and his group of researchers (2020) [10] furthered the previous techniques by opting to use a DNN, which they attached to Snort 2 IDS. The experimental setup that uses adaptive polling alongside sFlow data showed signs of potential in detecting DDoS attacks along the way. The robotisation of workplaces would entail a further increase in the sophistication and accuracy of the robots' strategy.

An SDN solution was proposed in 2022 by Anyanwu et al. [11] mainly for the sake of the prevention of DoS in VANETs. The initial manifestation of an intrusion detection model came with the application of AI algorithms, such as the fast learnable algorithm of GSCV or the radial basis function of the SVM classifier. However, detecting DDoS formally is seen to be with certain level of success still lacking in generalisations. An experimental study demonstrated that precision and efficiency should be improved in the future.

According Fouladi et al. (2022) [12], SDN networks definitely continue to be susceptible to traditional DDoS attacks. In addition, they granted a concrete design that helps indent and eradicate the risk. The statistical information from network traffic was extracted using discrete wavelet transform and then analysed with an autoencoder neural network to detect DDoS attacks. The activation of the DDoS detection mechanism led to a decrease in processing costs due to the average hit rate in the switch flow table.

Singh and Jayakumar introduced a thorough approach to identifying and stopping DDoS attacks within the SDN architecture in 2022 [13]. This system comprised two processes: identifying DDoS attacks and deploying countermeasures. Initially, characteristics were identified, then the IU-ROA was utilised for efficient feature selection; ultimately, the deep CNN model was applied for classification. During the second phase, a bait detection method was utilised to disable the intruder node.

In 2022 [5], Kareem and his team utilised machine learning to create a method for identifying DDoS attacks in SDN network data. They applied a feature selection

algorithm and multiple experiments to isolate a fitting feature set capable of categorisation in SDN settings. The evaluation of the efficiency and high accuracy rates of the PART classifier using multiple metrics established it as a strong classifier. The results revealed that ML algorithms, feature selection methods and the PART classifier applied to the SDN network are remarkably efficient in detecting DDoS attacks.

The authors of Al-Dunainawi et al. (2023) [14] proposed the introduction of Mininet along with a Ryu controller and a 1D-CNN system dedicated to identifying and combating DDoS assaults in an environment which is based on SDN. The 1D-CNN system identifies deviations which are given as the labelled network traffic data for training and are indicative of DDoS attacks. In this manner, the model could be more accurately tuned in a shorter period due to the NSGA-II's seven hyperparameter fine-tuning. The control system, i.e. RYU, aims at strengthening the network structure by employing suitable prevention strategies and modifying network regulations when it intends to cut down identified risks. Numerous tests conducted in a simulated SDN setting with an actual DDoS attack dataset verified the impressive 99.99% accuracy in detecting the method.

3. PROPOSED ATTACK DETECTION IN SDN: DESIGN AND METHODOLOGY

This section contains implementation details and our proposed method for identifying a DDoS attack. Our method is based on DNN; however, we include a new activation function called SETAF. The DDoS detection approach in an SDN context is summarised in Figure 2, and the next subsections discuss our methodology in depth.

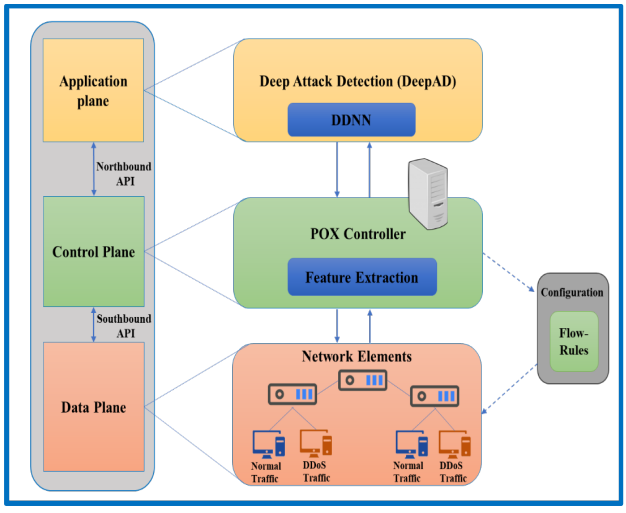

Figure 2. SDN-based proposed DDoS detection.

A. *DeepAD: Methodology*

A specialized method called DeepAD uses a sophisticated DNN model to analyze DDoS attack traffic in SDN systems. The process of building a DeepAD model goes through several stages, starting from preprocessing the CICIDS2017 dataset and building the trained DDNN model to evaluating and validating the performance of the trained model. The following explains the main steps in building the model. The block diagram illustrates these steps in Figure 3.

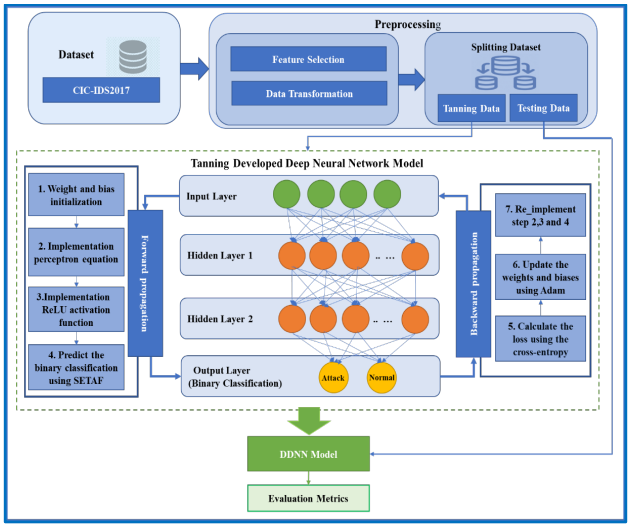

Figure 3. General diagram of DDNN.

We used a data preprocessing step in our work to improve DDoS attack detection efficiency. On the basis of their real-time operability and classification performance, four robust characteristics were chosen for this: Fwd IAT mean, destination port, destination IP and packet length mean [15]. We used data transformation techniques to obtain these characteristics to operate with the neural network model. This involved encoding the intended feature (labels) into numerical values using label encoding and turning IP addresses into scalar values for more straightforward representation. Additionally, we scaled the feature values within the range of [0, 1] by using min–max normalisation. Within the neural network model, this normalisation guaranteed comparability and interpretability.

Last, we split the dataset into testing and training sets, dividing it into 30% and 70%, respectively. This part was crucial to the training, evaluation and acquisition of an objective metric for the generalisation ability of the model.

While DDNNs are constructed from four layers in the model building stage. These layers include the first layer, which is the input layer that represents the traffic parameters, the second layer, which represents the first hidden layer, the third layer, which is the second hidden layer, and the fourth layer, which is the output layer, in which the traffic is classified as natural or attack, depending on the activation function used in it, which is SETAF, as shown in Figure 4, which also illustrates the connection structure of the DDNN architecture.

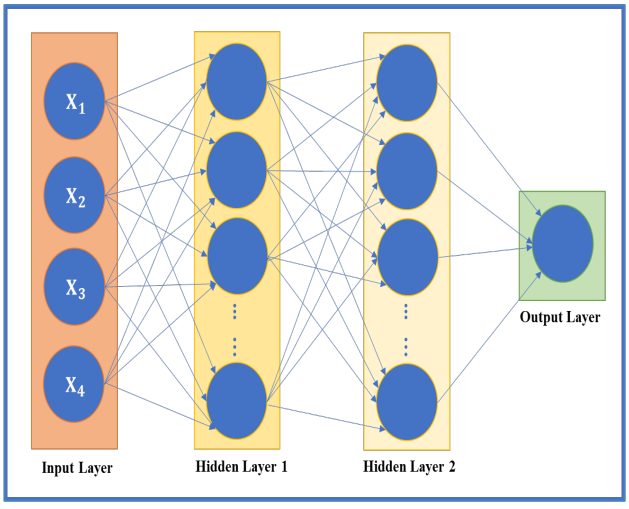

Figure 4. DDNN architecture.

- Starting from the initialization input data, the input layer takes four features to the model from the preprocessed CICIDS2017 dataset.
- The first hidden layer consists of 128 neurons. The operations of the nodes are represented by adding bias, matrix multiplication and using and activating the rectified linear unit (ReLU). This layer can generate its outputs on the basis of the capabilities of the operations that it features.
- Similar to the first hidden layer, the second hidden layer consists of 64 neurons. The outputs of the previous layer were used as its inputs, which were multiplied using matrix multiplication and addition with bias and then passed through the ReLU activation function.
- The output layer, which is the last layer in the architecture, consists of a neuron capable of generating classification results. Thus, three types of operations were implemented: matrix multiplication, bias addition and a new activation function called SETAF, which is explained in detail in the next subsection.

Several techniques were used during training to optimise the model parameters. The model is trained by maximising the cross-entropy loss function [16] to determine the difference between the predicted and true derivatives. The model performance has already been improved by tuning the parameters and minimising the loss function using the Adam optimiser [17]. Furthermore, among the evaluation criteria, accuracy was chosen as a measure of how well the model predicts outcomes.

B. Formulation of SETAF

The SETAF is an improvement of the activation functions commonly used in DNNs. By combining the beneficial features of sigmoid, exponential and threshold activation functions, SETAF obtains increased efficiency in the performance of its networks. Below is the method of calculating and deriving the SETAF equation.

In DNNs, the activation function plays the role of Sig(x); however, it is prominently known as the sigmoid function. As indicated by Eq. (1), it converts input values into a range between 0 and 1, resulting in a continuous and restricted activation response.

$$\mathrm{Sig}(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Firstly, Eq. (2) is used to scale the sigmoid function by an alpha α factor.

$$\mathrm{Sig}(x) = \frac{\propto}{1 + e^{-x}} \tag{2}$$

Secondly, Eq. (3) illustrates the exponential function.

$$\mathrm{Exp}(x) = e^{d} \tag{3}$$

Thirdly, Eq. (4) reveals that the squared distance function, also known as the squared difference function, represents the squared difference or squared difference between x and the threshold value.

$$d = (x - \delta)^2 \tag{4}$$

Eq. (4) may be used to obtain a number between 0 and 1 by inputting Eq. (4) into Eq. (3) after multiplying Eq. (4) by −1 to obtain a negative value for the exponential function, as shown in Eq. (5).

$$\mathrm{Exp}(x) = e^{-(x-\delta)^2} \tag{5}$$

The beta β factor in Eq. (6) also scales the exponential component.

$$\mathrm{Exp}(x) = \beta\, e^{-(x-\delta)^2} \tag{6}$$

Lastly, Eq. (7), also known as the SETAF equation, is produced by fusing Eqs. (2) and (6).

$$\mathrm{SETAT} = \frac{\propto}{1 + e^{-x}} + \beta\, e^{-(x-\delta)^2} \tag{7}$$

One way to simplify the equation is shown in Eq. (8).

$$\mathrm{SETAF} = \propto \mathrm{Sig}(x) + \beta\, e^{-(x-\delta)^2} \tag{8}$$

Enhancing DNN efficiency through a responsive and adaptive reaction is the aim of the SETAF. The use of sigmoid, exponential and threshold components gives a high degree of nonlinearity. Moreover, SETAF is endowed with characteristics that enable it to adjust parameters for particular requirements of the problem so that one can manipulate it differently to suit the purpose.

Exponential growth or decay patterns of data can be easily described through mathematical modelling. Such mathematical models allow the researchers to portray complex data relationships accurately. That is, SETAF can

handle nonlinear relationships because a simple sigmoid function may not be able to give an accurate picture of such relationships reliably.

4. Experimental Results and Analysis

In this section, we present the training and evaluation results of the approach using the widely recognized 2017 CICIDS dataset [18], in addition to implementing this approach in simulating SDN topologies.

A personal computer with 7.9 GB memory and an Intel Core i7 CPU E7500 running at a dual frequency of 3.4 GHz was set up to create a realistic SDN simulator and evaluate the model's performance during the training and testing phases.

A. Utilising the CICIDS2017 Dataset for Model Training and Testing

The program was developed in Python language and PyCharm IDE using the environment. The datasets were split, where 70% of the data is allocated as the training set and 30% of data is kept apart for testing purposes. Normal traffic was designated as 0, and attacks were designated as 1. The training dataset utilized for constructing the model consisted of 158,022 records. The model was trained for 100 epochs, representing the number of complete passes over the training dataset during training. Each epoch consists of multiple iterations, where the model receives batches of data and updates its parameters on the basis of the calculated loss and optimization algorithm.

Figure 5 demonstrates the results of the loss function for the training and validation phases of the model. A low loss value, particularly below 0.09 as seen in this instance, indicates strong performance in accurately predicting the desired output. Meanwhile, Figure 6 displays the accuracy results for the training and validation phases of the model. In this scenario, an accuracy value above 0.99 signifies exceptional performance in accurately predicting the desired output.

The good results for loss value and accuracy are due to the use of the SETAF, which characterizes the DeepAD model with nonlinearity and thus enables it to learn complex patterns.

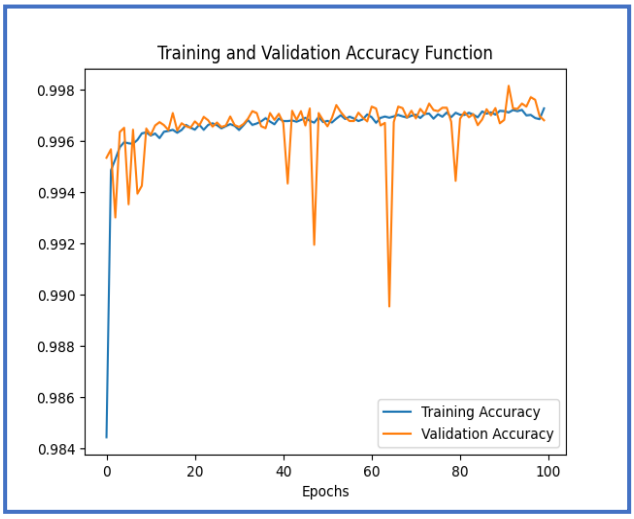

Figure 5. Training and validation accuracy.

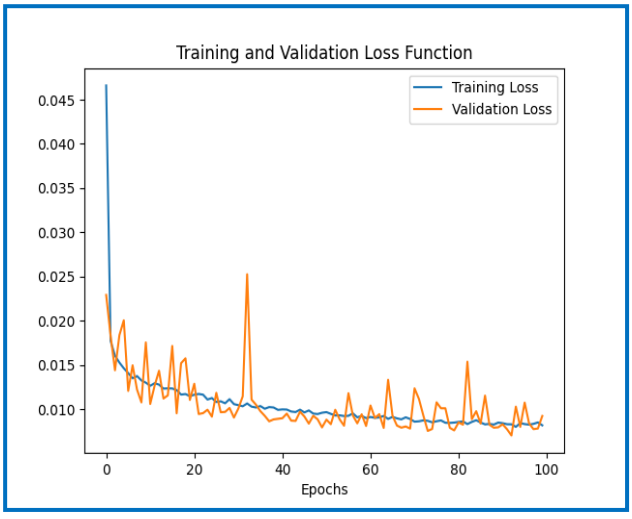

Figure 6. Training and validation loss function.

Figure 7 presents the results of accuracy, recall, precision, F-score and specificity for the test data. Additionally, a mean squared error of 0.0032 was recorded.

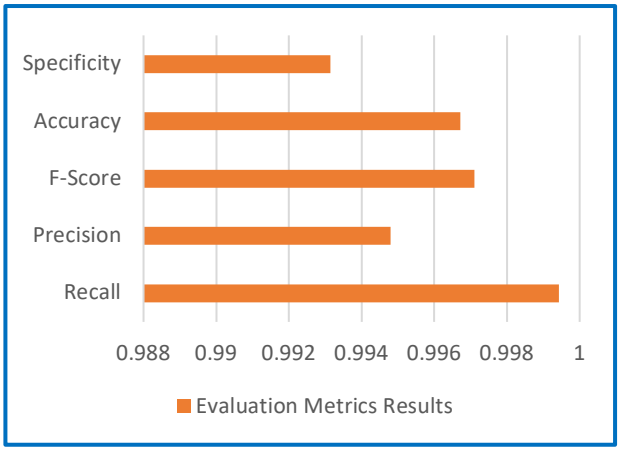

Figure 7. Evaluation results for binary classification of testing data.

B. Implementation and Evaluation Results of DeepAD in SDN Topology

We used an SDN architecture with a Pox controller serving as the control plane to implement DeepAD. The Pox controller is mostly written in Python and is selected for certain research objectives due to its lightweight and adaptable design.

VMware Workstation 15.5.7 was utilised in the implementation of the SDN network topology. Two virtual computers that were expressly set up to run Ubuntu 20.04 LTS were part of the configuration. One of the virtual machines utilised Mininet, enabling the creation of OpenFlow switches and end devices for the data plane [19]. The topology, depicted in Figure 8, adopted a tree structure with a depth of 2 and comprised five switches and 64 hosts.

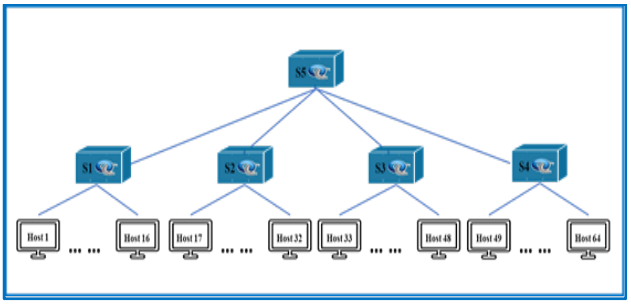

Figure 8. Mininet-constructed SDN network topology.

Open vSwitch was employed to handle forwarding, whilst network traffic packets were generated using Scapy, a versatile tool with capabilities for packet generation, sniffing and manipulation. Using Scapy, we generated two types of traffic: normal traffic and DDoS attack traffic.

The other virtual machine acted as a control plane, hosting the Pox controller, a software that acts as a network operator and communicates with network devices via the OpenFlow protocol. The Pox controller facilitated centralized control and network management, acting as an interface between switches and control applications [20].

DeepAD was implemented as an application within the Pox controller to detect DDoS attacks. This implementation involves generating attack traffic and targeting either a single host or a specific group of hosts within the SDN data plane. Figure 9 illustrates the attack traffic captured by the Wireshark tool, which utilises spoofed IP addresses displayed in the source column to target destination IP addresses, such as 10.0.0.10, 10.0.0.11, 10.0.0.12, 10.0.0.13 and so on.

DeepAD is implemented as an application within the Pox controller to detect DDoS attacks. This implementation includes generating attack traffic targeting a single host or a specific set of hosts within the SDN data plane, generating normal traffic, and monitoring the proposed model and its ability to detect the attack.

Figure 9 shows the generated attack traffic to attack multiple hosts captured by Wireshark, which uses the spoofing IP addresses displayed in the source column to target specific destination IP addresses, such as 10.0.0.10, 10.0.0.11, 10.0.0.12, 10.0.0.13, and so on.

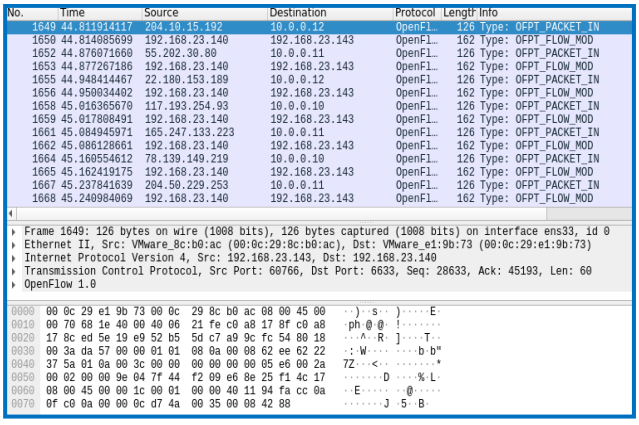

Figure 9. Analysis of attack traffic captured by Wireshark.

Upon receiving the traffic information, the Pox controller server utilises the DeepAD model for traffic classification. The DeepAD model analyses the incoming traffic and determines whether it should be classified as normal or an attack. Figure 10 visually represents the detection of an attack using the DeepAD model.

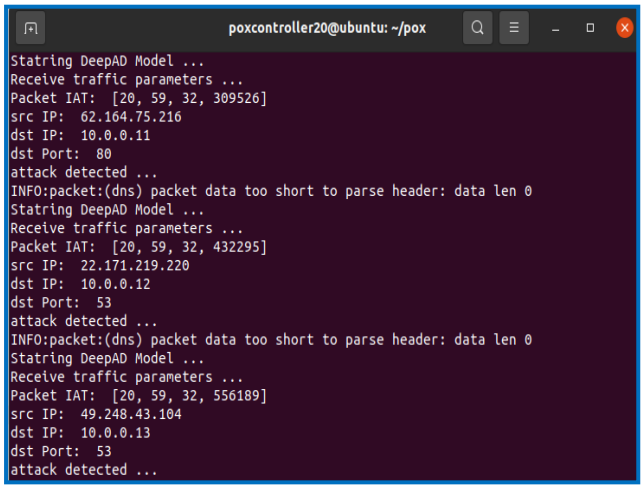

Figure 10. Detection of DDoS traffic.

In the context of SDN emulator, network performance metrics, namely CPU utilization and RAM network utilization, were calculated in the presence and absence of a DDoS attack. Figure 11 shows the CPU and RAM network utilization measurements in an SDN network before and during the attack.

By comparing the results of the CPU and RAM network utilization metrics before and during the attack, we notice the significant performance degradation caused by the attack. The high CPU utilization during the attack indicates increased processing requirements to handle the attack traffic or computational overhead of security mechanisms, which in turn damages the network infrastructure and the performance of normal users. Also, the increased traffic on the network leads to increased RAM network utilization, which leads to higher memory requirements due to the effects of the attack on data processing and storage.

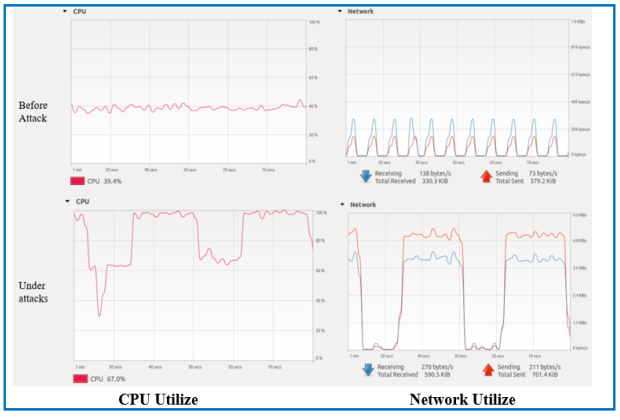

Figure 11. CPU and RAM of SDN before and under attack.

The link latency between Host 2 and Host 3 was also measured in both scenarios before and during the DDoS attack, where link latency refers to the time it takes for a packet to traverse a link between two hosts. Figure 12 illustrates the difference between latency in an SDN environment.

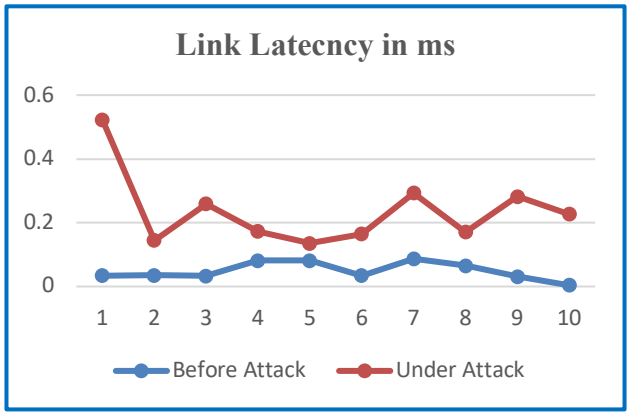

Figure 12. Figure 12: Link latency in an SDN network.

The link latency was measured using the ping command in the Mininet command line interface, as shown in Figure 13. In this figure, we illustrate the steps to accurately measure link latency in the case before and during the attack.

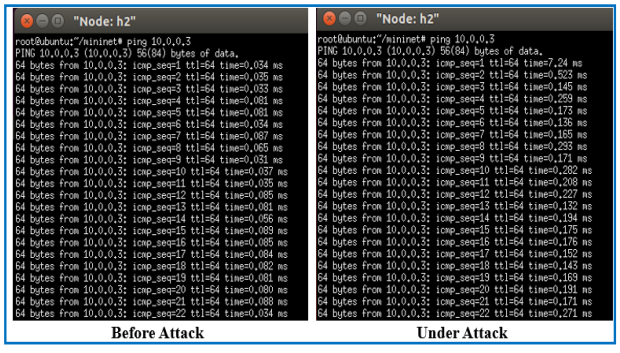

Figure 13. The Link Latency Calculation by PING Command Between Host 1 and Host 2

When comparing the results of link latency measurements before and during the attack, we can conclude that the attack has affected the performance of the link between hosts in addition to the burden on the entire network and the control layer in particular.

The increase in link latency during the attack indicates potential network congestion, longer packet processing times, or network resource constraints caused by the attack activity, which negatively affects normal users and the performance and availability of the network. In addition, these measurements provide valuable metrics about the effects of attacks on network response and help in developing strategies to mitigate latency-related issues, thereby enhancing the overall performance and security of the SDN network.

5. EVALUATION OF DEEPAD METHODOLOGY

A comparative analysis was conducted with a standard DNN model that employed the sigmoid activation function to evaluate the performance of the DeepAD model. By contrast, the DeepAD model utilised the SETAF specifically designed for anomaly detection. Both models were trained using the same dataset and adhered to the defined training and test data splits. Also, both models used the same batch size, optimiser and loss function throughout training, and they experienced the same number of epochs, making the comparison between the DeepAD model and the normal DNN model fair.

The performance analysis is shown graphically in Figure 11, which focuses on the accuracy and loss of training and validation.

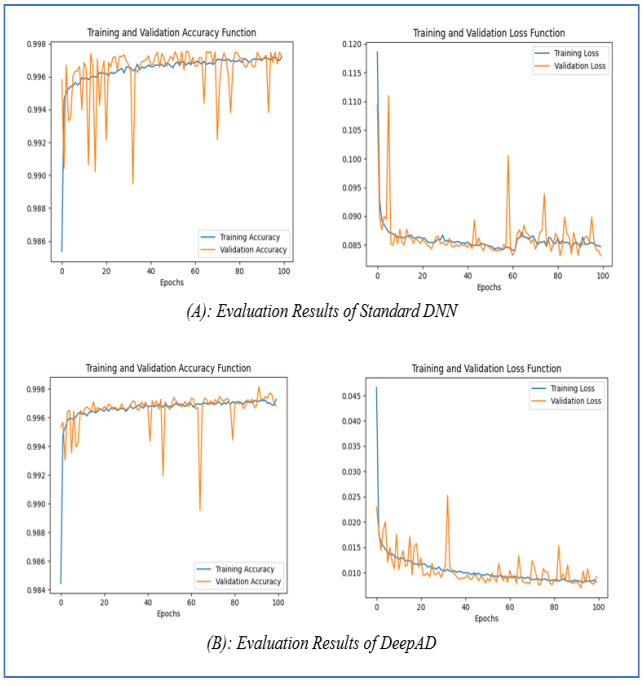

Figure 14. Visualising performance: standard DNN model and DeepAD model.

Using the sigmoid function as the typical activation function, Figure 11(A) shows the accuracy and loss throughout training and validation. Conversely, Figure 11(B) shows the accuracy and loss during training and validation that are obtained when SETAF is used. Both plots reveal that the training and validation losses dramatically dropped from 0.085 to 0.01. This drop indicates an improvement in the model's ability to minimise the difference between predicted and actual values, resulting in more accurate predictions. Lower loss values often suggest a better capacity to identify and extract the fundamental patterns and characteristics from the data.

When SETAF is used instead of the standard sigmoid activation function, the loss function is reduced further, as revealed by the analysis of the two graphs. In this case, the efficiency of the model that the SETAF helps to improve will be increased.

The testing phase then used test data to evaluate the models' performance accurately, as we calculated recall, precision, accuracy, F1 score and specificity for the standard model and DeepAD, as shown in the Figure 12.

This is due to the limitations of the sigmoid activation function in detecting DDoS attacks. These limitations lie in its inability to distinguish between different levels of DDoS attacks because DDoS attacks can vary in severity and scale, and the sigmoid activation function may not provide the flexibility needed to capture these nuances. Additionally, datasets are often unbalanced, with DDoS attacks being a minority group compared with normal traffic. Thus, the sigmoid activation function produces results that are biased towards the dominant class, which is normal traffic. This leads to lower sensitivity in detecting DDoS attacks and higher false negative detection rates.

In general, these limitations reduce the effectiveness of the sigmoid activation function in detecting DDoS attacks, especially in dealing with varying attack severity and imbalanced data distributions.

Meanwhile, SETAF enhances differentiation by providing a more flexible output range, enabling better discrimination of different levels of DDoS attacks in terms of intensity and scale. Additionally, SETAF improves sensitivity in handling imbalanced datasets commonly encountered in DDoS detection. By adjusting the exponential threshold, SETAF mitigates bias towards the dominant class (normal traffic), leading to improved sensitivity in detecting DDoS attacks and reducing false negative rates.

Moreover, SETAF maintains the nonlinear properties of sigmoid activation, facilitating the learning of complex patterns and nonlinear relationships in DDoS data. By incorporating an exponential threshold, SETAF extends the learning capability of the sigmoid function, enabling enhanced adaptation to the specific characteristics of DDoS attacks.

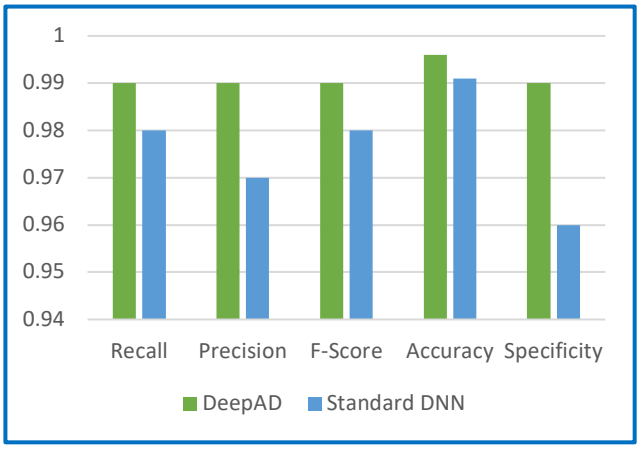

Figure 15. Comparison of performance metrics: standard DNN model and DeepAD model.

6. CONCLUSIONS

In this study, a new approach, DeepAD, is proposed, which consists of a DNN model consisting of two hidden layers. This approach is mainly based on a new activation function called SETAF. The SETAF activation function combines the features of both sigmoid and exponential functions, as well as thresholding for more nonlinearity, which works to detect different attack patterns and thus enhance security in SDN. DeepAD has demonstrated the success of SETAF in attack detection, with a detection accuracy of 0.996 for DDoS attacks and a low error rate of 0.0037. This result indicates that DeepAD is a promising solution for effective DDoS detection in SDN environments.

Future work can focus on optimising SETAF parameters to make them become more reliable and must consider evaluating the performance of SETAF in real SDN environments under different network disturbances to provide a comprehensive view of its stability and practical importance.

7. REFERENCES

[1] S. S. Mahdi and A. A. Abdullah, “Survey on Enabling Network Slicing Based on SDN/NFV”, in International Conference on Information Systems and Intelligent Applications, Springer, pp. 733–758, 2022.

[2] B. Sadkhan, M. S. Abbas, S. S. Mahdi, and S. A. Hussein, “Software-Defined Network Security-Status, Challenges, and Future trends”, in 2022 Muthanna International Conference on Engineering Science and Technology (MICEST), IEEE, pp. 10–15, 2022.

[3] M. I. Kareem and M. N. Jasim, “The Current Trends of DDoS Detection in SDN Environment”, in 2021 2nd Information Technology to Enhance e-learning and Other Application (IT-ELA), IEEE, 2021, pp. 29–34.

[4] S. D. Pande and A. Khamparia, “A review on detection of DDOS attack using machine learning and deep learning techniques”, Think India Journal, vol. 22, no. 16, pp. 2035–2043, 2019.

[5] M. I. Kareem and M. N. Jasim, “Machine Learning-Based DDoS Attack Detection in Software-Defined Networking,” in International Conference on New Trends in Information and Communications Technology Applications, Springer, pp. 264–28, 2022.

[6] Z. Zhang and Z. Zhihua, “Artificial neural network”, Multivariate time series analysis in climate and environmental research, pp.1-35, 2018.

[7] S. Sharma, S. Simone, and A. Athaiya, “Activation functions in neural networks”, Towards Data Sci, vol. 6, no. 12, pp. 310–316, 2017.

[8] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, “Method of intrusion detection using deep neural network”, in 2017 IEEE international conference on big data and smart computing (BigComp), IEEE, pp. 313–316, 2017.

[9] H. Polat, O. Polat, and A. Cetin, “Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models”, Sustainability, vol. 12, no. 3, p. 1035, 2020.

[10] R. M. A. Ujjan, Z. Pervez, K. Dahal, A. K. Bashir, R. Mumtaz, and J. González, “Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN”, Future Generation Computer Systems, vol. 111, pp. 763–779, 2020.

[11] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, “Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET”, IEEE Internet Things J, 2022.

[12] R. F. Fouladi, O. Ermiş, and E. Anarim, “A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN”, Computer Networks, vol. 214, p. 109140, 2022.

[13] S. Singh and S. K. V Jayakumar, “DDoS attack detection in SDN: optimized deep convolutional neural network with optimal feature set,” Wirel Pers Commun, vol. 125, no. 3, pp. 2781–2797, 2022.

[14] Y. Al-Dunainawi, B. R. Al-Kaseem, and H. S. Al-Raweshidy, “Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment”, IEEE Access, 2023.

[15] D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, “CICIDS-2017 dataset feature analysis with information gain for anomaly detection”, IEEE Access, vol. 8, pp. 132911–132921, 2020.

[16] Z. Zhang and M. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels”, Adv Neural Inf Process Syst, vol. 31, 2018.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980, 2014.

[18] D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, “CICIDS-2017 dataset feature analysis with information gain for anomaly detection”, IEEE Access, vol. 8, pp. 132911–132921, 2020.

[19] D. Dholakiya, T. Kshirsagar and A. Nayak, “Survey of mininet challenges, opportunities, and application in software-defined network (sdn)”, Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 2, pp.213-221, 2020.

[20] S. Kaur, J. Singh, and N. S. Ghumman, “Network programmability using POX controller,” in ICCCS International conference on communication, computing & systems, IEEE, p. 70, 2014.

Shahad A. Hussein is currently serving as an Assistant Lecturer at the University of Babylon in the College of Information Technology, located in Babylon, Iraq. She graduated with a Bachelor's degree in Information Technology with excellent grades, ranking first in her class from Babylon University in 2016. In 2017, she was awarded the first place at the national level for the Iraqi Science Day Award for her exceptional graduation research in her field of study. In 2022, she completed her M.Sc. degree from the College of Information Technology at Babylon University. Her current research field focuses on Quantum networks, Network Security, and various aspects related to the future internet.

Alharith A. Abdullah received his B.S. degree in Electrical Engineering from Military Engineering College, Iraq, in 2000. MSc. degree in Computer Engineering from University of Technology, Iraq, in 2005, and his PhD. in Computer Engineering from Eastern Mediterranean University, Turkey, in 2015. His research interests include Security, Network Security, Cryptography, Quantum Computation and Quantum Cryptography.