Federated Learning-Based Solution for DDoS Detection in SDN

1st Jovita Mateus
Computer Science Department
UWC
Cape Town, South Africa
3942909@myuwc.ac.za

2nd Guy-Alain Lusilao Zodi
Computer Science Department
NUST
Windhoek, Namibia
gzodi@nust.na

3rd Antoine Bagula
Computer Science Department
UWC
Cape Town, South Africa
abagula@uwc.ac.za

***Abstract*—One major threat to Software Defined Network (SDN) environments and other computing systems is Distributed Denial of Service (DDoS) attacks. For the longest time, conventional Machine Learning (ML) techniques have been used for DDoS detection, but they have limitations because of their centralised training requirements, which can lead to excessive bandwidth utilization, increased latency, and server overhead. Moreover, end-user data privacy and confidentiality are jeopardised by transferring the data to a central server. We present Federated Learning (FL) as a way to counteract DDoS attacks in SDN. FL reduces the need for centralised servers and lessens the limitations of conventional ML techniques by enabling decentralised training of ML models on distributed devices. FL lowers the possibility of data breaches and protects the privacy of sensitive data by training models locally on devices. We used FL to train three classifiers: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) to classify three classes of DDoS attacks, namely: UDP Flood, TCP SYN, and DNS Flood. The results demonstrate how well our FL models perform and how they can replace centralized and conventional methods for identifying DDoS attacks in SDN environments while protecting the privacy of users.**

***Index Terms*—DDoS, Federated Learning, Machine Learning, Deep Learning, Software Defined Networking, Security**

I. Introduction

Distributed Denial of Service (DDoS) attacks are widely recognized as one of the most lethal attacks against digital infrastructure. Software-Defined Networks (SDNs) have recently emerged as an integral architecture in implementing scalable, reliable, and efficient performing networks [1], replacing legacy networks that have proven insufficient over the years in handling the ever-growing number of Internet devices and the massive traffic they generate. SDNs provide programmability and dynamism for efficient network configuration via a central controller, hence optimizing network performance.

While the SDN paradigm increases convenience and better control over a network, its centralized architecture leaves room for cyberattacks such as DDoS attacks. Recent work on Intrusion Detection Systems (IDSs) focuses on the application of Machine Learning (ML) and Deep Learning (DL) instead of the native signature-based IDS which are no longer able to keep up with the increasing complexity of modern cyber-attacks [2], [3]. IDSs powered by ML and DL are known for rapidly identifying and countering sophisticated cyber threats because they can analyse vast volumes of data, spot anomalies in network traffic, and adapt to new threats.

Even though ML-based IDSs have proven to be highly accurate, they are also associated with a number of limitations. Traditional ML, for instance, relies on a ”Gather and Analyze” approach that requires moving large datasets from their original location to a centralised server [4]. Because of the massive volume of data created by consumer devices and delivered to the centralised server, this process results in increased latency, significant communication overhead, high bandwidth use, and power consumption. Furthermore, end-user data privacy and confidentiality are compromised by transferring the data to the server [5].

Federated Learning (FL), a novel ML-based technique that enables knowledge sharing while protecting the privacy of client's data [6], [7], was developed to address the shortcomings of standard ML techniques. This is accomplished by enabling edge devices to use their own data to locally train the shared model rather than sending it to the server. In order to improve the overall model, the edge devices only transmit the weighted training model parameters to the central server for aggregation. In addition to protecting data privacy, this also reduces the strain on the central server and improves the IDS's detection accuracy. The participating edge servers and clients then receives the updated global model from the aggregation server [8].

SDN's programmability and flexibility makes it an ideal technology for deploying FL solutions, thus enabling efficient communication and coordination between edge devices. Hence, this paper proposes the use of FL to design a cost-effective DDoS IDS for SDN networks. By training local models on the edge servers, we are able to address some of the key challenges, such as data privacy preservation, distribute the workload, thus reducing the strain on the main SDN controller, and improving the responsiveness of the system. This is particularly important in the context of DDoS attacks, where rapid detection and response are critical to mitigating the impact of the attack.

The remainder of this paper is organized as follows: Section II reviews the literature, and the proposed FL models are in Section III. In Section IV, we go over the experiment and performance evaluation of the proposed models, and Section

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

V concludes the paper.

II. Background

A. SDN Architecture and It's Susceptibility to DDoS Attacks

SDNs are managed by a programmable and a centralised controller which has a global view of the network and gathers network traffic information from distributed devices [1]. It then makes intelligent decisions from this information warranting the efficient use of network resources and a timely response to the ever-changing network requirements. The centralised controller resides on the control plane, which is one of the three planes in SDN. The data plane and the application plane are the other two planes. The data plane is made up of forwarding devices like switches and routers. Unlike traditional networks, the switches on the data plane lack intelligence and must rely on the controller for instructions and forwarding rules on how data should be passed throughout the network. The OpenFlow protocol [1] facilitates communication between the control plane and the data plane via the South-Bound APIs. The application plane, which includes all network applications such as load balancers, firewalls, and bandwidth optimization software, is the third plane. The North-Bound API facilitates communication between the applications on the application plane with the controller. In a multi-controlled topology, the East-West Bound API facilitates communications between different controllers.

While the flexible architecture of SDN makes it a promising technology that drives innovation, it also introduces new threat vectors and vulnerabilities. The abstraction of the data plane and control plane for example, introduces a single point of failure in case of a DDoS attack which can be exploited by attackers to compromise the whole SDN network. Authors in [9] highlighted several threat vectors that make SDN susceptible to cyber-attacks. We refer the reader to [10], and [11] for an extensive review of security challenges in SDN.

B. Taxonomy of DDoS attacks

DDoS attacks adds a many-to-one feature to Denial of Service (DoS). They accomplish viability by utilizing multiple compromised computer systems (also known as bots) as sources of attacks against a targeted server [12]. They break down applications and processes by executing a combination of exploits to overwhelm the victim. DDoS attacks prevent normal traffic from arriving at its destination by consuming much-needed resources such as memory, CPU, and bandwidth. While DDoS attacks are simple to carry out, they are difficult to entirely remove and could cause significant financial and reputational harm to businesses when successful.

DDoS attacks are broadly classified into three categories: 1) Bandwidth depletion attacks which aim to exhaust the available bandwidth of the victim system. In SDN, such attacks saturate the communication channel between the forwarding devices and the controller. The most common attacks in this category are UDP flood, ICMP flood, and DNS amplification attacks. 2) Resource depletion attacks which targets the memory, CPU, and application ports of the victim server. Common examples of resource depletion attacks are TCP SYN, HTTP Flood, and Ping of Death attacks. In SDN, these type of attacks exploits the Openflow protocol which is used by the controller to communicate with devices on the Data plane, and 3) Zero-day attacks which exploit software vulnerabilities that the software vendor is not aware of, leaving users vulnerable to exploitation [13]. Examples include the 2016 Mirai botnet [14], the 2017 IoT Reaper botnet [15], and the Memcached DDoS attack [16] which exploited Github in early 2018. These attacks can be particularly dangerous as they occur before a patch or update is released to address the vulnerability.

C. Overview of FL

Unlike traditional ML methods which requires data generated by the different devices to be gathered and moved to a central place for processing and training, FL is a distributed and a collaborative approach which is based on the idea of keeping the data where it originated and train it there. E.g. data generated in a hospital is processed and trained in the hospital. The FL training process involves multiple clients who receives and exchange model parameters with the central server. In so doing, FL allows knowledge sharing while preserving the privacy of the client's data. It also overcomes bandwidth and latency limitations for resource-poor organizations. Horizontal Federated Learning (HFL) [17], Vertical Federated Learning (VFL) [18], and Federated Transfer Learning (FTL) [19] are the three ways that one can use to train a FL model.

III. Related Work

The authors of [20] designed RL-Shield, which is a defence system to mitigate persistent link flooding attacks in SDN. They introduced a Detection algorithm (DA) which closely observe source IPs behaviours, thus tracking the network load at various links and ultimately reacting to link-flooding events. DA achieves this by using two Reinforcement Learning (RL) based statistic techniques, namely: the Dirichlet distribution and Bayesian statistics. In an event that a flooding attack is detected, the DA algorithm triggers traffic engineering to redirect normal traffic to new routes using their routing algorithms, Efficient Routing Algorithm (ERA) and the Detective Routing Algorithm (DRA). IPs sources that attempts to change their destinations after a new route has been defined are marked as link flooding bots and traffic from such sources is blocked and discarded.

The ATMoS framework in [21] leverages Deep RL to separate legitimate hosts from compromised hosts by placing them into different Virtual Networks (VNs). ATMoS consists of a security-monitoring module that inspects the status of the network by collecting data from all hosts in the network and sends alerts to the DRL agent as input. Based on these inputs, the DRL agent then makes a decision as to which virtual network a particular host should be placed. Every time the agent places a compromised or a legitimate host in the correct virtual network, it receives an award using an appropriate reward function. Their proposed has proved capable of defending the network not only from DDoS attacks, but from other attacks as well. Although ATMoS performs well, it is limited in the sense that it uses static neural networks; this implies that the number of hosts in the network must not change throughout the training and the deployment stages. This limits the DL agent in ATMoS from being used in arbitrary networks. As an improvement, the authors proposed ATMoS+ in [22] which extends the ATMoS framework making it suitable for deployment in arbitrary networks.

In the work of [23], a hybrid machine learning model for SDN-based cloud environments was proposed. The model aims at improving the performance of classification in network traffic, and it is a combination of Support Vector Machines (SV) and Self-Organising Maps (SOM). They further introduced eHIPF – a history-based IP filtering scheme, which focused on increasing the speed and detection rate of suspicious traffic targeting the cloud system. eHIPf and the two ML algorithms (SVM and SOM) were then merged to create a higher-performing security system that defends SDN-based cloud environments against DDoS attacks.

Another hybrid solution is found in [24] which combines Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) to detect various DDoS attacks in SDN networks. The model achieved an impressive accuracy of 99.33% on the CICIDS-2017 dataset and 99.93% on the CICIDS-2019 dataset.

While the models given above performed well, they all used a centralized strategy that required delivering the datasets to a central server for training.This technique is expensive in resource-constrained contexts. Secondly, leaving everything to the server can result in higher overhead and excessive latency, which can be troublesome in detecting DDoS attacks in real time. [25] presents FELIDS, a FL-based intrusion detection system that secures agricultural IoT infrastructures.The proposed IDS beats centralized versions of ML in terms of protecting the privacy of IoT device data while obtaining good detection accuracy.

Authors in [26] used four Feed Forward Neural Networks (FFNN) to classify Syn, Portmap, NetBIOS, LDAP, UDP, and UDPLag in distributed IoT environments. They trained their FL models on the CIC2019 dataset, and the best model with 129 neurons obtained an accuracy of 84.8% which is extremely poor compared to recent approaches. Similar to [25], they also used all features in the dataset which is not recommendable in a practical setup as it increases the complexity of the model. Additionally, using FNNs for DDos detection is not an ideal as FNNs are not well-suited to handle time-series data, which is crucial in DDoS attack detection.

Lastly, in [27] a technique called PCC FS that aids in the classification of DDoS attacks in SDN-enabled IIoT networks is presented. It uses a minimal-complexity CNN-MLP model that addresses the gradient disappearing problem and enhances learning efficacy through the use of residual connectivity and factorized convolution. The model surpasses previous methods with an accuracy of 98.37% on the CICDDoS2019 dataset, and a loss of 0.0639. They used twenty-five distinct features from the dataset.

IV. Proposed Approach

A. *Description of the Proposed Federated Architecture*

In our study, we used the Horizontal FL approach, which involves training models on the same set of features across all clients to ensure compatibility with the global model. We experimented with three different classifiers: CNN, DNN, and LSTM. CNN uses 1D convolutional layers to capture local patterns in the input data, which is effective for analyzing sequential data and detecting specific types of DDoS attacks. DNN consists of multiple dense layers with increasing units and an ReLU activation function, making it suitable for capturing complex patterns in the input data. LSTM on the other hand utilizes RNNs to capture long-term dependencies in the data, making it ideal for analyzing network traffic as time-series data. Our proposed federated learning architecture is presented in Fig. 1.

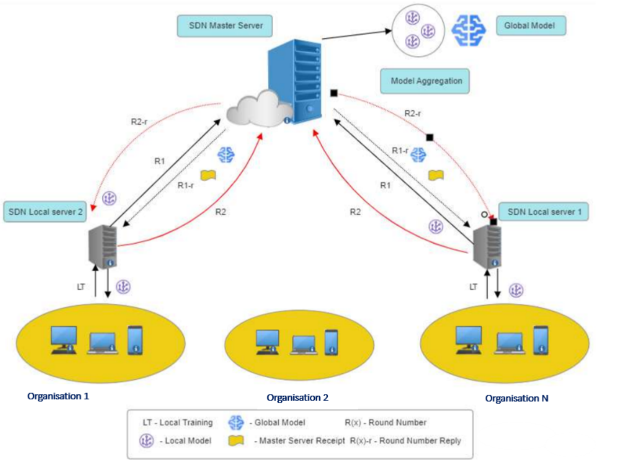


Fig. 1. Proposed Federated Learning Architecture

The steps of the FL process are described below:

1) The SDN Master Sever specifies the number of clients $N = 5$, to participate in the Federated Learning process.
2) The Master Server defines the global model $\mathbf{GB}$, also known as the model template and specify the parameters(batch size, the neural network model, initial weights, weight vectors,...).
3) It then compile the $\mathbf{GB}$ and share it with the clients - the SDN Local servers.
4) For each round $t = 1, 2, ..., 5$:
    a) For each client n=1,2,..., $N = 5$:
        i) Client n trains the shared model on its own partition of the dataset with a batch size of 32 for 10 epochs, and for 5 rounds.
        ii) At the Master server, an empty list **client_weights**, is created to store the weights of client n model.
        iii) The Master server gets the client model's weights w_n and append them to the list of weight vectors **client_weights**:

$$\mathbf{client\_weights.append(w_n)}$$

b) Update $\mathbf{GB}$ using FedAvg - a popular algorithm used in Federated Learning that averages the weights of the local models to create a global model.

i) Compute the weighted average $\mathbf{server\_weights}$ of the clients weights and store it:

$$\mathbf{server\_weights} = \frac{1}{N}\sum_{n=1}^{N} w_n$$

where w_n is the weight tensor of client n.

ii) Update the weight vector $\mathbf{w_{GB}}$ of $\mathbf{GB}$, to be equal to $\mathbf{server\_weights}$ using the `set_weights()` function:

$$\mathbf{w_{GB}} = set\_weights(\mathbf{server\_weights})$$

5) Return the final global model parameters, $w_{GB,N}$.

The training stopped when we achieved a good convergence of the global model on the testing set at round five. By using FedAvg, we were able to create a global model that was representative of the local models on each client while maintaining data privacy of the participating clients.

TABLE I
CENTRALIZED MODEL'S PERFORMANCE USING 5 FEATURES

Central Model	DDoS Type	Accuracy	Precision	Recall	F1-Score
DNN	UDP Flood	99.99%	100%	99.99%	99.99%
	TCP Sync	99.97%	99.97%	100%	99.98%
	DNS Flood	99.99%	100%	99.99%	99.99%
CNN	UDP Flood	99.99%	100%	99.99%	99.99%
	TCP Sync	99.97%	99.97%	100%	99.99%
	DNS Flood	99.99%	100%	99.98%	99.99%
LSTM	UDP Flood	99.93%	99.99%	99.93%	99.96%
	TCP Sync	99.99%	99.99%	99.99%	99.99%
	DNS Flood	99.99%	100%	99.99%	99.99%

V. Experiments and Results

A. Datasets Used

We used the Canadian Institute of Cyber-Security's (CIC) flow-based CICIDS 2019 dataset [28]. This dataset comprises of benign traffic as well as updates on new network assaults, and it fulfills real-world attack requirements. It comprises a total of 80 features that were retrieved from network traffic using the CICFlowMeter application. Botnets, Brute Force, DoS and DDoS attacks are all covered in this dataset. What distinguishes this dataset from others is that it includes popular DDoS attacks such as DNS, LDAP, MSSQL, and TFTP, as well as UDP, UDP-Lag SYN, and NetBios. Web, HeartBleed, and Infiltration attacks are also included.

B. Features Selection and Data Prepossessing

As DDoS attacks are executed differently and have different behavioral patterns, it is crucial to use the right features when building classification models for these attacks. For each DDoS type, we used the best five features adopted from [28], which were selected based on their importance scores/weights. By selecting only the most significant features, the models are simplified, and made more interpretable, their accuracy is improved, and overfitting is minimised.

For data preprocessing, we replaced null values with zero using the fillna() method, and unnecessary columns were dropped using the drop() method. We then used the StandardScaler technique to normalize the data. Afterward, we applied Label Encoding to convert categorical labels into numerical labels - this step is crucial as it makes it easier for the models to process the labels. We then split the data into training (80%) and testing sets (20%) to evaluate the performance of the models on unseen data. Lastly, the perfomance of the models were evaluated based on the following matrices: accuracy, precision, recall, and f1-score.

C. Experiment Setup

The simulations were done on a Linux-based high-performance computing server running Centos 7.82003 and powered by an Intel® Xeon® E5-2603 v4 CPU running at 1.70 GHz. TensorFlow and Keras were used to construct, test, and train our deep learning models, while Pandas and other essential libraries were used for data processing and analysis, and Matplotlib and Seaborn were used for data visualization.

D. FL Model's Performance

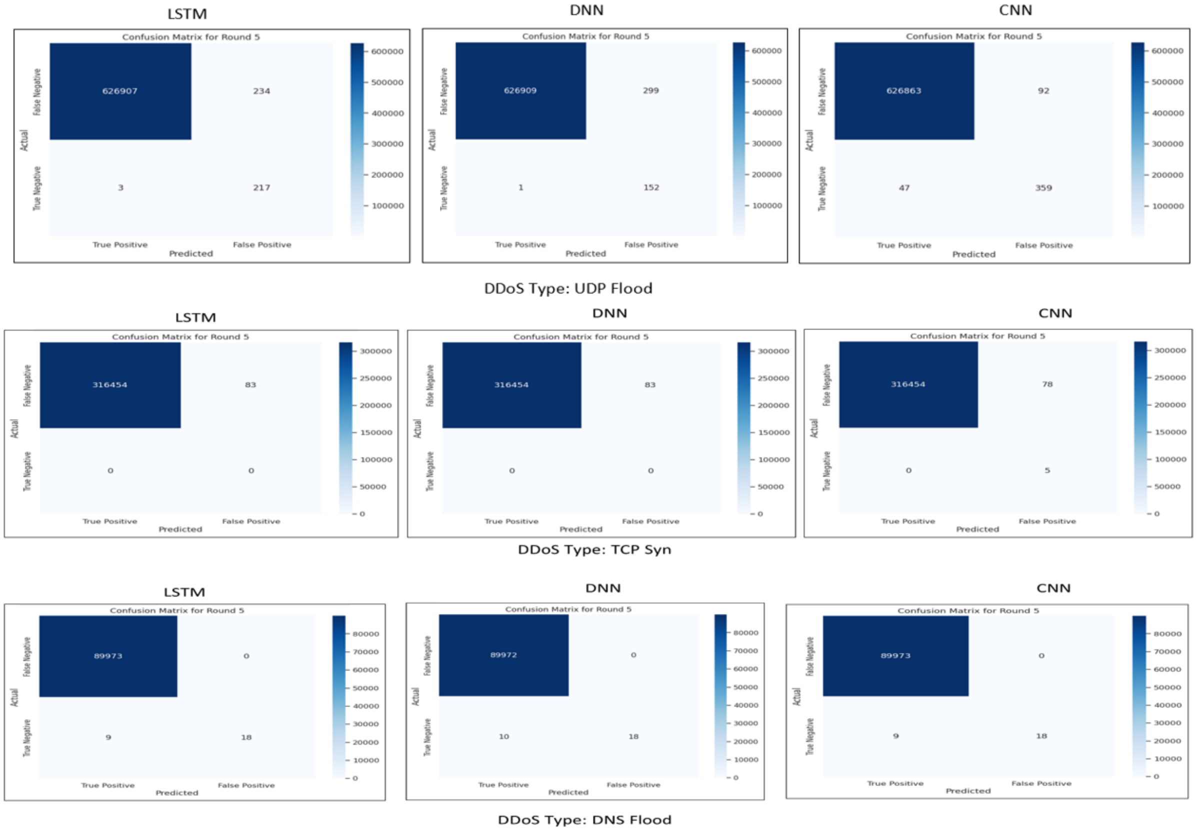
As seen in table II, all three FL classifiers (DNN, CNN, and LSTM) exhibit high performance in detecting different DDoS attacks. The classifiers demonstrate high accuracy, precision, recall, and F1-score, with accuracies ranging from 99.93% to 99.99%. We observed that the perfomance of the global models improves with each round of training. This shows that the global models were able to aggregate the knowledge learned by each of the five clients over time, so as more rounds of training are completed, the global models becomes more accurate and are better able to generalize to new data.

The DNN classifier performs best in detecting DNS Flood attacks, with an accuracy of 99.99%, while the CNN classifier performs best in detecting TCP SYN and DNS Flood attacks, with accuracies of 99.99% as well. The LSTM classifier also performs greately in detecting all three types of DDoS attacks, with accuracies ranging from 99.96% to 99.99%. The heatmaps in Fig. 2 indicates the perfomance of the global models by comparing the predicted values with the actual values on the test data in the final round, round five.

In table I, we performed centralised training using the same classifiers and the same dataset. We did this so that we could compare the perfomance of different classifiers when trained in a centralised way, and when trained in a federated way. We observed that in most cases, the federated global models achieved similar accuracy as the centralised models, proving yet again that there is no need to risk the privacy of the client's data by sending it to the central controller, consequently burdening it when we could achieve silimar perfomance with

TABLE II
FL MODEL'S PERFORMANCE USING 5 FEATURES WITH 5 CLIENTS

FL Global Model	DDoS Type	First Round				Final Round - Round 5			
		Accuracy	Precision	Recall	F1- score	Accuracy	Precision	Recall	F1 - score
DNN	UDP Flood	99.94%	99.99%	99.94%	99.97%	99.95%	99.95%	99.99%	99.98%
	TCP Sync	99.99%	99.99%	99.99%	99.99%	99.97%	99.97%	100%	99.99%
	DNS Flood	99.99%	100%	99.99%	99.99%	99.99%	100%	99.99%	99.99%
CNN	UDP Flood	99.75%	99.97%	99.78%	99.84%	99.98%	99.99%	99.99%	99.99%
	TCP Sync	99.98%	99.98%	100%	99.99%	99.97%	99.99%	99.99%	99.99%
	DNS Flood	99.99%	100%	99.99%	99.99%	99.99%	100%	99.99%	99.99%
LSTM	UDP Flood	99.93%	99.99%	99.93%	99.96%	99.96%	99.96%	99.99%	99.98%
	TCP Sync	99.97%	99.99%	100%	99.99%	99.97%	99.97%	100%	99.99%
	DNS Flood	99.98 %	100% %	99.99%	99.99 %	99.99%	100%	99.99%	99.99%



DDoS Type: UDP Flood

DDoS Type: TCP Syn

DDoS Type: DNS Flood

Fig. 2. Confusion Matrix for the FL Global Models on Three DDoS Types after the Final Round - R5)

federated learning. In the cases where the centralised models performed better, the difference was relatively small.

E. Proposed FL models Vs State-of-the-Art solutions

Our proposed FL models outperform some existing solutions as follows: the FFNNs in [26] which achieved 84.8% accuracy and trained with all features in the dataset, PCC FS introduced in [27] with 98.37% accuracy using 25 features. Training models with too many features only increases the complexity of the model, and not necessarily its performance. In contrast, we were able to achieve good results in few rounds by using the best but minimal 5 features for each DDoS type, as adopted from [28]. These features were selected based on their importance scores, which quantify the contribution of each feature towards the model's overall accuracy. The recent work in [29] studies the limitations of the FEDAVG algorithm and proposes a new approach called FLAD which could be used as a guideline in proposing better FL solution for detecting DDoS attacks and other cyber attacks in networks today. While some existing solutions achieved good performance with more complex models and longer training times, our approach demonstrates that good results can be obtained with minimal features and fewer training rounds.

VI. Conclusion

Detecting DDoS attacks in SDN is crucial for ensuring the availability and reliability of network services and addressing the security challenges introduced by SDN's centralised architecture. This paper contributes to addressing this issue by proposing a FL-based approach to detect DDoS attacks in SDN environments while preserving data privacy. Our comparison of federated global models with centralised models shows that there is no need to risk the privacy of the client's data by sending it to the central server. By decentralizing the training process and only allowing the main controller to do model aggregation, we reduce the burden of the main SDN controller, minimize latency, reduce bandwidth usage, and allow for fast detection of malicious attacks. The LSTM classifier performs the best in detecting all three types of DDoS attacks, with accuracies ranging from 99.96% to 99.99% on the CIC2019 dataset and 99.20% on the CIC2017 dataset. While the current paper focuses on the performance of Federated deep learning models, future work will look deeper into the challenges and limitations of Federated learning in SDN environments. We also plan to enhance this work by safeguarding the training data of individual clients against data poisoning attacks using various privacy-preserving techniques. Additionally, we intend to explore the use of Vertical Federated learning to investigate how the results will differ from our current approach.

References

[1] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2015, doi: 10.1109/JPROC.2014.2371999.

[2] B. Ahmad, W. Jian and Z. Anwar Ali, "Role of Machine Learning and Data Mining in Internet Security: Standing State with Future Directi]ons", *Journal of Computer Networks and Communications*, vol. 2018, 2018.

[3] E. Hodo , X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C.Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, Submitted to *ACM Survey*, 2017, [Online]. Available:http://arxiv.org/abs/1701.02145.

[4] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos and N. C. Fernandes, "A Survey on Securing Federated Learning: Analysis of Applications, Attacks, Challenges, and Trends," in *IEEE Access*, vol. 11, pp. 41928-41953, 2023, doi: 10.1109/ACCESS.2023.3269980.

[5] M.N. Ali, M. Imran, M.S.U. Din, and B.S. Kim, "Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network," *Applied Sciences*, vol. 13, no.3, pp.1-21, 2023.

[6] P. M. Mammen, "Federated learning: Opportunities and challenges", arXiv:2101.05428, 2021.

[7] J. Wen et al., "A survey on federated learning: Challenges and applications," *Int. J. Mach. Learn. Cybern.*, vol. 14, pp. 1–23, Nov. 2022.

[8] K. Zhang, X. Song, C. Zhang and S. Yu, "Challenges and future directions of secure federated learning: A survey", *Front. Comput. Sci.*, vol. 16, no. 5, pp. 1-8, Oct. 2022.

[9] D. Kreutz, F. M. Ramos and P. Verissimo, "Towards secure and dependable software-defined networks", *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.,* pp. 55-60, 2013.

[10] S. Scott-Hayward, S. Natarajan and S. Sezer, "A Survey of Security in Software Defined Networks," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623-654, Firstquarter 2016, doi: 10.1109/COMST.2015.2453114.

[11] D. B. Rawat and S. R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325-346, Firstquarter 2017, doi: 10.1109/COMST.2016.2618874.

[12] A. P. Fajar and T. W. Purboyo, "A survey paper of distributed denial-of-service attack in software defined networking (sdn)," *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 476–482, 2018.

[13] K. Radhakrishnan, R. R. Menon and H. V. Nath, "A survey of zero-day malware attacks and its detection methodology", *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India, 2019, pp. 533-539, doi: 10.1109/TENCON.2019.8929620.

[14] X. Zhang, O. Upton, N. L. Beebe and K.-K.-R. Choo, "IoT botnet forensics: A comprehensive digital forensic case study on mirai botnet servers", Forensic Sci. Int. Digit. Invest., vol. 32, Apr. 2020.

[15] A. Chadd, "DDoS attacks: past present and future", Network Security, pp. 13-15, 2018.

[16] K. Singh and A. Singh, "Memcached DDoS Exploits: Operations, Vulnerabilities, Preventions and Mitigations," 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 2018, pp. 171-179, doi: 10.1109/CCCS.2018.8586810.

[17] K. Zhang, X. Song, C. Zhang and S. Yu, "Challenges and future directions of secure federated learning: A survey", *Front. Comput. Sci.*, vol. 16, no. 5, pp. 1-8, Oct. 2022.

[18] B. Gu, A. Xu, Z. Huo, C. Deng and H. Huang, "Privacy-Preserving Asynchronous Vertical Federated Learning Algorithms for Multiparty Collaborative Learning," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6103-6115, Nov. 2022, doi: 10.1109/TNNLS.2021.3072238.

[19] Li, Bo, et al. "Research on multi-UAV task decision-making based on improved MADDPG algorithm and transfer learning." *International Journal of Bio-Inspired Computation* 18.2 (2021): 82-91.

[20] A. Rezapour and W. -G. Tzeng, "RL-Shield: Mitigating Target Link-Flooding Attacks using SDN and Deep Reinforcement Learning Routing Algorithm," *in IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2021.3118081.

[21] I. Akbari, E. Tahoun, M. A. Salahuddin, N. Limam and R. Boutaba, "ATMoS: Autonomous Threat Mitigation in SDN using Reinforcement Learning," *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1-9, doi: 10.1109/NOMS47738.2020.9110426.

[22] H. Tsang, I. Akbari, M. A. Salahuddin, N. Limam and R. Boutaba, "ATMoS+: Generalizable Threat Mitigation in SDN Using Permutation Equivariant and Invariant Deep Reinforcement Learning," *in IEEE Communications Magazine*, vol. 59, no. 12, pp. 105-111, December 2021, doi: 10.1109/MCOM.009.2100389.

[23] T. V. Phan and M. Park, "Efficient Distributed Denial-of-Service Attack Defense in SDN-Based Cloud," in *IEEE Access*, vol. 7, pp. 18701-18714, 2019, doi: 10.1109/ACCESS.2019.2896783.

[24] J. Mateus, G. -A. L. Zodi, A. Bagula and M. Landry, "Building DDoS Resilient SDNs Using Hybridised Deep Learning Methods," 2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Windhoek, Namibia, 2023, pp. 1-7, doi: 10.1109/ETNCC59188.2023.10284972.

[25] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K.-R. Choo and M. Nafaa, "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things", *J. Parallel Distrib. Comput., vol. 165, pp. 17-31, Jul. 2022.*

[26] E. C. P. Neto, S. Dadkhah and A. A. Ghorbani, "Collaborative DDoS Detection in Distributed Multi-Tenant IoT using Federated Learning," *2022 19th Annual International Conference on Privacy, Security and Trust (PST)*, Fredericton, NB, Canada, 2022, pp. 1-10, doi: 10.1109/PST55820.2022.9851984.

[27] A. Zainudin, R. Akter, D. -S. Kim and J. -M. Lee, "FedDDoS: An Efficient Federated Learning-based DDoS Attacks Classification in SDN-Enabled IIoT Networks,"*2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of, 2022, pp. 1279-1283, doi: 10.1109/ICTC55196.2022.9952610.

[28] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.

[29] R. Doriguzzi-Corin and D. Siracusa, "FLAD: Adaptive Federated Learning for DDoS Attack Detection", May 2022.