DDoS Attacks Detection and Mitigation in 5G and Beyond Networks: A Deep Learning-based Approach

Badre Bousalem, Vinicius Silva, Rami Langar, Sylvain Cherrier

DDoS Attacks Detection and Mitigation in 5G and Beyond Networks: A Deep Learning-based Approach

Badre Bousalem*, Vinicius F. Silva*, Rami Langar*†, Sylvain Cherrier*
*University Gustave Eiffel, LIGM-CNRS UMR 8049, F-77454, Marne-la-Vallée, France
†Software and IT Engineering Department, École de Technologie Supérieure (ÉTS), Montréal, QC H3C1K3, Canada
E-mails: {badre.bousalem, vinicius.fonsecaesilva, rami.langar, sylvain.cherrier}@univ-eiffel.fr ; rami.langar@etsmtl.ca

***Abstract*—Network slicing, where a single physical network is partitioned into several fit-for-purpose virtual networks with different degrees of isolation and quality of service (QoS), is a key enabler of 5G and beyond mobile networks. However, it is prone to security threats such as Distributed Denial-of-Service (DDoS) attacks. In this paper, we propose a solution based on Deep Learning (DL) that detects such attacks, and then creates a sinkhole-type slice with a small portion of physical resources to isolate and mitigate the attackers' action. Using our 5G prototype based on *OpenAirInterface*, we evaluate our approach by comparing several DL models in terms of detection accuracy, false positive rate, execution time, among other Machine Learning-related metrics. We also assess the performance of created 5G network slices in terms of benign/malicious users' throughput, as well as the processing time during the slicing operations. Results show that our approach is able to detect DDoS attacks in a timely manner with an accuracy of almost 97% and a false positive rate of less than 4%. We also show that our approach decreases the network throughput for the malicious users by a factor of 15, while maintaining a high network throughput for benign users.**

***Index Terms*—5G, Slicing, Cybersecurity, Deep Learning.**

I. Introduction

5G networks powered by network slicing features uncover several challenges in the cybersecurity context, which have not been properly addressed yet by 5G standards. In such a context, heterogeneous nodes demand different network services and present intermittent connections, where traditional security approaches are not always accurate.

Network slicing allows to create multiple logical instances of the physical network, the so-called "network slices", ensuring traffic isolation among them, and tailoring the network resources of each slice to a specific class of applications, by leveraging the concepts of Software Defined Networking (SDN) and Network Function Virtualization (NFV). Network slicing enables the coexistence of a wide range of mobile services in the same network infrastructure.

Given the above context, it is of paramount importance having a simple, yet controlled environment that focuses on 5G cybersecurity applications and leverages Deep Learning (DL)-based network slicing to detect and mitigate attacks from malicious users. To achieve this, we present in this paper a DL-based approach that detects Distributed Denial of Service (DDoS) attacks, hence creating a sinkhole-type slice with a small portion of physical resource blocks (PRBs) to isolate and mitigate such attacks. A DDoS attack on a slice could indeed lead to the exhaustion of available resources and a breach of the availability of PRBs on the slices.

To gauge the effectiveness of our approach, we performed experiments using our 5G prototype, with a custom dataset based on DDoS attack samples. Our prototype, based on *OpenAirInterface* [1] and the *FlexRAN* controller [2], allows creating network slices and managing PRBs dynamically according to the users' behavior, while considering the inputs from a northbound SDN application. In particular, in the 5G security context, we design a network slicing security policy leveraging DL. We compare several developed DL models in terms of detection accuracy, false positive rate, execution time, among other Machine Learning (ML)-related metrics. We also assess the performance of created 5G network slices in terms of benign/malicious users' throughput, as well as the processing time during the slicing operations (i.e., the time needed to create the sinkhole-type slice and to migrate the malicious users to it). Obtained results show that our developed DL models are able to achieve a detection accuracy of almost 97% and a false positive rate of less than 4%. In addition, we show that our approach decreases the network throughput for the malicious users by a factor of 15, while maintaining a high throughput for benign users.

The remainder of this paper is organized as follows. Section II presents the related work. Section III provides our methodology for DDoS attacks detection and mitigation using DL models. Section IV shows the experimental setup, describing our 5G prototype and the performance metrics considered to compare the DL models' effectiveness, as well as to assess the slices' performance. Obtained results are discussed in Section V. Finally, in Section VI, we provide our final remarks and future work.

II. Related Work

Several works have been proposed in the literature to optimize security solutions whilst considering 5G network requirements. However, there is still a lack of works that consider network slicing along with ML and DL-based

approaches, to protect 5G networks from cyber-attacks such as DDoS.

As stated earlier, it is essential to have a simple yet controlled environment composed by real devices that focuses on 5G cybersecurity applications to validate the effectiveness of the proposed solutions. A number of prototypes have been proposed in the literature to address the challenges imposed by sliced 5G networks. In view of this, authors in [3] described how to use *FlexRAN* [2] and *OpenAirInterface* (OAI) [1] to deploy a Cloud Radio Access Network (C-RAN) architecture in an automated and virtualized way. Authors in [4] described their experience building a 5G prototype that uses dynamic network slicing for Internet of Things (IoT) and Enhanced Mobile Broadband (eMBB) services. Authors in [5] presented their prototyping platform called SCOPE, that integrates an open source container for instantiating softwarized and programmable cellular network elements, accompanied with an emulation module for testing new solutions in real-world deployments. SCOPE also has a data collection module that can be used for ML-based solutions, with multiple APIs that allow users to control network functionalities in real-time.

Differently from the works presented above, our prototype mainly focuses on DL-based solutions to protect 5G and beyond mobile networks from security threats, leveraging real-time network slicing.

In the ML and DL context, relevant works include the application of DL and Deep Reinforcement Learning to predict the network load [6] [7], classify traffic [8] [9], control and configure 5G platforms automatically [10], and detect and mitigate cross-layer attacks in wireless networks through Bayesian learning [11].

The closest works compared to our approach are the ones presented in [12] and [13]. In view of this, authors in [12] propose an optimization model to proactively mitigate DDoS attacks in the 5G Core Network (CN) through on-demand intra/inter slice isolation, hence guaranteeing network performance requirements for 5G CN slices. However, this referred work only focuses on protecting 5G CN slicing, where an on-demand network service/function distribution between slices occurs. In our work, we focus rather on Radio Access Network (RAN) slicing, where physical resources are shared between connected benign/malicious users. In addition, as opposed to [12], our proposal leverages DL-based techniques for DDoS attacks detection and mitigation, which ensure a continuous interaction with the environment, by analyzing both benign and malicious users' behavior, to improve the accuracy of our models.

Similarly, authors in [13] propose *DeepSecure*, a framework that uses a Long Short Term Memory (LSTM) DL-based model to classify users' network traffic as DDoS or benign, as well as a model that predicts the appropriate slice for users previously classified as benign. The main drawback of the authors' proposal is that they design and evaluate the performance of their framework using a dataset not directly related to a 5G-based environment. In our work, the training of each DL model is directly based on data collected from real devices using our 5G prototype, hence validating the effectiveness of our proposal under a realistic scenario. In addition, differently from [13], we consider attack mitigation through a proactive isolation of malicious users, by moving them to a sinkhole-type slice with few physical resources, hence protecting physical resources previously allocated to benign users.

III. Methodology

In this section, we present our methodology for DDoS attacks detection and mitigation. Indeed, in order to detect and mitigate those attacks, more specifically *TCP SYN* flood attacks in the 5G security context, we initially developed and trained 100 DL models, while taking into account the performance, as well as the inference and training times for each model. We have limited the number of developed models to 100 since all remaining models beyond these ones gave similar results. This was confirmed through preliminary tests obtained with the 10 latest models developed. The DL models were developed based on supervised learning and Convolutional Neural Networks (CNN), using several tools such as *Tensorflow* [14], *Keras* [15] and *Lucid* [16].

To train, validate and test our DL models, we first built a labeled custom dataset that contains synthetic DDoS attack samples generated by *Metasploit* [17] and *Mausezahn* [18], as well as benign traffic samples generated through *iperf3* [19] using our 5G prototype, which will be described in Section IV-A. Specifically, the dataset is in the format of a PCAP file of $\sim$3.7 MB, with $\sim$300,000 lines that resemble real world data. The lines are represented by mixed network packets sent from benign users (mainly *HTTP* and *TCP* packets) and malicious users (*TCP SYN* flood packets), with each line consisting of 7 columns: a) The packet number; b) The time when the packet was captured; c) The source IP address; d) The destination IP address; e) The protocol that was used; f) The packet length in bytes; and g) The packet type. To generate this traffic, we deployed two real users using Commercial Off-The-Shelf (COTS) smartphones with spoofed IP addresses, and three virtual ones.

After collecting all samples, the obtained dataset is pre-processed to make it suitable for all DL models. The pre-processing steps consist of normalizing, merging and balancing the benign and the DDoS attack samples, and then splitting the whole dataset between training, validation and test sets. In our case, we applied 80% of the dataset to the training set (of which 10% is used for the validation set), and the remaining 20% was applied to the test set. Once done, we started the training and validation of the DL models, by setting the sigmoid function (Eq. 1) as the activation function. The sigmoid function restricts the value of the DDoS attack detection probability between 0 and 1.

$$y = \frac{1}{1 + e^{-x}} \tag{1}$$

We then start tuning, for each model, a selected subset of *hyper-parameters* presented below. Such a subset is chosen

since it provides the highest impact on the DL models' performance during our preliminary experiments:

- **Learning Rate:** Controls the weights of neural networks based on the loss gradient, and defines how quickly the neural network is updated;
- **Batch Size:** Corresponds to the number of training examples used in one iteration;
- **Number of Convolutional Filters:** Corresponds to the number of kernels used in a CNN;
- **Height of Convolutional Filters:** Refers to the height of the filters in a CNN;
- **Time Window:** Simulates the capture process of online systems by splitting flows into subflows of fixed duration;
- **Max. Num. Packets:** Refers to the maximum number of packets recorded in a flow over time.

In addition to the tuning of the aforementioned hyper-parameters in the training phase, we also tune the following parameters: a) Number of epochs; b) Dropout; and c) Regularization. From all 100 DL models developed, we picked the five ones that had the best performance results in our preliminary experiments. We denote each model by the term *DLM-X*, i.e., *DL Model number X*.

After testing multiple dropout values in preliminary experiments, we set this parameter to 40 for all five best DL models since it provides optimal performance results. The same can be said for the regularization parameter, which we set to L2 instead of L1.

The main parameter that tends to have the most impact in the performance of each DL model is the number of epochs. According to our preliminary experiments, we have adjusted this parameter to 10, 50, 100, 200 and 300 for our studied DL models: *DLM-1*, *DLM-2*, *DLM-3*, *DLM-4* and *DLM-5*, respectively, in order to obtain their optimal performance.

Algorithm 1 shows the resulting network slicing security policy, which uses model *DLM-X* ($X \in [1..5]$) to detect the occurrence of a DDoS attack. Note that once an attack event is detected, a sinkhole-type slice is automatically created by a dedicated Software-defined RAN controller (SDN *FlexRAN* in our case, as described in Section IV-A), while allocating the smallest possible portion of PRBs. Then, malicious users will be moved to this particular slice, thus mitigating their actions.

Algorithm 1 Slicing Security Policy for DDoS Attack Detection and Mitigation

P_{cap} = Capture packets at the PGW for t_{cap} seconds
$P_{cap,\text{SYN}}$ = Filter packet capture by *TCP SYN* packets
DLM_{out} = Run *DLM-X* with $P_{cap,\text{SYN}}$ as input
if DLM_{out} equals 1 **then** ▷ DDoS attack detected
    Create sinkhole-type slice at the RAN
    IP_{addrs} = Get list of connected users' IP addresses
    for each IP in IP_{addrs} **do**
        $P_{cap,\text{IP}}$ = Filter $P_{cap,\text{SYN}}$ by one user's IP address
        $DLM_{out,\text{IP}}$ = Run *DLM-X* with $P_{cap,\text{IP}}$ as input
        if $DLM_{out,\text{IP}}$ equals 1 **then** ▷ Attacker identified
            Move DDoS attacker to the sinkhole-type slice
        end if
    end for
end if

IV. Experimental Setup

In this section, we describe our experimental setup, with a detailed description of our 5G prototype and the performance metrics considered in our analysis.

A. 5G Prototype Description

To emulate the cellular network elements (i.e., the Core Network – CN – and the Radio Access Network – RAN), we use *OpenAirInterface* (OAI). OAI is an open-source software developed by Eurecom to support mobile telecommunication systems like 4G Long Term Evolution (LTE) and 5G New Radio (NR).

To deploy the CN elements, composed in our scenario by the Home Subscriber Server (HSS), the Mobility Management Entity (MME), the Serving Gateway (SGW) and the Package Gateway (PGW), we use a Dell Precision 3551 laptop, which provides Internet access. On the other hand, for the RAN part, we use a Dell Optiplex 7780 AIO desktop PC, which in turn is connected to a USRP X310 card. The USRP card is responsible to emulate the Radio Unit (RU), thus creating a communication interface between the RAN and the users, represented here by two different COTS smartphone models: Samsung Galaxy S20 FE 5G and Samsung Galaxy A42 5G.

To emulate the attack scenario, we use one Dell Precision 3551 and one Dell Latitude 5490, each connected to Wi-Fi hotspots created by the smartphones, which in turn connect to the 5G network. The attacks are generated using *Mausezahn* on the first laptop, while the second laptop generates benign traffic simultaneously through *iperf3*.

Our prototype makes also use of the SDN *FlexRAN* controller, that enables remote control of the OAI MAC layer through a specific southbound interface (SBI), based on *Google Protobuf* [2]. On the top of *FlexRAN*, we have developed a northbound *Slicing APP* application, which enables the network administrator to deploy network slicing policies in a user-friendly and abstracted way. The network slicing policies have indeed a dedicated tab, where the network administrator can configure and trigger the proper policy according to network changes and end-to-end service requirements. In particular, in the 5G security context, we have implemented the network slicing security policy leveraging DL techniques, as described in Algorithm 1. Please refer to Fig. 1 for illustration.

B. DL Models' Performance Metrics

In this section, we present the performance metrics used to evaluate the developed DL models in our 5G prototype. Our objective here is to compare both validation and test results obtained with our custom dataset, in order to determine if there is any overfitting or underfitting in one or more DL models.

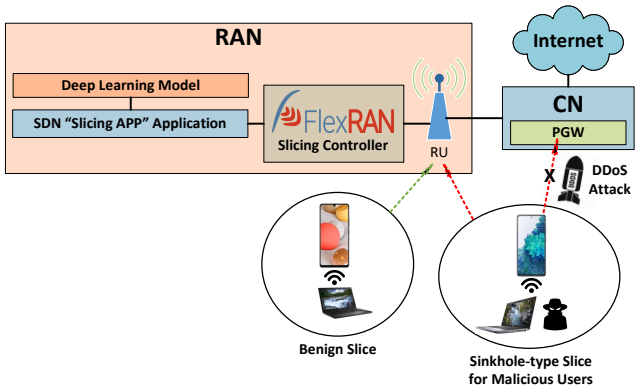

Fig. 1. DDoS attack scenario implementation in the 5G platform.

To do so, we use the following metrics, defined in terms of the total number of predictions as true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN):

- **The Accuracy of the Model (ACC):** How precise a model is to detect benign and malicious traffic:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2}$$

- **False Positive Rate (FPR):** The probability that a benign traffic will be classified as malicious:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{3}$$

- **Precision (PPV):** The model's positive predictive value:

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

- **True Positive Rate - Recall (TPR):** The probability that an actual attack will be detected:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

- **F1 Score (F1):** The harmonic mean between precision (PPV) and recall (TPR):

$$\text{F1} = 2 \times \frac{\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} \tag{6}$$

- **Execution Time (T):** The execution time (in seconds) of a DL model on a validation or test set. In our context, it is also the time needed to detect a DDoS attack.

C. Benign and Sinkhole-type Slices' Performance Metrics

To evaluate the performance of benign/sinkhole-type slices, we consider the following metrics:

- **Throughput:** The amount of bits per second sent between the CN's PGW and a DDoS attacker (respectively, a benign user) for the sinkhole-type slice (respectively, for the benign slice);
- **DDoS Attack Success Rate:** The percentage of packets sent by a malicious user that successfully arrived at the CN's PGW during a DDoS attack;
- **Slicing Operation Time:** The processing time needed to interact with the *FlexRAN* controller in order to create a sinkhole-type slice and move a malicious user to it.

V. Experimental Results

In this section, we present and discuss the main results obtained through our 5G prototype by implementing the scenario illustrated in Fig. 1. We start by evaluating the aforementioned DL models (i.e., *DLM-X*, $X \in [1..5]$) in terms of ACC, FPR, PPV, TPR, F1 and T metrics presented in the previous section. Then, we present the performance evaluation of the benign and sinkhole-type slices in terms of network throughput, DDoS attack success rate, and slicing operation time.

Note that in [20], we proposed a demo that describes how we emulated a DDoS attack during our experiments, as illustrated in Fig. 1. A description of the demo steps can also be found in a video publicly available [21]. By using the *Speedtest* tool, we demonstrated that the network throughput for the malicious user is downgraded from ∼30 Mbps to ∼2 Mbps (i.e., a decrease by a factor of 15), while that of the benign user is maintained high at ∼30 Mbps.

A. DL Models' Performance Evaluation

Table I shows the performance results obtained for all DL models with the validation (V) and test (T) sets as inputs.

TABLE I
DL Models Performance - Validation and Test Sets

DL Model	Set	Performance Metric					
		ACC	FPR	PPV	TPR	F1	T
DLM-1	V	0.5437	0.4190	0.5286	0.5590	0.5438	0.067
	T	0.5382	0.4776	0.5231	0.5537	0.5384	0.061
DLM-2	V	1.0000	0.0000	1.0000	1.0000	1.0000	0.053
	T	0.8476	0.1595	0.8407	0.8545	0.8476	0.049
DLM-3	V	1.0000	0.0000	1.0000	1.0000	1.0000	0.062
	T	0.9252	0.0877	0.9127	0.9379	0.9253	0.053
DLM-4	V	1.0000	0.0000	1.0000	1.0000	1.0000	0.051
	T	0.9653	0.0384	0.9621	0.9687	0.9654	0.041
DLM-5	V	0.9943	0.0108	0.9897	0.9989	0.9943	0.058
	T	0.6181	0.3869	0.6142	0.6222	0.6182	0.044

For *DLM-1*, we can see that ACC, PPV, TPR and F1 are low for both the validation and test sets, in comparison to the other models. Such a result shows that this model is underfitting, which prevents us to deploy it in our 5G prototype.

For *DLM-2*, ACC, PPV, TPR and F1 are higher in comparison with *DLM-1*. Additionally, FPR is lower, specially with the validation set as input, where false positives did not occur. *DLM-2*'s higher performance in terms of FPR is confirmed with the test set as input, being 66.6% lower in comparison with *DLM-1*.

In the midterm, *DLM-5* also performs better compared to *DLM-1*, with the validation set as input. With the test set as input, *DLM-5* remains better than *DLM-1*. However, its performance decreases in terms of ACC, PPV, TPR and F1, along with a higher FPR, being this last one almost 36 times higher in comparison with the validation set result. Such a behavior shows the presence of overfitting, which also

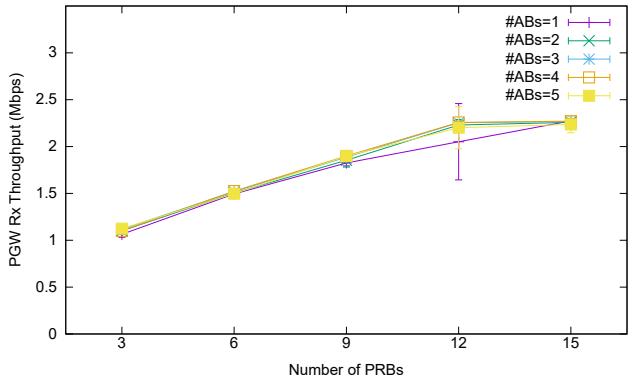

Fig. 2. Sinkhole-type slice: PGW Rx throughput.

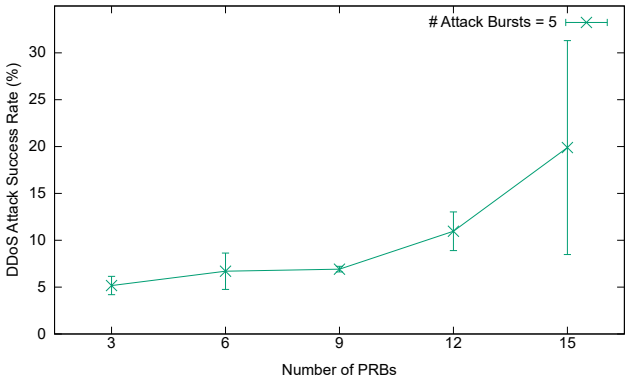

Fig. 3. Sinkhole-type slice: DDoS attack success rate.

prevents us to use this model to properly detect and mitigate DDoS attacks.

Finally, *DLM-3* and *DLM-4* models provide better results in comparison with the previous ones, both with the validation and test sets as inputs, in terms of all performance metrics. Such results demonstrate the absence of underfitting and overfitting on these models.

According to these observations, we give special attention to *DLM-4*, since it shows the best results for all metrics, which means it is the most suitable one to be deployed to detect and mitigate DDoS attacks. Considering the test set as input, *DLM-4* is able to achieve almost 97% in ACC, PPV, TPR and F1, and less than 4% in FPR. Considering the same set and comparing *DLM-4* with *DLM-1*, which is the worst model among all five DL models, *DLM-4* is 79.36%, 83.92%, 74.95% and 79.31% higher in terms of ACC, PPV, TPR and F1, respectively. *DLM-4* also shows a FPR 91.96% lower and executed 32.79% faster than *DLM-1*.

B. Benign/Sinkhole-type Slices' Performance Evaluation

To evaluate the slices' performance, we run each scenario five times, and collect the average value of each performance metric with a 95% confidence interval. We vary both the benign and sinkhole-type slices' sizes, in terms of number of allocated PRBs. In our experiments, the 5G network is deployed with a total number of 48 PRBs. We hence vary the number of allocated PRBs in the benign slice between 33 and 45, and between 3 and 15 in the sinkhole-type slice, both with a step of 3 PRBs.

It is worth noting that for the sinkhole-type slice, besides the number of allocated PRBs, we also vary the number of packets sent in a DDoS attack through the *Mausezahn* tool, by varying what we define here as the "Number of Attack Bursts" (#ABs). In our experiments, this parameter is varied between 1 and 5. For each attack burst, 51,255 *TCP SYN* packets are generated. Such an amount is a result of using different combinations of fake source IP addresses in the whole 5G network address range (set between 12.1.1.1 and 12.1.1.254 in our prototype), along with source ports ranging between 800 and 1000 and the destination port fixed to 80.

Fig. 2 shows the impact of varying the number of allocated PRBs for the sinkhole-type slice on the average PGW Rx throughput for several values of Attack Bursts (#ABs). Remember that the PGW is the victim of the DDoS attacks. As we can see, the volume of traffic perceived by the PGW is similar for any number of attack bursts, for each sinkhole-type slice size. This can be explained by the fact that even for the smallest number of attack bursts (#ABs=1), the amount of packets is already high enough to overload the wireless channel. In addition, we can observe a slight increase in the average throughput for sinkhole-type slices with sizes between 3 and 12 PRBs, followed by a slight stabilization between 12 and 15 PRBs. Such an increase for bigger slices is expected since more physical resources are allocated to them, which allow higher traffic volumes. However, we note that the throughput for this particular type of slice remains low between 1 and 2 Mbps, approximately, which limits the attackers' actions.

To further show the effects of a DDoS attack in our 5G platform, we plot in Fig. 3 the DDoS attack success rate metric (defined in the previous section), while varying the number of allocated PRBs for the sinkhole-type slice and fixing the number of attack bursts to 5. We can observe that this metric increases with the increase in the size of the sinkhole-type slice, since more allocated resources means that more *TCP SYN* packets will be forwarded successfully to the PGW during a DDoS attack. This result shows the importance of setting such a slice type with the smallest amount of physical resources as possible, in order to mitigate the malicious user's action.

In order to assess how the allocated resources of the benign slice are affected by the DDoS attacks, we plot in Fig. 4 the Tx/Rx throughput perceived by the benign user with the use of *iperf3*. As we can see, both uplink and downlink throughput are increased with the number of allocated PRBs, since more resources are allocated to the benign slice. In particular, the downlink throughput for this particular type of slice remains high between 17 and 22 Mbps approximately, which indicates that benign users are not severely affected by the DDoS attacks.

Finally, we assessed the processing time needed by the SDN *FlexRAN* controller to create a sinkhole-type slice, as well as to move a malicious user to it. Our results showed

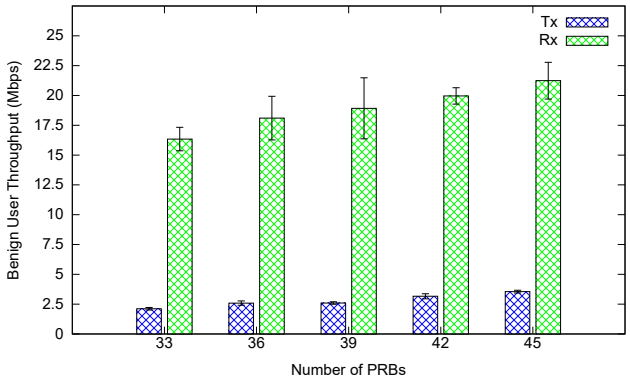

Fig. 4. Benign slice: user Tx/Rx throughput.

that the controller takes an average time of 5.6 milliseconds (with a 95% confidence interval ranging between 4.92 and 6.28 milliseconds) to create a sinkhole-type slice. We observed the same average time and variation to move a malicious user to such a slice. These results can be explained by the fact that such an operation does not rely on user's actions, but is done rather internally at the RAN part by the *FlexRAN* controller through simple control messages based on *Google Protobuf* [2].

VI. Conclusion

In this paper, we presented a new approach based on DL models to detect and mitigate DDoS attacks in 5G and beyond mobile networks. To evaluate and validate the effectiveness of our approach, we built a 5G prototype that allows us to manage users in network slices, according to the inputs from a northbound SDN application here so-called *Slicing APP*. The *Slicing APP* is integrated to the *FlexRAN* SDN controller which deals directly with the slicing management.

We compared our DL models in terms of prediction accuracy, false positive rate, execution time, among other relevant ML-related performance metrics. In addition, we assessed the performance of benign and sinkhole-type slices by measuring the throughput between users and the CN, as well as the attack success rate, in terms of the number of attack packets successfully sent to the CN. Obtained results show that our DL models are able to timely trigger the deployment of countermeasures, along with the flexibility of SDN and NFV for secure network slice reconfiguration. Regarding the detection performance, our DL models were able to achieve an accuracy of almost 97% and a false positive rate of less than 4%, through a test set built with data generated by our 5G prototype.

As future work, we intend to develop other ML and DL-based techniques that allow the release of a user from a sinkhole-type slice, once it is not classified anymore as malicious. We also intend to assess the performance of our techniques under a Vehicle-to-Everything (V2X) environment, and considering other types of attacks like jamming and false information injection.

References

[1] Eurecom, "OpenAirInterface." [Online]. Available: https://openairinterface.org/

[2] X. Foukas *et al.*, "FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks," in *Proceedings of the 12th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2016, p. 427–441.

[3] R. Schmidt, C.-Y. Chang, and N. Nikaein, "FlexVRAN: A Flexible Controller for Virtualized RAN Over Heterogeneous Deployments," in *IEEE International Conference on Communications (ICC)*, 2019.

[4] S. Costanzo, S. Cherrier, and R. Langar, "Network Slicing Orchestration of IoT-BeC^3 applications and eMBB services in C-RAN," in *Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 975–976.

[5] L. Bonati *et al.*, "SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems," in *Proc. of ACM Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, Virtual Conf., June 2021.

[6] N. Salhab *et al.*, "Autonomous Network Slicing Prototype Using Machine-Learning-Based Forecasting for Radio Resources," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 73–79, 2021.

[7] D. Bega *et al.*, "DeepCog: Optimizing Resource Provisioning in Network Slicing With AI-Based Capacity Forecasting," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 361–376, 2020.

[8] Y. Li, B. Liang, and A. Tizghadam, "Robust Online Learning against Malicious Manipulation and Feedback Delay With Application to Network Flow Classification," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2648–2663, 2021.

[9] T. N. Weerasinghe, I. A. M. Balapuwaduge, and F. Y. Li, "Supervised Learning based Arrival Prediction and Dynamic Preamble Allocation for Bursty Traffic," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019.

[10] T. A. Khan *et al.*, "Machine Learning Approach for Automatic Configuration and Management of 5G Platforms," in *20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019.

[11] L. Zhang *et al.*, "Learning to detect and mitigate cross-layer attacks in wireless networks: Framework and applications," in *Proc. of IEEE Conf. on Communications and Network Security (CNS)*, 2017, pp. 1–9.

[12] D. Sattar and A. Matrawy, "Towards Secure Slicing: Using Slice Isolation to Mitigate DDoS Attacks on 5G Core Network Slices," in *Proc. of IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 82–90.

[13] N. A. E. Kuadey *et al.*, "DeepSecure: Detection of Distributed Denial of Service Attacks on 5G Network Slicing—Deep Learning Approach," *IEEE Wireless Communications Letters*, vol. 11, no. 3, 2022.

[14] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." [Online]. Available: https://www.tensorflow.org/

[15] F. Chollet *et al.*, "Keras." [Online]. Available: https://github.com/fchollet/keras

[16] R. Doriguzzi-Corin *et al.*, "Lucid: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, 2020.

[17] D. Kennedy *et al.*, *Metasploit: The Penetration Tester's Guide*, 1st ed. USA: No Starch Press, 2011.

[18] Netsniff-ng, "Mausezahn." [Online]. Available: http://netsniff-ng.org/

[19] ESnet, "iperf3." [Online]. Available: https://downloads.es.net/pub/iperf/

[20] B. Bousalem *et al.*, "Deep Learning-based Approach for DDoS Attacks Detection and Mitigation in 5G and Beyond Mobile Networks," in *Proc. of IEEE International Conference on Network Softwarization (IEEE NetSoft), Demo Paper*, June 2022.

[21] B. Bousalem *et al.*, "Deep Learning-based Approach for DDoS Attacks Detection and Mitigation in 5G and Beyond Networks (Demo Video)." [Online]. Available: https://www.youtube.com/watch?v=YBx22va56N0