A Deep Transfer Learning-powered EDoS Detection Mechanism for 5G and Beyond Network Slicing

Chafika Benzaïd*, Tarik Taleb*, Ashkan Sami†, and Othmane Hireche§
* University of Oulu, † Edinburgh Napier University, § Univ.of Sciences and Technology Houari Boumediene
Emails: chafika.benzaid@oulu.fi, tarik.taleb@oulu.fi, a.sami@napier.ac.uk, othmane.hireche@nanoform.com

***Abstract*—Network slicing is recognized as a key enabler for 5G and beyond (B5G) services. However, its dynamic nature and the growing sophistication of DDoS attacks put it at risk of Economical Denial of Sustainability (EDoS) attack, causing economic losses to service provider due to the increased elastic use of resources. Motivated by the limitations of existing solutions, we propose FortisEDoS, a novel framework that aims at enabling EDoS-aware elastic B5G services. FortisEDoS integrates a new deep learning-based DDoS anomaly detection model, called CG-GRU, that leverages the capabilities of emerging graph and recurrent neural networks in capturing spatio-temporal correlations to accurately identify malicious behavior, allowing proactive mitigation of EDoS attacks. Moreover, FortisEDoS uses transfer learning to effectively counteract EDoS attacks in newly deployed slices by leveraging the knowledge acquired in previously deployed slice. The experimental results show the superiority of transfer learning-powered CG-GRU in achieving higher detection performance with lower computation overhead, compared to other baseline methods.**

I. Introduction

The joint use of network virtualization and softwarization is vital for realizing network slicing to enable next-generation mobile networks support diversified deployment scenarios, whereby multiple services can share the same substrate [1]. Each slice is designed with custom network capabilities to meet the performance needs of a specific service type [2].

A key life cycle management operation of network slices is auto-scaling, which dynamically expands or contracts the capacity of a network slice instance to accommodate resources to slice workload to fulfill the performance desire. However, the auto-scaling capability is a double-edged sword when a (Distributed) Denial of Service – (D)DoS – attack is underway. Indeed, the auto-scaling capability can reshape an undetected (D)DoS attack into an Economical Denial of Sustainability (EDoS) attack, which results in economic damages to service provider owing to the increased elastic usage of resources [3]. The undesirable economic impact of EDoS is a critical issue as it may spread beyond the slice under attack, affecting slices co-hosted on the same substrate [4]. Hence, providing reliable dynamic resource provisioning that is EDoS attack aware is paramount to enable profitable beyond 5G services.

Realizing the aforementioned goal is challenging due to the growing trend towards more stealthier DDoS attacks that are aiming at the application layer rather than the network layer. Such trend is mainly due to the capacity of those attacks to mimic legitimate behavior with low network bandwidth usage. Despite the extensive work that has been undertaken to counteract DDoS attacks, the stealthy application-layer DDoS (AL-DDoS) issue is far from being addressed, and even less in 5G and beyond network slicing environment. Existing solutions have a number of limitations, which hampers their effectiveness and efficiency. The total isolation among slices promoted by resource isolation based approaches (e.g., [5]) may result in over-provisioning of resources or may not be possible to achieve due to lack of strong hardware isolation in the emerging cloud-native platforms [6]. By imitating legitimate traffic, AL-DDoS attacks can escape detection by network traffic analysis based solutions (e.g., [7], [8]) Using resource scaling as a mitigation strategy by resource allocation-based methods (e.g., [9]) grows the risk of revamping an undetected AL-DDoS attack into an EDoS attack [10]. The recent anomaly detection approaches (e.g., [6], [11], [12]) that leverage the potential of Deep Learning (DL) to recognize abnormal behavior based on anomalies detected in service performance and/or resource usage metrics are a promising direction to tackle EDoS attack. However, existing anomaly detection methods only consider temporal correlations between metrics and/or assume that sufficient amount of historical data is available for training the DL models to learn normal behavior.

Driven by aforementioned limitations of DL-based anomaly detection approaches and the limited research work tackling the EDoS issue in 5G network slicing, we propose FortisEDoS, a novel framework that incorporates a deep transfer learning model to enable highly elastic 5G and beyond services that can deliver the desired quality of experience while being impervious to EDoS attacks. FortisEDoS leverages the capabilities of emerging DL techniques, particularly convolutional neural networks (CNN), recurrent neural networks and graph neural networks, to capture both temporal and spatial dependencies among resource usage and service performance metrics and adopt a dynamic thresholding strategy to accurately recognize anomalous status of a slice's virtual network function (VNF) under AL-DDoS attack. Moreover, FortisEDoS exploits the concept of transfer learning to empower effective identification of anomalous VNF's status even when representative historical data of normal behavior are scare. To the best of our knowledge, this is the first contribution of deep transfer learning in tackling EDoS attacks against network slicing.

The remainder of the paper is organized as follows. Section II introduces the proposed FortisEDoS framework, delineating its architecture and the design of the deep transfer learning-based DDoS anomaly detection model. Section III describes the experimental setup and provides a comprehensive analysis of the performance results. Finally, Section IV concludes the paper and highlights future research directions.

II. FortisEDoS Elastic Mobile vCDN Framework

A. Framework Overview

In the following, we introduce FortisEDoS, a novel framework that aims to enable elastic 5G network slicing while intelligently preventing malicious resource scaling requests generated by AL-DDoS attacks. As depicted in Fig. 1, we consider a vCDN provided as a service over a MEC-enabled 5G networks to deliver video content. A vCDN service is dynamically deployed on-demand as a slice into the MNO's network. Each slice consists of a set of basic VNFs (e.g., streamers, caches) chained together to provide a vCDN service. The vCDN slices can share 5G core network (CN) functions (e.g., AMF and SMF) or have their dedicated 5G CN functions (e.g., UPF and I-UPF). The VNFs of a vCDN slice can be deployed over several edge compute nodes and the VNFs of different vCDN slices can be co-hosted on the same edge compute node.

The FortisEDoS framework includes a *vCDN Management Layer* that comprises a set of modules providing required functionalities to enable EDoS-aware elastic vCDN services. Specifically, it incorporates the following core components:

- *Monitoring System* is continuously tracking, via the deployed monitoring agents, data related to resource usage (e.g., CPU, RAM) and performance (e.g., response time) metrics of the different vCDN slice's VNFs and their hosting nodes. The collected data are used to drive the resource scaling and anomaly detection decisions made by the auto-scaling module and DDoS Mitigator, respectively.
- *Auto-scaling Module* dynamically expands or shrinks the capacity of a vCDN slice instance to adjust resources to slice workload in order to fulfill the agreed Service Level Agreement (SLA). The scaling decision happens at the VNF level based on its performance and resource usage metrics according to the associated auto-scaling policies. It is worth mentioning that a VNF can either be scaled horizontally by increasing/decreasing the number of VNF instances or vertically by increasing/decreasing the resources used by a VNF instance.
- *Admission Controller* is in charge of intercepting the scaling-up/out requests triggered by the Auto-scaling Module in order to entrust the scaling decision to the DDoS Mitigator for validation.
- *DDoS Mitigator* exploits the potential of DL to automatically discriminate malicious scaling requests engendered by AL-DDoS attacks from those caused by legitimate load. It includes a DL-based anomaly detection model which can effectively identify anomalous VNFs' metrics using a data-driven forecasting-based approach. Indeed, the anomalies are detected when the predicted metrics' values drift considerably from the observed ones. If an anomaly is identified, the scaling operation is flagged as malicious and will be refused by the Admission Controller. Details on the proposed DL model and the selection of the anomaly threshold will be provided in the subsequent sections.

B. Attacker Model

We assume that the attacker has control over a subset of user devices that can legitimately use a 5G vCDN service delivered via HTTP-based technologies. The attacker aims at depleting the vCDN slice's resources (e.g., CPU, RAM) to inhibit legitimate users from accessing the vCDN service or at the very least increase the service response time. To this end, we suppose that the attacker can conduct AL-DDoS attacks against the vCDN's VNFs exposed to end user, such as the video streamer. Specifically, the attacker is able to carry out both high-rate and low-rate HTTP-based DDoS attacks. In high-rate mode, the attacker floods the exposed service with a large number of legitimately formed HTTP requests. In the low-rate mode, the attacker establishes multiple HTTP connections with the exposed service by sending partial HTTP requests at a very slow rate, which leads to exhausting the connection queue space.

We further assume that the attacker can generate stealthier DDoS traffic patterns that can escape detection by network traffic analysis based mechanisms [8]. Thus, the malicious traffic will reach the exposed VNF and results in requesting additional resources through auto-scaling, allowing to reshape the AL-DDoS attack into an EDoS attack.

C. Problem Formulation and Methodology

We consider a set of n slices $\mathcal{S} = \{S_1, \cdots, S_n\}$, where each slice S_i is composed of a set of m VNFs $\mathcal{V}_i = \{f_1^i, \cdots, f_m^i\}$. Each VNF $f_j^i \in \mathcal{V}_i$ is defined by a set of features $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$ representing its resource usage and performance metrics. $\boldsymbol{d}$ refers to the dimension of the VNF's features set.

The VNF's metrics collected at regular intervals over a period of time can be formulated as a multivariate time series $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\} \in \mathbb{R}^{T \times d}$, where T is the length of the time series. Each step $\boldsymbol{x}^{(t)} \in \mathbb{R}^d$ in the time series is a d-dimensional vector $\{\boldsymbol{x}_1^{(t)}, \boldsymbol{x}_2^{(t)}, \cdots, \boldsymbol{x}_d^{(t)}\}$ representing the VNF's metrics data observed at time t.

We aim to detect the AL-DDoS attack by determining anomalies in VNF's metrics using a forecasting-based approach, where an anomalous VNF's status is detected when the expected metrics values diverge greatly from the measured ones. As each metric may not only depend on its own historical values, but also on other metrics' past, we embrace a multivariate time series forecasting approach in order to improve the anomaly detection accuracy. Given the observed metrics values of previous w time steps $\boldsymbol{x}^{(t-w+1)}, \cdots, \boldsymbol{x}^{(t)}$, the multivariate time series forecasting task intends to learn a model $F : \mathbb{R}^{w \times d} \mapsto \mathbb{R}^{h \times d}$ for predicting the future metrics values for the next h time steps, denoted by $\hat{\boldsymbol{x}}^{(t+1)}, \cdots, \hat{\boldsymbol{x}}^{(t+h)}$. It can be formally written as

$$[\hat{\boldsymbol{x}}^{(t+h)}, \cdots, \hat{\boldsymbol{x}}^{(t+1)}] = F(\boldsymbol{x}^{(t)}, \cdots, \boldsymbol{x}^{(t-w+1)}) \qquad (1)$$

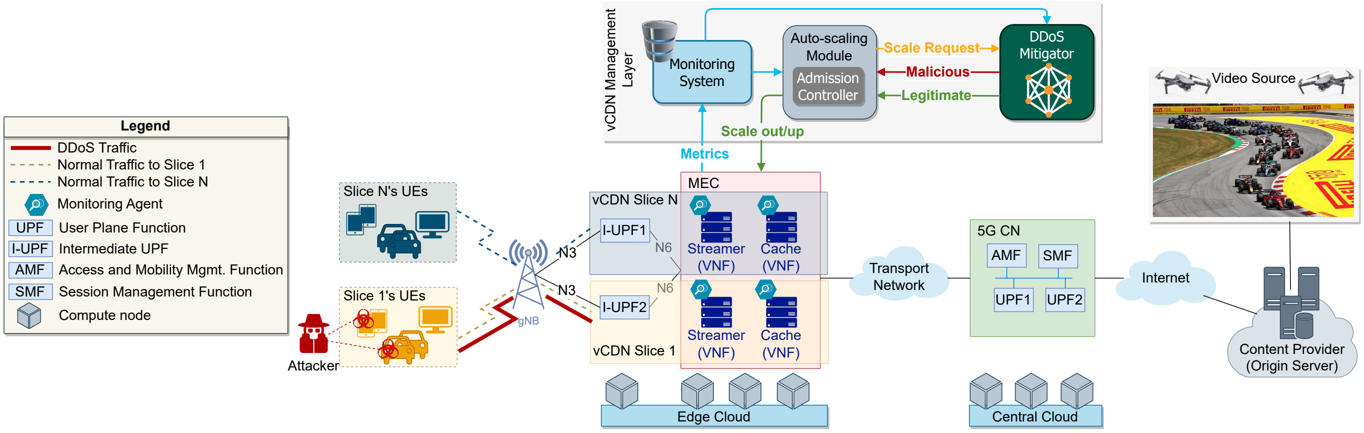

Fig. 1: The overall architecture of FortisEDoS Elastic 5G vCDN Framework.

The forecasting model is trained to successfully predict future metrics values from normal values by minimizing the prediction error. Thus, during the inference, the prediction error is anticipated to rise in the presence of anomalous metrics values due to DDoS attack. Leaning on this hypothesis, we use the prediction error to measure the anomaly score, which represents the deviation of true metrics values from the predicted ones. If the derived anomaly score is above a detection threshold, the VNF status is tagged as anomalous.

D. Data Preprocessing

The data pre-processing module prepares the raw time series data into the appropriate format to fit for the forecasting model. The train dataset includes only data of normal behavior.

Firstly, the raw time series data are cleaned by imputing missing/infinity values. They are then normalized using the Min-Max scaling technique, which scales the values in each time series to be in the range $[0, 1]$. The data normalization helps in improving the model stability and training time by removing scaling differences between metrics. Finally, the normalized data are segmented into a series of sub-sequences by applying a sliding window technique. As shown in Fig. 2, the training dataset is constructed as a supervised dataset, where the inputs are the observed metrics values of previous w time steps and the outputs are the future values to forecast for the next h $(= 1$ in Fig 2) time steps.

E. Forecasting Model Architecture

The upper part of Fig. 2 elucidates the overall architecture of CG-GRU model. It is an hybrid model that consolidates the potential of different DL algorithms to provide both feature extraction and forecasting capabilities. In fact, DL techniques have the potential of uncovering complex patterns from a large-amount of data, delivering accurate decisions [13].

The feature extraction stage consists in capturing both temporal and spatial dependencies within the multivariate time series using three types of neural network layers:

- Leveraging the high capability of CNN in extracting high-level representations from data, the local relevant features within a sliding window are extracted from the pre-processed multivariate time series using a one-dimensional Convolutional (Conv1D) layer. The local features are derived by first convolving the input data with a learned convolution kernel and then applying a non-linear activation function.
- The resulting features are then fed into a Graph Attention (GAT) layer to extract the spatial correlations between the VNF's metrics. Thanks to the attention mechanism of GAT, different weights are assigned to each pair of features, allowing to measure the degree of influence of VNF's metrics on each other. It is worth noting that unlike previous graph-based methods, GATs have the advantage of capturing the importance levels, not requiring prior knowledge of the global graph structure, being storage and computationally efficient, and providing the interpretability of the model [14].
- The features extracted by the GAT layer are processed by multiple Gated Recurrent Unit (GRU) layers to derive the temporal dynamics of the VNF's metrics. Each GRU layer contains several hidden units to update the hidden state, each of which consists of two gates, called *reset gate* and *update gate*. The reset gate forgets irrelevant past information, while the update gate retains relevant information from the previous time step.The use of GRU is motivated by their demonstrated effectiveness and efficiency in modeling long-term temporal sequences thanks to their capability to remember useful past observations while reducing computation [15].

The forecasting stage takes the spatio-temporal representations learned by the feature extraction block as inputs to predict the future VNF's metrics values. It hinges on a Multi-Layer Perceptron (MLP) network consisting of multiple fully-connected layers.

F. Forecast-based DDoS Anomaly Detection

To set an appropriate threshold for detecting malicious VNF scaling requests, we adopt a dynamic thresholding methodology [12]. This method allows to calculate an anomaly detection threshold that is automatically adjusted according

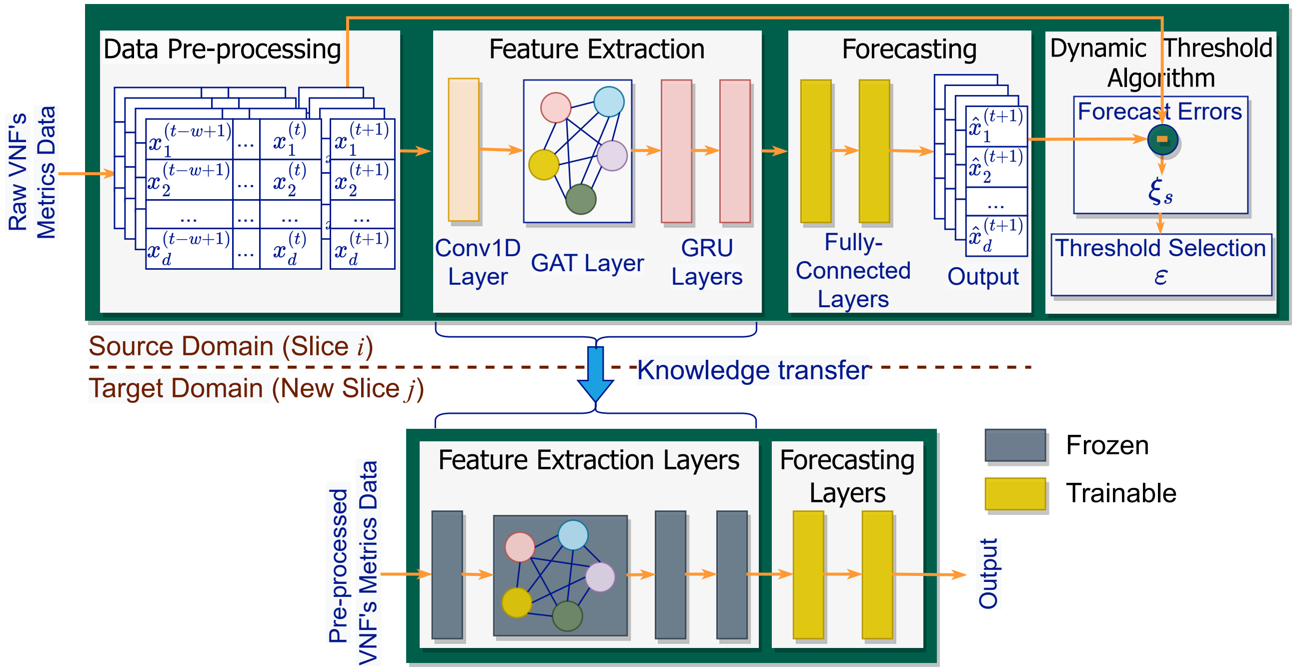

Fig. 2: The overall architecture of training the TL-powered CG-GRU model and selecting the dynamic anomaly threshold.

to the past smoothed forecasting errors. The key advantage of this method is its leaning on a non-parametric probability distribution estimation approach, which overcomes the limitations of traditional Gaussian assumptions on the past smoothed forecasting error distribution.

The forecasting error of the i-th VNF's metric at time step t, $e_i^{(t)}$, is calculated by

$$e_i^{(t)} = |x_i^{(t+1)} - \hat{x}_i^{(t+1)}| \tag{2}$$

where $x_i^{(t+1)}$ and $\hat{x}_i^{(t+1)}$ are, respectively, the actual value and forecast value of the i-th VNF's metric at time step t.

Using the metric-specific forecasting errors, the global forecasting error of the VNF at time step t is computed as $e^{(t)} = \frac{1}{d}\sum_{i=1}^{d} e_i^{(t)}$.

The dynamic threshold is derived using the smoothed global forecasting errors at time step t, $\xi_s = [e_s^{(t-w)}, \cdots, e_s^{(t-1)}, e_s^{(t)}]$, where w is the historical window size. The exponentially weighted moving average (EWMA) algorithm is applied to smooth the global forecasting errors, enabling reduced false positives. The threshold ε is picked from the set $\boldsymbol{\epsilon} = \mu(\xi_s) + \beta * \delta(\xi_s)$ such that:

$$\varepsilon = \operatorname{argmax}(\boldsymbol{\epsilon}) = \frac{\Delta\mu\left(\xi_s\right)/\mu\left(\xi_s\right) + \Delta\delta\left(\xi_s\right)/\delta\left(\xi_s\right)}{|\boldsymbol{\xi}_a| + |\mathbf{E}_{seq}|^2} \tag{3}$$

where

$$\begin{aligned}
\Delta\mu\left(\xi_s\right) &= \mu\left(\xi_s\right) - \mu\left(\{e_s \in \xi_s \mid e_s < \varepsilon\}\right)\\
\Delta\delta\left(\xi_s\right) &= \delta\left(\xi_s\right) - \delta\left(\{e_s \in \xi_s \mid e_s < \varepsilon\}\right)\\
\boldsymbol{\xi}_a &= \{e_s \in \xi_s \mid e_s > \varepsilon\}\\
\mathbf{E}_{seq} &= \text{continuous sequences of } \xi_a \in \boldsymbol{\xi}_a
\end{aligned}$$

Note that $\Delta\mu\left(\xi_s\right)$ and $\Delta\delta\left(\xi_s\right)$ represent the decrease in the mean and the standard deviation of the global forecasting errors, respectively. $\boldsymbol{\xi}_a$ denotes all the global forecasting errors that are above the dynamic threshold. The parameter β is selected from an ordered set B of positive values representing the standard deviations above $\mu\left(\xi_s\right)$.

The values of the VNF's metrics at a time step t are considered anomalous if the corresponding smoothed global forecasting error $e_s^{(t)}$ exceeds the calculated threshold.

G. Transfer Learning empowered DDoS Anomaly Detection

A newly instantiated vCDN slice will possibly lack representative training data that incorporate all variations of their VNFs' normal behavior, which may result in performance degradation of DDoS anomaly detection. Furthermore, collecting such representative data to build the forecasting-based model is time and resource consuming. Thus, it is vital to reduce the (re)training time and cost to enable prompt detection of attacks and guarantee service profitability, particularly when a massive number of slices is deployed.

To deal with the above-mentioned issues, we exploit the potential of Transfer Learning (TL) to leverage knowledge gained by a model in previously instantiated slice (denoted as *source domain*) for enhancing and speeding up the learning of a model in a newly deployed slice (denoted as *target domain*). Specifically, we consider transferring the knowledge about feature representations of a normal behavior derived by the model of the source domain to the new model of the target domain. Indeed, DL networks allow transferability of general features across domains, thanks to their capability to learn general features (i.e., domain-independent) on the first layers and specific features (i.e., domain-dependent) on the layers closer to the output [16]. Driven by that, we realize the TL by setting the weights of the feature extraction layers of the new CG-GRU model to ones inherited from the pre-trained CG-GRU model. New fully-connected layers are added and fine-tuned on the target data to customize the model for the associated VNF. Note that the weights of the feature extraction layers are frozen during the tuning phase to preserve the transferred knowledge. Fig. 2 illustrates the proposed TL process.

III. Performance Evaluation

A. Experimental Setup

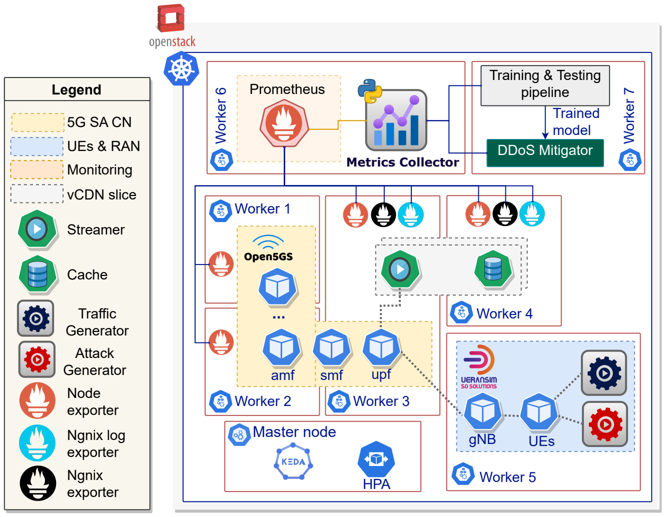

Fig. 3: The overall testbed architecture.

To assess the performance of FortisEDoS, we built a testbed based on a Kubernetes (K8s) cluster set up on an OpenStack cloud. As illustrated in Fig. 3, the K8s cluster comprises one master node and seven worker nodes. Open5GS (https://github.com/Gradiant/openverso-charts) and UERANSIM (https://github.com/aligungr/UERANSIM) are used to implement the 5G SA Core and simulate the 5G RAN and User Equipments (UEs), respectively. It is worth noting that Open5GS supports network slicing by allowing the configuration of various slice parameters, including slice type, assigned bandwidth, and quality of service class. The simulated UEs are used to generate normal or malicious load. The vCDN slices consist of two Cloud-native Network Functions (CNFs), namely a video streamer and a cache, chained together to provide an HTTP-based on-demand video streaming service. The two CNFs are deployed as K8s services running a NGINX web server and are spread over two worker nodes. The video streamer service is exposed to the end user, which can access the service via the 5G network. Each vCDN slice instance has its own namespace to guarantee isolation of API resources between slices. The Monitoring System uses Prometheus API and different probes to extract the raw resource usage and performance metrics from the vCDN slices' CNFs and their hosting nodes. The DDoS Mitigator is deployed on a separate worker node, which serves also as a platform for training and testing the CG-GRU model implemented using Pytorch library.

The Auto-scaling module is implemented using the K8s Horizontal Pod Autoscaling (HPA) functionality which we extended with the event-driven scale feature provided by the KEDA tool (https://keda.sh). We defined various scaling policies to handle the load on a CNF based on per-pod metrics (e.g., CPU) or external metrics obtained from Prometheus (e.g., response time).

B. Dataset Generation & Model Training

Due to the lack of real data, we used our testbed to generate realistic datasets to train and test CG-GRU. To this end, we developed a normal load generator that models the arrival times of legitimate video streaming requests according to Poisson process with fixed hourly rate. The normal load generator includes a python script that controls the Selenium WebDriver (https://www.selenium.dev) for automating the loading and playback of the requested videos in a web browser. The "impatient user" behavior is approximated by randomly varying the duration of the video streaming sessions. To generate malicious load, we implemented the attack agents using Slowloris tool (https://github.com/gkbrk/slowloris) for low-rate DDoS attacks and Hulk tool (https://github.com/grafov/hulk) for high-rate DDoS attacks. The DDoS attacks are carried out against the video streamer service.

The raw resource usage and performance metrics data were recorded from two vCDN slices over a period of 5 days. A time series for each metric was recorded using a data sampling period of 60s. The training data were collected during the first 4 days of attack-free activity. The 5th day served to create the testing dataset where AL-DDoS attacks were executed on different periods of the day. Specifically, three Hulk attacks with different intensities and one Slowloris attack were launched. Training and testing datasets were generated for each vCDN slice's CNF with a total of 5401 and 2701 samples, respectively. During training, 20% of samples in the training dataset are used for validation.

As the performance of CG-GRU model is sensitive to hyper-parameter settings, we leveraged grid searchand ASHA [17] to find the best model's configuration that minimized the forecast error on the validation set. Each possible configuration is trained at most 100 epochs using Rectified Linear Unit (ReLU) as the activation function, Adam as the optimizer and Mean Squared Error (MSE) as loss function. The hyper-parameters setting of the best model, achieving a forecasting loss of 3.61%, uses 2 GRU layers with 90 neurons per layer, 2 dense layers with 60 neurons per layer, a kernel size of 3, a historical window size of 80, a learning rate of 0.001, a dropout rate of 0.2, and a batch size of 150.

A more detailed description of the implemented normal load generator and the hyperparameter tuning process can be found in [18]. These details have been omitted here due to page-limit.

C. Performance Metrics

The effectiveness of FortisEDoS in preventing fraudulent resource scaling requests is assessed by measuring the performance of CG-GRU in detecting AL-DDoS attacks over the testing dataset using Precision, Recall (a.k.a. sensitivity) and F1-score metrics, which are respectively calculated as $\frac{TP}{TP+FP}$, $\frac{TP}{TP+FN}$, and $2 \times \frac{Recall \times Precision}{Recall+Precision}$. Note that TP (True Positive) represents the number of correctly detected anomalies FN (False Negative) denotes the number of anomalies that are falsely detected as normal samples, FP (Flase Positive) is the number of the normal samples that are wrongly flagged as anomalous ones, and TN (True Negative) refers to the number of the normal samples that are correctly detected.

Besides its effectiveness, we evaluate the efficiency of FortisEDoS in terms of the economic damage repair (EDR), which is measured by the difference in extra CNF replicas induced by AL-DDoS attack with and without DDoS Mitigator.

D. Performance Results

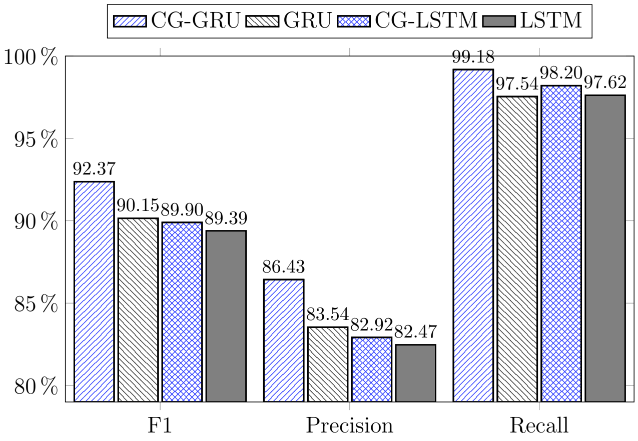

Fig. 4: Attack Detection Performances.

1) Attack Detection Performances: We compare CG-GRU with the LSTM-based anomaly detection model proposed in [12]. Moreover, we conduct a layer ablation study to assess the impact of the local features extracted by Conv1D layer and the spatial features derived by GAT layers on the DDoS attack detection. To this end, we compare CG-GRU with GRU, where only the temporal information extracted by GRU layers is used. We also transplant the Conv1D and GAT layers to LSTM model proposed in [12] to assess their impact on the model's performances.

The results depicted in Fig. 4 demonstrate the superiority of CG-GRU model in achieving the highest performance scores compared to all other models. In fact, CG-GRU model exhibits a high sensitivity in identifying anomalous CNF's status while yielding an acceptable Precision of 86.43% and a F1 score of 92.37%. It is worth noting that in our case a high Recall is preferred over a high Precision, as the unsuccessful detection of anomalous CNF's status may lead to economic losses due to accepting malicious resource scaling operations. Compared with the LSTM-based model, CG-GRU improves the Precision, Recall and F1 scores by at least 3.96%, 1.56% and 2.98%, respectively. This improvement is attributed to the quality of the learned spatio-temporal features, which allows better estimation of the anomaly threshold. This statement is corroborated by the results of the ablation study, which demonstrate the importance of capturing both spatial and temporal dependencies from convolved data. The results reported in Fig. 4 show that adding Conv1D and GAT layers allows CG-GRU model to outperforms the baseline GRU model, increasing the Precision, Recall and F1 scores by at least 2.89%, 1.64% and 2.22%, respectively.

2) Effectiveness of TL: To test the effectiveness of applying TL in terms of both attack detection and training time, we transfer the CG-GRU model trained on data collected from the video streamer CNF of vCDN slice 1 (denoted *vStreamer1*) to a newly deployed video streamer CNF of vCDN slice 2 (denoted *vStreamer2*). Unlike vStreamer1, only few interactions have been performed between the simulated legitimate users and vStreamer2 and with access to the same video file. Hence, the training dataset collected from vStreamer2 is not representative of a normal behavior.

Let TL-CG-GRU denote the CG-GRU model for vStreamer2 built using TL. For comparison, we train another CG-GRU model for vStreamer2 from scratch using its training dataset (hereafter denoted as CG-GRU-vS2).

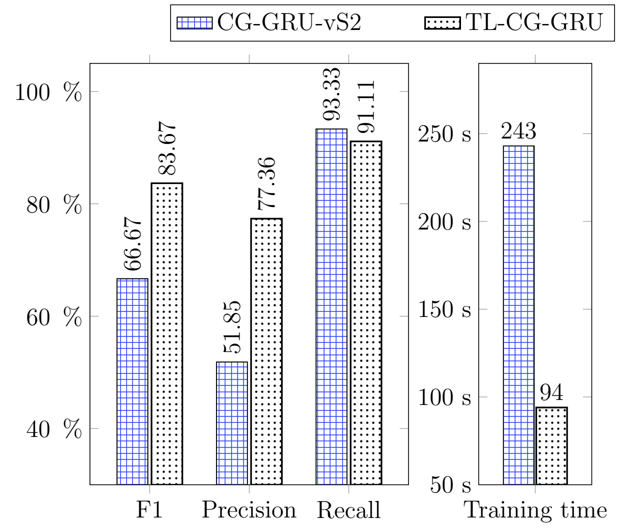

Fig. 5: The performances and training time of CG-GRU model with and without transfer learning.

Fig. 5 reports the training time as well as the attack detection performance indicators over Streamer2's testing dataset. The results demonstrate the superiority of the transferred model TL-CG-GRU in boosting the detection performance while considerably reducing the training overhead. Indeed, TL-CG-GRU results in a gain of 25.51% and 17% in precision and F1-score, respectively, compared to CG-GRU-vS2. This supports our idea that the spatio-temporal features derived by the feature extraction layers are more generic and therefore can be transferred among CNFs of different slices. Moreover, TL-CG-GRU substantially speeds up the training of CG-GRU-vS2 model by at least 61%, thanks to the reuse of knowledge regarding feature representations during the fine-tuning phase.

3) Economic Damage: Fig. 6 depicts the number of extra replicas induced by Hulk and Slowloris attacks on vCDN

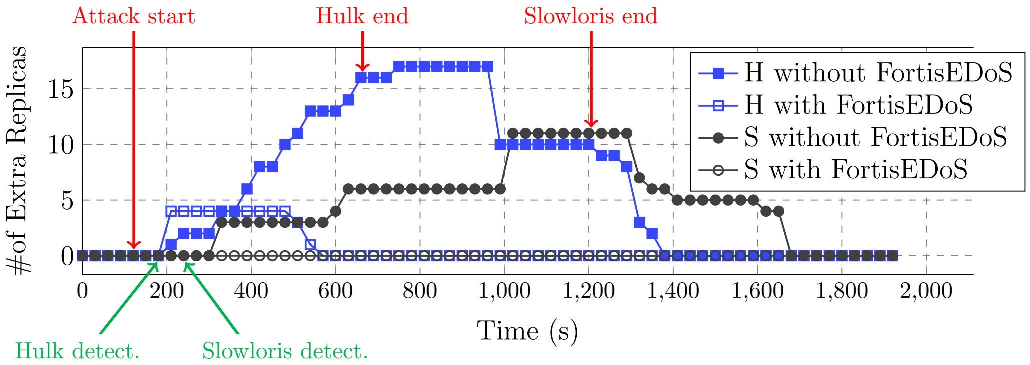

Fig. 6: Economic damage caused by Hulk (H) and Slowloris (S) attacks on vCDN slice1, with and without FortisEDoS.

slice 1, with and without using FortisEDoS. Note that without FortisEDoS, the attack is stopped when the maximum replica count of vStreamer CNF is reached (set to 15 in our experiment). The results show that FortisEDoS can reduce the economic damage, computed as the area under the curve, from 12180s to 1320s under Hulk attack, achieving 89.16% EDR. In the case of Slowloris, the EDR is 100% thanks to the early detection of this attack.

IV. Conclusion

In this paper, we proposed FortisEDoS, a novel framework for enabling highly elastic B5G services while being immune to EDoS attacks. FortisEDoS achieves its goal by (i) integrating CG-GRU, a new DL-powered DDoS anomaly detection model which exploits the forecasting errors between the observed VNF's metrics and the predicted ones to determine malicious VNF scaling requests due to stealthy AL-DDoS attacks; and (ii) adopting the concept of transfer learning to yield effective detection of EDoS attack in newly deployed slices. The experimental results showed the superior performance of FortisEDoS in accurately detecting EDoS attack, reducing related economic damage, and confirmed the benefit of transfer learning in boosting both attack detection effectiveness and training speed when representative historical data of normal behavior are scare. In the future, we intend to devise an advanced mechanism for selecting the appropriate VNFs/slices for knowledge transfer.

Acknowledgment

This research work is partially supported by the Business Finland 6Bridge 6Core project under Grant No. 8410/31/2022, the Research Council of Finland (former Academy of Finland) under IDEA-MILL project (Grant No. 352428) and 6G Flagship program (Grant No. 346208), the EU's Horizon 2020 research and innovation programme under the CHARITY project (Grant No. 101016509), and the European Union's Horizon Europe research and innovation programme HORIZON-JU-SNS-2022 under the RIGOUROUS project (Grant No. 101095933) and the 6G-SANDBOX project (Grant No. 101096328). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains.

References

[1] C. Benzaid, T. Taleb, and M. Z. Farooqi, "Trust in 5G and Beyond Networks," *IEEE Network*, vol. 35, no. 3, pp. 212 – 222, May 2021.

[2] A. Javadpour, F. Ja'fari, T. Taleb, and C. Benzaid, "Reinforcement Learning-based Slice Isolation Against DDoS Attacks in Beyond 5G Networks," *IEEE Trans. Net. Service Manag.*, pp. 1–1, 2023.

[3] A. Bremler-Barr, E. Brosh, and M. Sides, "DDoS Attack on Cloud Auto-Scaling Mechanisms," in *Proc. of the IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1 – 9.

[4] NGMN, "5G Security Recommendations Package #2: Network Slicing," Apr. 2016.

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

[6] C. Benzaid, T. Taleb, and J. Song, "AI-based Autonomic & Scalable Security Management Architecture for Secure Network Slicing in B5G," *IEEE Network (Early Access)*, pp. 1 – 9, 2022.

[7] A. Thantharate et al., "Secure5G: A Deep Learning Framework Towards a Secure Network Slicing in 5G and Beyond," in *In Proc. of the 10th Annu. Comput. and Commun. Workshop and Conf. (CCWC)*, Jan. 2020, pp. 0852 – 0857.

[8] C. Benzaid, M. Boukhalfa, and T. Taleb, "Robust Self-Protection Against Application-Layer (D)DoS Attacks in SDN Environment," in *Proc. of IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020.

[9] Z. Li et al., "Exploring New Opportunities to Defeat Low-Rate DDoS Attack in Container-Based Cloud Environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 695 – 706, March 2020.

[10] R. S. Silva et al., "REPEL: A Strategic Approach for Defending 5G Control Plane From DDoS Signalling Attacks," *IEEE Trans. Net. Service Manag.*, vol. 18, no. 3, pp. 3231 – 3243, Sept.March 2021.

[11] P. T. Dinh and M. Park, "R-EDoS: Robust Economic Denial of Sustainability Detection in an SDN-Based Cloud Through Stochastic Recurrent Neural Network," *IEEE Access*, vol. 9, pp. 35 057 – 35 074, Feb. 2021.

[12] J. Yu et al., "Telemetry Data-Based Spacecraft Anomaly Detection With Spatial–Temporal Generative Adversarial Networks," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1 – 9, Apr. 2021.

[13] C. Benzaid and T. Taleb, "AI for Beyond 5G Networks: A Cyber-Security Defense or Offense Enabler?" *IEEE Network*, Sept. 2020.

[14] P. Veličković et al., "Graph attention networks," *CoRR*, vol. abs/1710.10903, 2017.

[15] K. Cho et al., "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," in *Proc. of the 8th Workshop on Syntax, Semantics and Struct. in Statistical Translation*, Oct. 2014, pp. 103–111.

[16] C. T. Nguyen et al., "Transfer Learning for Wireless Networks: A Comprehensive Survey," *Proc. IEEE*, pp. 1 – 43, 2022.

[17] L. Li et al., "A System of Massively Parallel Hyperparameter Tuning," in *In Proc. of the 3rd Mach. Learn. Syst. Conf. (MLSys)*, March 2020.

[18] C. B. et al., "FortisEDoS: A Deep Transfer Learning-empowered Economical Denial of Sustainability Detection Framework for Cloud-Native Network Slicing," *IEEE Trans. Dependable Secure Comput. (To appear)*.