5G-Fuzz: An Attack Generator for Fuzzing 5GC, using Generative Adversarial Networks

George Nakas[†], Panagiotis Radoglou-Grammatikis[†‖], George Amponis[†], Thomas Lagkas[‡], Vasileios Argyriou[§],
Sotirios Goudos[¶] and Panagiotis Sarigiannidis[‖]

***Abstract*—The evolution of fifth-generation (5G) networks represents a significant technological leap towards a seamless and advanced user experience, allowing hyperconnected use cases with faster data transfer, lower latency and better connectivity for a wide range of mobile devices. In particular, a key element of 5G is the 5G Core (5GC) which follows a service-based architecture, enabling network slicing and improved Quality of Service (QoS). However, despite the benefits of 5GC, it also creates important security and privacy concerns. First, 5GC can combine heterogeneous technologies that can increase the growing attack surface. On the other hand, 5G handles a vast amount of sensitive data that may reflect an attractive goal for potential cyberattackers. Based on the previous remarks, in this paper, we introduce `5G-Fuzz`. `5G-Fuzz` is a smart fuzzer which takes full advantage of historical data in order to fuzz and target the Packet Forwarding Control Protocol (PFCP) communications between the Session Management Function (SMF) and User Plane Function (UPF). For this purpose, two PFCP attacks are used. In contrast to conventional fuzzers, `5G-Fuzz` adopts two Generative Adversarial Networks (GANs) in order to identify and generate the appropriate values of Session Endpoint Identifier (SEID) and sequence number (seq) utilised in the PFCP sessions, thus accelerating the PFCP attacks. Finally, `5G-Fuzz` composes and replays the malicious PFCP packets against UPF.**

***Index Terms*—5G, Fuzzing, Generative Adversarial Networks, PFCP**

I. Introduction

Beginning in the early 2010s when the International Telecommunication Union (ITU) announced the need for next-generation wireless networks, the evolution of fifth-generation (5G) networks has provided several benefits, such as faster speeds, increased capacity and improved reliability [1]. A key architectural element of 5G is the 5G Core (5GC), which adopts an efficient service-based architecture, allowing the harmonic collaboration, modularisation and decoupling of several Network Functions (NFs), such as the Session Management Function (SMF), User Plane Function (UPF) and the Access and Mobility Management Function (AMF). In particular, the next-generation base station (also known as gNodeB or gNB) communicates first with 5GC in order to establish the necessary connections for the User Equipment (UE), including various functions, such as authentication, session management and policy enforcement. On the other hand, 5GC communicates with gNB in order to provide the necessary information and instructions regarding radio resource allocation, handovers, network coverage and, in general, the effective use of the network resources. The service-based architecture of 5GC provides several beneficial services, such as network slicing and improved Quality of Service (QoS). For this purpose, 5GC combines a variety of technologies, such as Network Function Virtualization (NFV), Software-Defined Networking (SDN) and cloud/edge computing. However, this heterogeneity raises crucial security and privacy concerns.

More specifically, despite the advantages of 5GC, its distributed and complicated architecture creates an increased attack surface, where cyberattackers may exploit potential security weaknesses and take control of NFs, thus compromising the essential security principles (i.e., confidentiality, integrity and availability) [2]. Moreover, the involvement of efficient technologies, such as NFV and SDN, creates new cybersecurity risks. Furthermore, 5GC handles a huge amount of data that may include sensitive information, thus raising privacy concerns about data collection, retention and profiling. Both academia and industry investigate the 5GC security issues. For instance, based on MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), FiGHT is a common knowledge base of tactics, techniques and sub-techniques for 5G environments. The techniques can be categorised into three main classes: (a) observed, (b) Proof of Concept (PoC) and (c) theoretical. Therefore, in light of the aforementioned remarks, it is evident that the development of appropriate security services is necessary. In this context, a fuzzer is a security mechanism which can send random or malformed input data to a target system in order to cause unexpected behaviour and discover potential security flaws. Conventional fuzzing techniques generate random or brute-force data in order to test the target system. On the other hand, with the advent

[*]*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 952672 (SANCUS). Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.*

[†]G. Nakas, P. Radoglou-Grammatikis and G. Amponis are with K3Y Ltd, William Gladstone 31, 1000 Sofia, Bulgaria - E-Mail: { gnakas, gamponis pradoglou}@k3y.bg

[‡] T. Lagkas is with the Department of Computer Science, International Hellenic University, Kavala Campus, Kavala, 65404, Greece - E-Mail: tlagkas@cs.ihu.gr

[§] V. Argyriou is with the Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, UK - E-Mail: vasileios.argyriou@kingston.ac.uk

[¶] S. Goudos is with School of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece - E-Mail: sgoudo@physics.auth.gr

[‖] P. Radoglou-Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, Kozani, 50100, Greece - E-Mail: {pradoglou,psarigiannidis@uowm}.gr

of Artificial Intelligence (AI), a smart fuzzer can follow a more targeted and intelligent approach in order to generate and execute more meaningful and effective actions.

With reference to the aforementioned remarks, this parer presents a smart fuzzer called 5G-FUZZ, which targets the Packet Forwarding Control Protocol (PFCP) communications between SMF and UPF in 5GC. In particular, the proposed fuzzer takes full advantage of existing PFCP-related cyber-attacks and adopts three Generative Adversarial Networks (GANs) in order to compose a combination of malicious PFCP traffic data that will next be redirected to UPF. Therefore, on the one hand, 5G-FUZZ can be used to uncover potential PFCP vulnerabilities against particular 5GC instances, and on the other hand, 5G-FUZZ can create malicious data that next can be used by AI-powered Intrusion Detection and Prevention Systems (IDPS) that rely on Machine Learning (ML) and Deep Learning (DL) techniques. Based on the aforementioned remarks, the paper contributions are summarised as follows.

- **Implementation of a Smart Fuzzer for 5GC**: A smart fuzzer is provided, targeting the PFCP communications in 5GC. Therefore, PFCP-related vulnerabilities can be discovered and remediated in the context of particular 5GC instances.
- **Generating Malicious Network Traffic Data**: The proposed fuzzer can synthesise and generate malicious PFCP traffic data (i.e., pcap files) that next can be further processed and used to train ML and DL models for intrusion detection.

The rest of this paper is organised as follows. Section II describes similar works in this research area. Next, section III discusses the PFCP-related attacks that are taken into account by 5G-FUZZ. In section IV, the architecture of 5G-FUZZ is presented. Finally, section V presents the evaluation results, while Section VI concludes this paper.

II. Related Work

Taking into account various papers that investigate the security issues of 5G, in this paper, we focus our attention on existing works that introduce attack generators for 5G environments. A brief overview of them is provided below.

In [3], G. Amponis et al. focus on PFCP Denial of Service (DoS) attacks against 5GC. Five attacks are studied: PFCP Session Deletion Request, Modification Request, Establishment, Misconfiguration, and Eavesdropping User Traffic. The first attack affects 5GC's N4 and N3 interfaces. A violated SMF sends to UPF a flood of PFCP Session Deletion Control Messages. The second attack enforces UPF to lose packet-handling parameters. A violated SMF again sends to UPF multiple PFCP Session Modification Requests with a DROP fag. The third attack depletes UPF's supplies. A malicious entity playing the role of SMF floods UPF with Session Establishment Requests. Next, Unauthorised UPF Forwarding Rules Misconfiguration can breach the /proc/sys/net/ipv4 directory if the attacker has compromised UPF. The last attack, an extension of the Unauthorised PFCP Session Modification Request, redirects the UE traffic from the UPF to a compromised entity. Finally, it is noteworthy that using the first three attacks, the authors published the 5GC PFCP Intrusion Detection Dataset in IEEE Dataport and Zenodo.

Z. Salazar et al. introduce 5G-Replay in [4]. This fuzzer is specifically designed for testing 5G protocols and allows the evaluation of various components in a 5G network, including both 5GC and radio-layer elements like cellular transceivers. The authors conducted experiments to assess the effectiveness of 5G-Replay using open-source 5G packages like Open5GS and Free5GC. It is worth noting that even when modifying protocol-specific attributes, the replayed 5G traffic is successfully parsed by the corresponding elements, and the expected responses are received.

In [5], Z. Lin et al. introduce IDSGAN. IDSGAN relies on Wasserstein Generative Adversarial Network (WGAN) in order to generate appropriate data that next can feed AI-powered Intrusion Detection Systems (IDS). The architecture of WGAN consists of two complementary networks, namely, generator and discriminator. The generator is composed of five linear layers, which are responsible for generating malicious data. On the other hand, the discriminator's role is to classify the data generated by the generator as real or fictitious. The implementation of IDSGAN is built upon the PyTorch framework. The performance of the proposed tool is evaluated in terms of its detection rate and evasion increase rate. The evaluation results show that IDSGAN achieves an average adversarial detection rate of 0.495% and an average evasion increase rate of 98.93%.

In [6], the authors introduce DoS-WGAN. DoS-WGAN adopts also WGAN to autonomously generate features of Denial of Service (DoS) traffic by leveraging the probability distribution of normal network packets. The architectural design of DoS-WGAN includes two complement neural networks: (a) the generator and (b) the discriminator. The objective of the generator is to generate malicious data related to DoS attacks, while the discriminator's purpose is to classify this data as genuine or counterfeit. Moreover, there is a converter which is used to convert the data into a suitable format. The evaluation process involves a Convolutional Neural Network (CNN)-based Network Intrusion Detection System (NIDS). DoS-WGAN decreases the True Positive Rate (TPR) of the previous NIDS from 97.3461% to 47.627%.

Undoubtedly, the previous works introduce interesting methodologies and solutions in order to generate malicious data, taking full advantage of GAN. However, it is worth mentioning that none of them focuses on actual 5GC environments. 5GReplay can only generate malicious data within a 5GC network; nevertheless, it does not include any AI mechanism. Finally, none of the previous works focuses on PFCP. Based on the previous remarks. to the best of our knowledge, this is the first work which introduces a smart fuzzer for the PFCP protocol, which is in the context of the N4 interface in the context of 5GC.

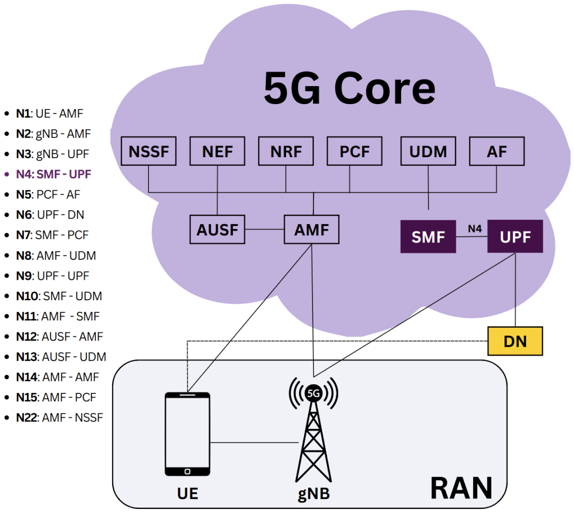

Fig. 1. Visual Representation of 5GC

III. Attack Modelling

As illustrated in Fig. 1, 5G-Fuzz targets the N4 interface of 5GC between UPF and SMF. For this interface, PFCP is used. Based on our previous work in [3], we focus our attention on the following PFCP-related attacks.

PFCP Session Deletion DoS Attack: The goal of this cyberattack is to drop the connection between a User Equipment (UE) and the Data Network (DN). More specifically, the Python script behind this cyberattack disrupts the sessions between the UE and DN's Protocol Data Units (PDUs) without disconnecting the UE from the 5G Radio Access Network (RAN) or 5GC. Consequently, this cyberattack affects the N4 (between SMF and UPF) and N6 (between UPF and DN) interfaces. It is worth mentioning that in order to accomplish this cyberattack in a successful manner, the cyberattacker has to identify first a valid Session Endpoint Identifier (SEID) used between SMF and UPF

PFCP Session Modification DoS Attack (DROP): The goal of this attack is to remove the packet-handling rules of a specific session between SMF and UPF. In particular, the cyberattacker, in this case, intends to remove the entries of the Forwarding Action Rules (FAR) that are associated with the Tunnel Endpoint Identifier (TEID) and the gNB IP address. Therefore, in the context of the N6 interface (between UPF and DN), the General Packet Radio Service (GPRS) Tunneling Protocol (GTP) tunnel used for transmitting downlink data to the UE is compromised, without allowing access to the DN. Similarly to the previous cyberattack, in this case, also, the N4 and N6 interfaces of 5GC are affected. However, the cyberattacker has to identify also the appropriate SEID in order to execute the attack successfully.

Although the previous cyberattacks are efficient by themselves, an attacker without prior knowledge about the target 5GC may require a lot of time for the identification of the appropriate SEID. Therefore, the goal of 5G-Fuzz is to fully leverage historical data in order to generate similar, malicious data related to the parameters of the previous PFCP attacks. More specifically, 5G-Fuzz focuses on the seid and seq fields of the PFCP Session Deletion Request and PFCP Session Modification Request packets, as illustrated in Fig. 2 and Fig. 3, respectively. seid is an identifier utilised in all PFCP packets in order to identify the PFCP session of a specific UE with the DN. On the other side, seq is a 16-bit identifier which denotes the number of the PFCP packets that are transmitted between SMF and UPF. Changing seid could lead the SMF and UPF to send packets to the wrong PFCP sessions. In particular, in the context of the PFCP Session Deletion DoS Attack, fuzzing seid in a smart manner, taking into account historical data, can drop established PFCP sessions between UPF and SMF. In contrast, without any intelligence, a PFCP Session Deletion DoS Attack will increment the value of seid one by one. It is evident that this would require more time in order to identify and target the seid of interests. The intelligence of 5G-Fuzz accelerates the attack process by identifying and targeting faster the seid of interest and thus disabling a larger proportion of the affected network slice's user plane. In the context of PFCP Session Modification DoS Attack, fuzzing seid could drop the packet-handling rules of established PFCP sessions. Similarly, in this case, the intelligence of 5G-Fuzz allows the cyberattacker to target faster the seid of interest, accelerating the overall process. On the other hand, changing seq can result in receiving PFCP messages out of order. Similarly to the previous cases, a smart fuzzer like 5G-Fuzz can identify faster the seq of interest compared to conventional PFCP Session Deletion DoS and PFCP Session Modification DoS attacks.

```
###[ PFCP (v1) Header ]###
           version   = 1
           spare_b2  = 0x0
           spare_b3  = 0x0
           spare_b4  = 0x0
           MP        = 0
           S         = 1
           message_type= session_deletion_request
           length    = 12
           seid      = 0x1
           seq       = 101
           spare_oct = 0
```

Fig. 2. Structure of PFCP Session Deletion Request

IV. 5G-FUZZ Architecture

The architecture of 5G-Fuzz consists of two main application building blocks, namely (a) Training and (b) Inference. In the first application building block, there are three components, namely (a) Data Receiving Module, (b) PcapToCSVConverter and (c) Training Module. The first module is responsible for receiving and reading the PFCP network traffic data (i.e., PFCP pcap files). For this purpose, Scapy is used. Next,

```
###[ PFCP (v1) Header ]###
           version   = 1
           spare_b2  = 0x0
           spare_b3  = 0x0
           spare_b4  = 0x0
           MP        = 0
           S         = 1
           message_type= session_modification_request
           length    = 52
           seid      = 0x5
           seq       = 106
           spare_oct = 0
###[ PFCP Session Modification Request ]###
              \IE_list   \
               |###[ IE Update FAR ]###
               |  ietype    = Update FAR
               |  length    = 36
               |  \IE_list   \
               |   |###[ IE FAR ID ]###
               |   |  ietype    = FAR ID
               |   |  length    = 4
               |   |  id        = 1
               |   |  extra_data= ''
               |   |###[ IE Apply Action ]###
               |   |  ietype    = Apply Action
               |   |  length    = 1
               |   |  spare     = 0x0
               |   |  DUPL      = 0
               |   |  NOCP      = 0
               |   |  BUFF      = 0
               |   |  FORW      = 0
               |   |  DROP      = 1
               |   |  extra_data= ''
               |   |###[ IE Update Forwarding Parameters ]###
               |   |  ietype    = Update Forwarding Parameters
               |   |  length    = 19
               |   |  \IE_list   \
               |   |   |###[ IE Destination Interface ]###
               |   |   |  ietype    = Destination Interface
               |   |   |  length    = 1
               |   |   |  spare     = 0x0
               |   |   |  interface = Access
               |   |   |  extra_data= ''
               ...
```

Fig. 3. Structure of PFCP Session Modification Request

PcapToCSVConverter undertakes to parse the PFCP packets and extract the PFCP attributes from the PFCP Session Modification Request and PFCP Session Deletion Request packets. Then, a Comma-Separated Values (CSV) file is created for each category of packets (i.e., PFCP Session Modification Request and PFCP Session Deletion Request). Finally, the Training Module uses the previous CSV files in order to train two Conditional Tabular GANs (CTGANs), namely `DEL-GAN` and `MOD-GAN`. According to the structure of the PFCP Session Deletion Requests and PFCP Session Modification Requests packets as illustrated in Fig. 2 and Fig. 3, `DEL-GAN` is responsible for generating only the attributes of PFCP Session Deletion Requests, while the `MOD-GAN` generates only PFCP Session Modification Requests. On the other hand, in the context of Inference, the previous GANs generate the attributes of PFCP Session Deletion Requests and PFCP Session Modification Requests, respectively. The output of both GANs is received by the CSVToPCAPConverter, which undertakes to transform the previous PFCP attributes into PFCP Session Deletion Requests and PFCP Session Modification Requests packets, thus composing a single pcap file with PFPC packets that refer to both cyberattacks described previously. To this end, `Scapy` is used. Finally, the Replay Module uses `tcpreplay` in order to replay previous PFCP packets to the target UPF.

For the implementation of the `DEL-GAN` and the `MOD-GAN`, CTGAN is utilised. In this type of GAN architecture, conditional information is incorporated into the training process in order to guide the generation process and generate samples conditioned on specific characteristics, such as discrete values. The architecture of both GANs is illustrated in Fig. 5. In particular, the architecture of both GANs consists of two complementary neural networks: (a) the conditional generator and (b) the discriminator. The conditional generator takes as input samples from latent space Z and the conditional information and produces synthetic samples. The discriminator receives both samples from the real dataset and the synthetic samples, including the corresponding conditional information. Next, the aim of the discriminator is to distinguish between real and artificially generated samples while also considering the conditional information.

During the training process, the generator tries to generate samples that the discriminator identifies as real, while the discriminator aims to correctly identify both real samples and generated samples according to their conditional information. This adversarial training process aims to lead the generator to improve its ability to generate realistic synthetic samples.

The generator consists of two fully-connected hidden layers of 256 neurons, each with batch-normalization and the ReLU activation function (Equation 1). The discriminator also consists of two fully-connected hidden layers of 256 neurons each, with dropout and the LeakyReLU activation (Equation 2). The WGAN loss with gradient penalty (Equation 3) is used for the models training, as this loss results in more stable training of GANs. The models are trained for 300 epochs, utilising the Adam optimiser.

$$ReLU = \begin{cases} x & , \ x > 0 \\ 0 & , \ x \leq 0 \end{cases} \tag{1}$$

$$LeakyReLU = \begin{cases} x & , \ x > 0 \\ 0.01x & , \ x \leq 0 \end{cases} \tag{2}$$

$$\begin{aligned} L = \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g} \left[D\left(\tilde{\mathbf{x}}\right) \right] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r} \left[D\left(\mathbf{x}\right) \right] + \\ \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}} \left[\left(|| \nabla_{\tilde{\mathbf{x}}} D\left(\tilde{\mathbf{x}}\right) ||_2 - 1 \right)^2 \right] \end{aligned} \tag{3}$$

where D is the Discriminator, $\mathbb{P}_g$ is the generator distribution, $\mathbb{P}_r$ is the data distribution, and λ is the penalty coefficient.

V. Evaluation Analysis

For the evaluation of `5G-FUZZ`, the `5GC Intrusion Detection Dataset` [7] is used, while also a set of

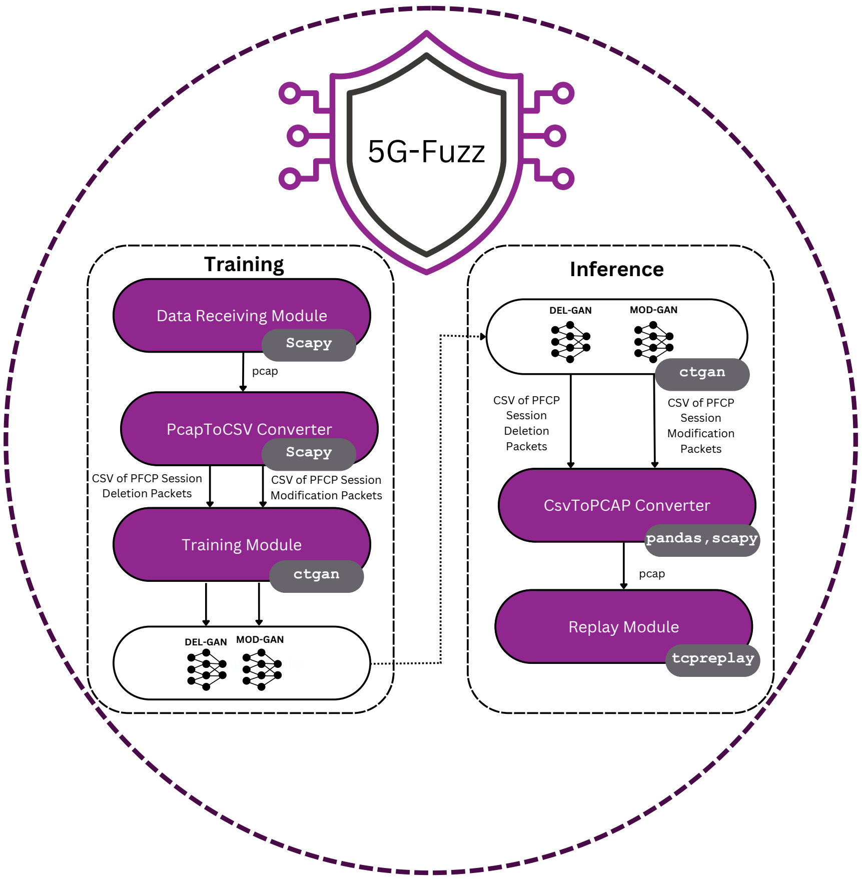

Fig. 4. Visual Representation of 5G-Fuzz Architecture

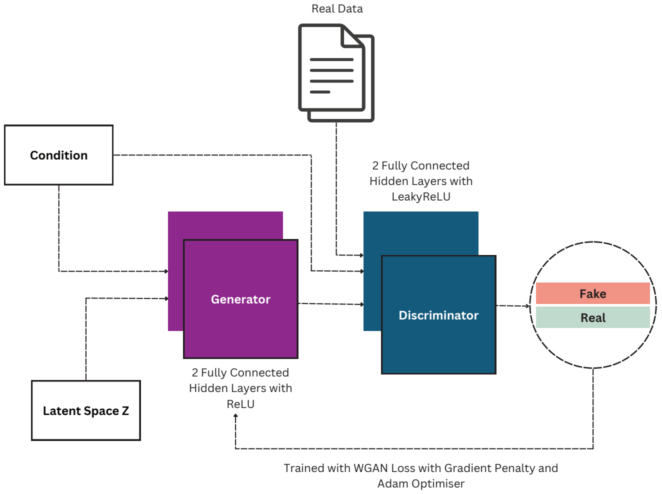
Real Data
Condition
2 Fully Connected Hidden Layers with LeakyReLU
Generator
Discriminator
Fake
Real
Latent Space Z
2 Fully Connected Hidden Layers with ReLU
Trained with WGAN Loss with Gradient Penalty and Adam Optimiser

Fig. 5. Visual Representation of DEL-GAN and MOD-GAN

metrics has been defined, that evaluates the synthetic data compared to the real one. The metrics are calculated for both the outputs of the `DEL−GAN` and `MOD−GAN`. In particular, the following metrics are used to evaluate the features with discrete values:

The Total Variation Distance (TVD) metric (Equation 4) computes the similarity of a real data and a synthetic data column in terms of the column shapes.

$$TVD = 1 - \frac{1}{2} \sum_{\omega \in \Omega} |R_\omega - S_\omega| \tag{4}$$

where ω describes all the possible categories in column Ω, and R, S describe the real and synthetic frequencies for those categories respectively.

The $p-value$ of the Chi-Squared Test (Equation 5) is used to quantify the similarity of real and synthetic columns by considering the column shapes.

$$X_c^2 = \frac{\sum_i (O_i - E_i)^2}{E_i} \tag{5}$$

where O is the observed value and E is the expected value.

The Category Coverage (CatCov) metric (Equation 6) measures if a discrete features column from the synthetic data covers all the possible categories that are present in the same column from the real data.

$$CategoryCoverage = \frac{c_r}{c_s} \tag{6}$$

where c_r, c_s are the unique categories in synthetic and real columns, r and s respectively.

The Discrete Kullback–Leibler divergence (D-KLD) metric (Equation 7), measures how one discrete probability distribution P is different from another discrete probability distribution Q. The final value for this metric is normilized by: $1/(1 + DD_{KL})$.

$$DD_{KL}(P(x)||Q(x)) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)} \quad (7)$$

where P, Q, are discrete probability distributions.

On the other hand, the following metrics are used to evaluate the features with continuous values:

The Inverted Kolmogorov–Smirnov (IKS) metric (Equation 8) can be used to assess whether two one-dimensional probability distributions differ.

$$IKS = 1 - sup\,|F_{1,n}(x) - F_{2,m}(x)| \quad (8)$$

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of the first and the second sample respectively, sup is the supremum function, and n, m are the sizes of first and second samples respectively.

The Pearson's correlation coefficient (PCC) metric computes a correlation coefficient on the real and synthetic data.

$$PCC = \frac{cov(X, Y)}{\sigma_x \sigma_y} \quad (9)$$

The Statistic Similarity (StatS) metric (Equation 10) computes the mean value for real data and synthetic columns, r and s, normalizes the score and takes its complement.

$$StatS = 1 - \frac{|mean(r) - mean(s)|}{max(r) - min(s)} \quad (10)$$

The Continuous Kullback–Leibler divergence (C-KLD) metric (Equation 11) measures the relative entropy between distributions p, q of continuous variables. The final value for this metric is normalized by: $1/(1 + CD_{KL})$.

$$CD_{KL}(p(x)||q(x)) = \int_{-\infty}^{\infty} p(x) log \frac{p(x)}{q(x)} dx \quad (11)$$

Based on the previous evaluation metrics, Table I and Table II summarise similarity evaluation metrics for `DEL-GAN` and `MOD-GAN`, respectively.

TABLE I
SUMMARY OF DEL-GAN SIMILARITY EVALUATION METRICS

Discrete Features

Metric	Value	Min	Max	Goal
TVD	1	0	1	Maximize
Chi-Squared	1	0	1	Maximize
CatCov	1	0	1	Maximize
D-KLD	1	0	1	Maximize

Continuous Features

Metric	Value	Min	Max	Goal
IKS	0.98	0	1	Maximize
PCC	0.99	-1	1	Maximize
StatS	0.97	0	1	Maximize
C-KLD	0.98	0	1	Maximize

TABLE II
SUMMARY OF MOD-GAN SIMILARITY EVALUATION METRICS

Discrete Features

Metric	Value	Min	Max	Goal
TVD	1	0	1	Maximize
Chi-Squared	1	0	1	Maximize
CatCov	1	0	1	Maximize
D-KLD	1	0	1	Maximize

Continuous Features

Metric	Value	Min	Max	Goal
IKS	0.99	0	1	Maximize
PCC	1	-1	1	Maximize
StatS	0.91	0	1	Maximize
C-KLD	0.99	0	1	Maximize

VI. CONCLUSIONS

Despite the advantages of 5G, it also comes with significant security and privacy concerns. In this paper, we present `5G-Fuzz`, a smart fuzzer that leverages historical data to fuzz and target the PFCP communications between SMF and UPF of 5GC. Unlike conventional fuzzers, `5G-Fuzz` employs two GANs to identify and generate appropriate values for the SEID and seq used in the PFCP sessions, thereby accelerating the various PFCP attacks. Then, `5G-Fuzz` creates and replays malicious PFCP packets against UPF.

REFERENCES

[1] S. Giannakidou, P. Radoglou-Grammatikis, S. Koussouris, M. Pertselakis, N. Kanakaris, A. Lekidis, K. Kaltakis, M. P. Koidou, C. Metallidou, K. E. Psannis *et al.*, "5G-Enabled NetApp for Predictive Maintenance in Critical Infrastructures," in *2022 5th World Symposium on Communication Engineering (WSCE)*. Nagoya, Japan: IEEE, 2022, pp. 129–132.

[2] G. Amponis, P. Radoglou-Grammatikis, T. Lagkas, S. Ouzounidis, M. Zevgara, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Towards Securing Next-Generation Networks: Attacking 5G Core/RAN Testbed," in *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*. Tripolis, Greece: IEEE, 2022, pp. 1–4.

[3] G. Amponis, P. Radoglou-Grammatikis, T. Lagkas, W. Mallouli, A. Cavalli, D. Klonidis, E. Markakis, and P. Sarigiannidis, "Threatening the 5G core via PFCP DoS attacks: the case of blocking UAV communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, pp. 1–27, 2022.

[4] Z. Salazar, H. N. Nguyen, W. Mallouli, A. R. Cavalli, and E. Montes de Oca, "5greplay: A 5g network traffic fuzzer-application to attack injection," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*. Vienna, Austria: Association for Computing Machinery, 2021, pp. 1–8.

[5] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," in *Pacific-asia conference on knowledge discovery and data mining*. Cham: Springer, 2022, pp. 79–91.

[6] Q. Yan, M. Wang, W. Huang, X. Luo, and F. R. Yu, "Automatically synthesizing DoS attack traces using generative adversarial networks," *International journal of machine learning and cybernetics*, vol. 10, no. 12, pp. 3387–3396, 2019.

[7] G. Amponis, P. Radoglou-Grammatikis, G. Nakas, S. Goudos, V. Argyriou, T. Lagkas, and P. Sarigiannidis, "5G Core PFCP Intrusion Detection Dataset," in *2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. Athens, Greece: IEEE, 2023, pp. 1–4.