OPEN ACCESS

KING'S College LONDON

King's Research Portal

Document Version
Peer reviewed version

Link to publication record in King's Research Portal

Citation for published version (APA):
Fernando, O. A., Xiao, H., & Spring, W. J. (Accepted/In press). Developing a Testbed with P4 to Generate Datasets for the Analysis of 5G-MEC Security. In *2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, United States* IEEE.

Citing this paper
Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights
Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

•Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
•You may not further distribute the material or use it for any profit-making activity or commercial gain
•You may freely distribute the URL identifying the publication in the Research Portal

Take down policy
If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. Dec. 2023

Developing a Testbed with P4 to Generate Datasets for the Analysis of 5G-MEC Security

Omesh A Fernando
Department of Computer Science
University of Hertfordshire
Hatfield, UK
w.k.fernando@herts.ac.uk

Hannan Xiao
Department of Informatics
King's College London
London, UK
hannan.xiao@kcl.ac.uk

Joseph Spring
Department of Computer Science
University of Hertfordshire
Hatfield, UK
j.spring@herts.ac.uk

***Abstract*—Service providers have now entered the implementation phase for 5G mobile telecommunication networks. With this, the concept of Multi-access Edge Computing (MEC) will play a crucial role when providing services on-the-go with low latency, high availability and high bandwidth. However, due to the low processing power of MEC nodes, adversaries may target the platform for malevolent purposes. In this paper we focus on building a realistic 5G-MEC testbed to run legitimate traffic and network attacks, and to collect 5G datasets for 5G-MEC. We also apply a Convolutional Neural Network to the dataset created on our testbed and to publicly available datasets. Our datasets and detection rate suggest that the employment of current public datasets for research based on 5G-MEC security, is now inappropriate.**

***Index Terms*—5G, Testbed, Network Attacks, 5G-MEC, Security, Datasets, UNSW NB-15, P4**

I. Introduction

5G mobile networks will become the key enabler and foundation for Information Communication Technology, catering to a diverse set of use cases (enhanced mobile broadband, massive machine type communication and ultra-low reliable low latency communication) with a range of different requirements. Providing support to each of the use cases using a common architecture has led to a significant change in design philosophy for core and radio access networks, and the application of Multi-access Edge Computing (MEC) involves a significant modification at the service based architectural level in order to (potentially) cater for the diverse use cases involved.

Multi-access Edge Computing and methods of deployment have been presented in [1] and as awareness and interest in 5G-MEC has grown within the industry and academia, research based on resource allocation, energy awareness and network slicing has also grown. Flexibility, scalability, virtualisation, and availability at the edge have created many advantages for both users and service providers. The importance of securing this architecture has gained the attention of researchers working in this field, however, their respective research in traffic analysis and intrusion detection for 5G-MEC does not employ publicly available datasets collected from a mobile telecommunication 5G testbed. Currently, popular datasets include KDD Cup'99 [2], NSL-KDD [3], CTU-13 [4] and UNSW NB-15 [5]. Of the above datasets, KDD Cup'99 was employed for example in [6] and [7], the CTU dataset

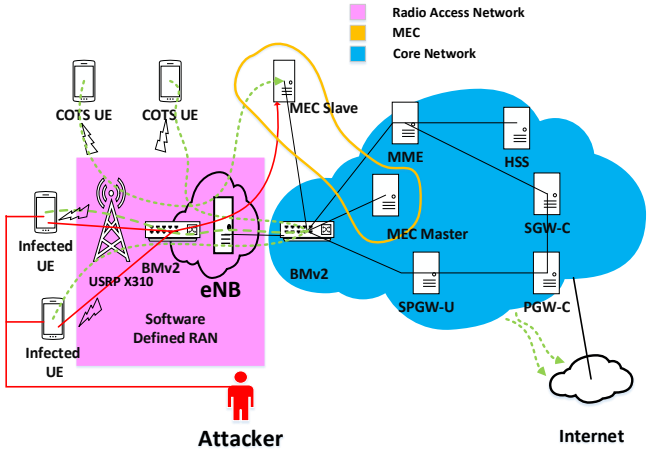


Figure 1: Testbed used to generate datasets for 5G-MEC security analysis

in [8] and UNSW NB-15 in [9] and [10]. However, KDD Cup'99 was collected in 1999 from a U.S. Air Force LAN network and NSL-KDD was created to improve the original 1999 dataset. Being outdated, collected on a LAN network and the simplicity of the traffic make the acquired datasets questionable for use in 5G. The CTU dataset was created by the CTU University of Prague in 2011 on a campus network and doesn't represent the traffic of a mobile telecommunication network. Finally, the UNSW NB-15 dataset created in 2015 using the IXIA tool on three virtualised servers, was not collected on a mobile telecommunication testbed. Traffic such as Quick UDP Internet Connections (QUIC), GPRS Tunnelling Protocol (GTP), S1 Application Protocol (S1AP), Stream Control Transmission Protocol (SCTP) and Simple Service Discovery Protocol (SSDP) are not represented in the above datasets. We therefore believe that, it is time to develop a testbed to generate realistic datasets for 5G based research.

From a security perspective frameworks/applications for 5G have been presented in [11]–[13]. However these studies do not include testbeds for their respective work but rather present frameworks based on literature for specific use cases. The inclusion of testbeds will, we believe provide a deeper understanding for the feasibility of the frameworks/applications. In this paper, we present a mobile telecommunication testbed capable of transmitting, capturing, and processing various

types of 5G mobile traffic.

The heterogeneity, diversity and complexity of the applications due to complex use cases in 5G (and beyond), will we believe benefit in having a flexible parser that can carry out 'match+action' in parallel. Having the capability to reconfigure, and to be both protocol and target (network devices i.e switches) independent, can aid towards heterogeneity, diversity and the complexity of the applications. For this research Programming Protocol Independent Packet Processing (P4), [14] serves as the prime candidate for inclusion in our testbed.

A. Research Questions

This research was conducted in order to address to the following questions.

- Which state-of-the-art platforms can be used to develop an appropriate testbed to generate 5G mobile network traffic?
- What configurations of P4 switches should be implemented on the 5G testbed?
- To what extent is the dataset generated by the developed 5G testbed effective for the analysis of 5G-MEC security?

B. Contributions

The key contributions from this study:

- A Testbed was developed employing functions of a 5G mobile telecommunication network with fully functioning P4-BMv2 switches in the Radio Access Network and Core Network.
- A dataset was generated from the testbed containing both legitimate and malicious types of traffic found in 5G networks. This dataset can help in mitigating the False Positive rate for an Intrusion Detection System, applied to a 5G mobile telecommunication network.
- The dataset was tested and compared to existing datasets not generated from a 5G testbed.

II. A Mobile Telecommunication Testbed

In order to develop a mobile telecommunication testbed we considered the following state-of-the-art deployment. Open Mobile Evolved Core (OMEC) , Network Simulator 3 (NS3) 5G-LENA project, Omnet++ and Mininet-wifi. The high resource requirements for OMEC, simulation nature of NS3 and Omnet++ and the missing LTE functions of Mininet-wifi made these options inappropriate. A better option we believe, was the OpenAirInterface (OAI) which we employed throughout this research.

OpenAirInterface [15] and [16] is an open-source development managed by the OpenAirInterface Software Alliance. It enables researchers to test and evaluate 5G (and beyond) case studies and scenarios whilst maintaining compliance to the 3GPP standards for both core networks (CN) and radio access networks (RAN).

In this research, OAI was used in emulating CN's consisting of the mobility management entity (MME), home subscriber server (HSS), the control and user plane separation serving gateway (SPGW-C/U and PGW-C). For this research OAI was installed on a Ubuntu Bionic distribution with 64GB of RAM running on a Core i7 CPU with 3.4GHz. Hyper-threading[1], CPU C-States and Speed-Step[2] have been disabled prior initialising the testbed. Our model of a real mobile telecommunication network is as shown in Fig.1.

A. User Equipment (UE)

OpenCells Subscriber Identity Modules (SIM) cards [17] were programmed to communicate with the mobile testbed. To connect legitimate and malicious users, Huawei E3372 Dongles with programmed SIM cards were used to connect to the mobile test-bed.

B. Radio Access Network (RAN)

RAN is denoted by the purple area in Fig.1. To model the Radio Access Network, the OAI core was connected to OAI5g [18] using a Software Defined Radio namely USRP x310, [19]. An Intel 10GbE x520 NIC[3] was used to connect the USRP x310 to the eNB PC. An LTE-Softmodem on band 7 was used in order connect eNB to the USRP x310, which facilitates access towards the UEs.

- SDR USRP x310: The Universal Software Radio Peripheral (USRP) denotes radio equipment developed by Ettus Research. The employment of USRP allows the user to define and create a Software Defined Radio (SDR) connected to a host PC via a Gigabit Ethernet port. An Intel 10GbE x520 NIC connects the two Gigabit Ethernet ports between the eNB and the USRP device. Two UBX160 daughter boards were employed on USRP. For the purpose of this research radio frequency band 7 was employed.
- eNB: Evolved Node B was instantiated using the host PC under the Ubuntu distribution with a low latency Linux kernel. The software implementation of eNB is equipped with a scheduler that handles the Upper Link and Down Link of a PHY radio. As per the 3GPP specification, Frequency Division Duplex (FDD) and Physical Uplink Shared Channel (PUSCH) information is also available upon activation of the eNB.
- P4-BMv2 Switch [20] : Given the requirements for faster processing of packets, a BMv2 switch was installed between RAN-CN (on an NIC card). An ingress pipeline, a custom parser and an egress pipeline was developed working in parallel to perform match+action. The increased capacity for faster processing of packets established our motivation for their employment. As shown in our previous publication [21], UDP traffic has a greater throughput with a P4 implementation than with Open vSwitch (OvS). Given that the communication between USRP and eNB are based on UDP traffic, the employment of a P4-BMv2 switch is an appropriate choice.

[1]A technology provided by Intel®which allowed more than one thread to run on each core

[2]A technology that allows the clock speed of the processor to be dynamically changed or adjusted by a software

[3]Intel®Ethernet Converged Network Adapter x520 provides better flexibility and scalability for cloud and data centre environments

C. Core Network (CN)

The CN is denoted in Fig.1 by the blue cloud. The following describe the various components of the CN, initialised on a Ubuntu UVT-Cloud [22] environment.

- HSS: HSS is a centralised database containing information relating to registered users and subscriptions. HSS provides data used for session creation, authentication and authorisation. The HSS database connects to the MME using a local loop-back. A Cassandra database stores and updates the records of the connected UEs.
- MME: The MME contains global information relating to the network. This includes attached and connected UEs, connected eNBs and bearers (default and S1U). MME communicates with eNBs, HSS and SPGW-C.
- SPGW-C: Provides control plane functions for the Serving Gateway (SGW-C) and Packet Data Network Gateway (PGW-C). It handles control requests from the MME and communicates with the SPGW-U. User traffic is tunnelled by SPGW-C as GTP (GPRS-Tunnelling Protocol).
- SPGW-U: This forwards user traffic between the Packet Data Network and the Internet. It is also, connected to the eNB.

D. Multi Access Edge Computing Platform (MEC)

The MEC deployment is depicted in yellow in Fig.1. For realising a 5G-MEC architecture four different types of deployments [1] are presented. We have employed (III) the MEC connected to a network edge point scenario due to its popularity [23]–[26].

The MEC architecture was instantiated using two UVT-Cloud environments, the MEC-Server and the MEC-Client. Both the MEC Slave and the MEC Master have been virtualized as UVT-Cloud environments running Ubuntu Bionic servers. Each server has been instantiated with 2GB of RAM, 10GB of HDD and 1 CPU core. This is to reflect the light weight, low processing power, that MEC nodes possess. An iPerf UDP server has been initiated on the MEC Client node with the bind option, for the UE's to transfer UDP data.

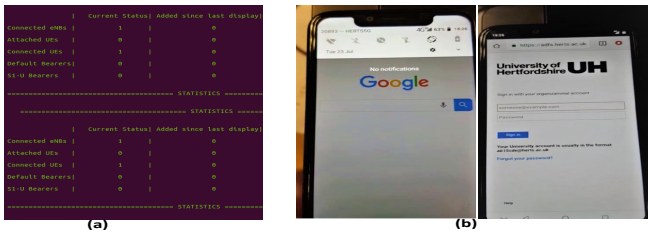


Figure 2: Testing the Connection. (a) MME periodic update illustrating connected eNBs and UEs (b) Access of Google and University learning platform on a COTS UE.

III. Evaluation of the Testbed

The traffic generation and workflow of the 5G testbed is shown in Fig. 1. Evaluation of the testbed is as follows. First we test the connectivity of the components in the testbed followed by the testbed with the UEs. Then, we evaluated the underlying 5G network traffic and finally we evaluated the user traffic in the developed testbed.

A. Testing the Connection

We first tested the UE connections. Fig.2(a) shows the periodic updates from the MME containing the connected devices of the network. Fig.2(b) shows the screenshots for the connection established after authentication and registration of the UEs to the testbed. The UEs were able to connect to the internet using the programmed mobile network- HERTS5G, which is an experimental 5G campus network. A ping message to the DNS also confirmed the successful connection in our testbed. We collected the underlying 5G network traffic once the UE connections had been established. Another key finding gathered from this experiment was that the P4-BMv2 switch was able to perform switching and packet processing for a higher volume of data (UDP traffic between USRP and RAN, see Fig 3f) with an average throughput of $2*10^9$bps, without the switch collapsing.

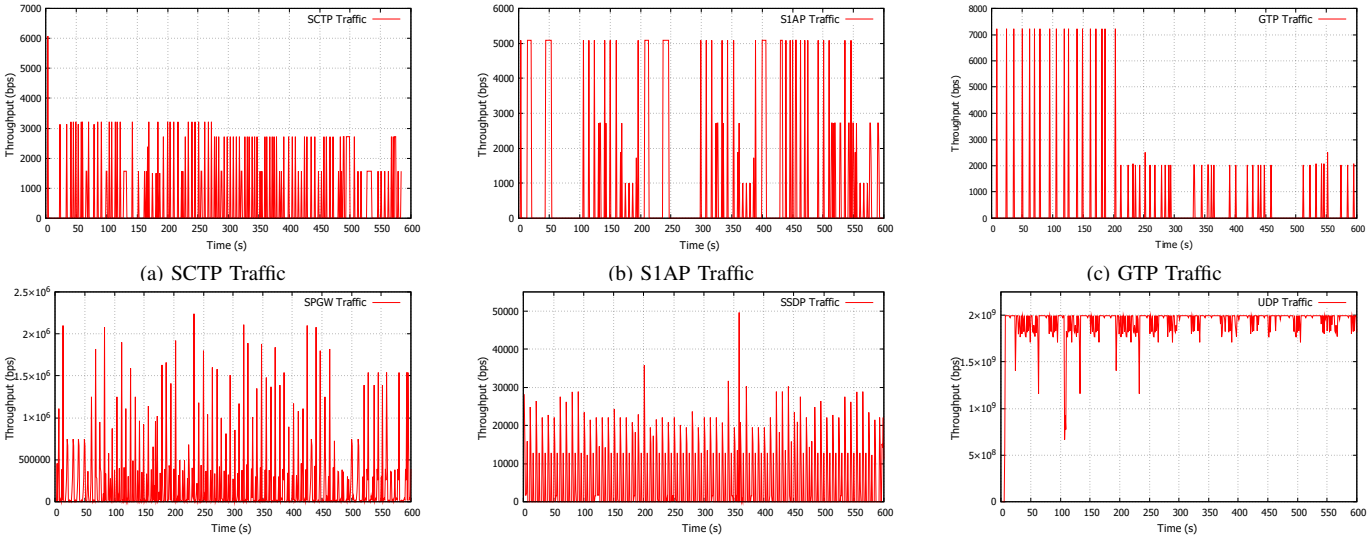
B. Underlying 5G Network Traffic

This traffic provides different types and protocols of traffic found in a 5G mobile telecommunication testbed. The underlying traffic (SCTP, S1AP, GTP, TCP (between MME and eNB), SSDP (between SPGW-U and DNS) and UDP (between USRP and eNB)) were observed in the collected dataset. Without introducing any user traffic, we collected the underlying traffic for a period of 600s. This type of traffic provides a dataset that can be used for training and testing algorithms used to enhance security for 5G communication.

Fig.3 presents six types of underlying 5G network traffic. SCTP traffic (Fig. 3a) between between SPGW-U, SGW-C and PGW-C occurs in the form of a heartbeat. Once generated, it is responded to with a heartbeat-acknowledgement. SCTP traffic carries the protocol number 132 and is an essential type of traffic in 5G mobile telecommunication for monitoring and detecting loss of sessions. S1AP traffic (Fig.3b) provides control plane signalling between RAN and EPC. S1AP traffic carries the protocol number 36412. The next type of traffic visible in a 5G mobile telecommunication network is GTP (Fig.3c) which is used by UEs to communicate with the DNS and other resources in the internet. GTP carries the protocol number 3386.

Fig.3d, represents the periodic TCP communication between MME and eNB. Since the MME require periodic updates about the connected/attached UEs, eNBs and bearers (tunnels for connecting UEs to packet data network) this periodic communication is extremely important for the network to maintain an accurate state. SSDP traffic (Fig.3e) aids the local network in service discovery. The captured traffic involves both SPGW-U and DNS. Fig.3f, represents the UDP traffic between the USRP and eNB. Due to the high volume and high speed required for communication between the two nodes, the communication is encapsulated as UDP traffic.

The above protocols and their underlying traffic have not been included in the following datasets (KDD Cup 99, NSL-KDD, CTU and UNSW NB-15). We note that the underlying TCP/UDP traffic in a 5G network is different in both volume and pattern from user-generated TCP/UDP traffic.



(a) SCTP Traffic (b) S1AP Traffic (c) GTP Traffic

(d) Periodic TCP traffic between eNB and MME (e) SSDP traffic from SPGW-U towards DNS (f) UDP data between eNB and USRP

Figure 3: Underlying 5G Network Traffic

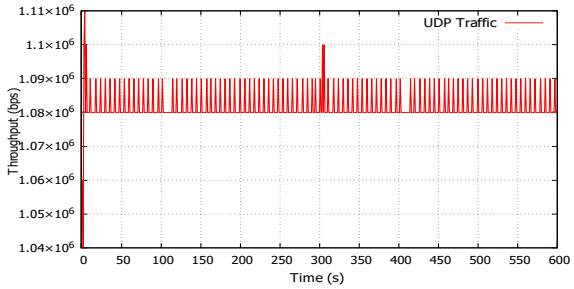


Figure 4: UDP Traffic directed towards MEC node

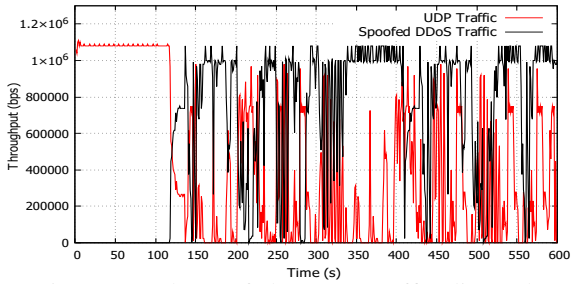


Figure 5: UDP and Spoofed DDoS Traffic directed towards MEC node

C. User Traffic

The third type of experiments generated legitimate UDP traffic from UE's to MEC node. A total of six iPerf clients were instantiated to communicate with the iPerf server at the MEC node. Data was collected for a period of 600s on each of the six hosts.

Fig.4 represents the average UDP throughput ($1.08*10^6$ bps) between the UE's and the MEC node. The fluctuations in the traffic are due to interference in the wireless medium. Underlying 5G network network traffic was also present but were not included in calculating the average UDP throughput.

D. Malicious Traffic

The fourth type of experiments generated malicious traffic from UE's towards MEC. A total of six clients initiated malicious traffic using hping3. Traffic such as DoS, Spoofed DDoS, botnet traffic, port scanning and TCP Syn Flood were employed in the malicious traffic category. Data was collected for a period of 600s on each of the six hosts. Underlying 5G network traffic and user traffic were also present during the malicious traffic transmission.

Fig.5 represents Spoofed DDoS traffic sent during the UDP transmission. The constant throughput of user traffic has been obstructed by the DDoS traffic. The throughput can be seen reaching 0bps at various intervals. Spoofed DDoS traffic was illustrated since it created a considerable service disruption compared to other types of malicious traffic.

IV. Generating and Evaluating Datasets

A. 5G Dataset

The collected traffic from the developed testbed was filtered using Python scripts for creating 5G datasets. Datasets included fields such as flow ID, source IP, source MAC, destination IP, source port, destination port, protocol, packet length, acknowledgement and a binary label for classification. The number of unique rows in the 5G dataset is shown in Table I. Out of the collected traffic, we filtered and refined the traffic to a total of 1,865,935 rows to remove redundant, repetitive and empty rows of data. We employed several Python scripts for refinement and creation of the datasets. In our 5G dataset, depicted in Table I, the underlying 5G network traffic and user traffic have been classified as legitimate and network attack traffic as malicious. Labelling network traffic into classes and the algorithms employed for same will shortly follow in a future publication. Table II presents a summary of the UNSW NB-15 dataset which we employed for detection along with our 5G dataset. From Tables I and II, we note that the underlying 5G network traffic is absent from the UNSW NB-15 dataset. Xu *et al.* [27] highlight that a result of an imbalanced or partial traffic classification, a higher False Positive rate follows. Although augmentation schemes have been employed to mitigate this drawback, Xu *et al.* state that the correct representation of generated datasets provide

Traffic Type		Number of Flows
Underlying 5G Network Traffic	SCTP	150,706
	S1AP	100,472
	GTP Traffic	50,236
	eNB and MME Traffic	251,176
	SSDP Traffic	25,118
	eNB-USRP UDP Traffic	401,882
	Total	979,590
User Traffic	UDP Traffic	468,847
Malicious Traffic	DoS	14,976
	DDoS	39,700
	TCP Syn Flood	142,604
	Botnet Traffic	39,700
	Port Scanning	180,518
	Total	886,345
Total Flows		1,865,935

Table I: Number of unique rows in the collected 5G dataset

UNSW NB-15		Number of Flows
User Traffic	TCP	1,492,153
	UDP	990,144
	ICMP	524
	Other	524
	Total	2,218,755
Malicious Traffic	Fuzzers	24,246
	Reconnaissance	13,987
	Shellcode	1511
	Analysis	2677
	DoS	16,353
	Exploits	44,525
	Worms	174
	Generic	215,481
	Total	321,283
Total Flows		2,540,038

Table II: Description of UNSW NB-15 dataset

better results with a negligible False positive rate. Hence, it is important to be able to classify underlying 5G network traffic as legitimate traffic. Total number of flows in both datasets (Table I and Table II) can be classified into two classes, malicious and legitimate. Hence, we employed binary classification in our detection methodology (CNN).

B. Results on Detection

Upon collection of the data and the creation of the dataset, we evaluated the effectiveness of the data with a deep learning algorithm, the Convolutional Neural Network (CNN) [28]. Ding and Zhai [29], evaluated the performance of CNN against other well-known classifiers (Long Short Term Memory, Deep Belief Networks, Random Forest and Support Vector Machine) and established the high accuracy and performance of CNN backed by the power of Deep Learning [27]. Hence, we have employed CNN to detect malicious traffic in 5G-MEC networks, since Deep Learning has a high reputation in performance [13] for data with high complexity and dimensions [30] (data generated through 5G mobile telecommunication). We also evaluated our 5G dataset, with the UNSW NB-15 dataset. Binary classification of recently collected network traffic was the rationale behind the employment of UNSW NB-15. The following formulae [31] were used as measures for accuracy (A), precision (P), recall (R) and F1 score.

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{P * R * 2}{P + R} \tag{4}$$

The abbreviations for the TP, FP, TN and FN are as follows. TP stands for the number of positively predicted attacks. FP stands for negatively predicted attacks on non-malicious traffic. TN represents non-malicious traffic that was correctly predicted as normal. Finally, FN stands for malicious traffic that was predicted as normal.

	Accuracy	Precision	Recall	F1-score
5G dataset	0.96	0.95	0.97	0.96
UNSW	0.84	0.85	0.83	0.83

Table III: Confusion matrix of the CNN for the two datasets.

The collected dataset was segregated in the ratio of 80:20, where the former was used as a training dataset while the latter was used as a validation (testing) dataset. Fig.6a, presents the acquired accuracy and loss (training and validation) for both datasets (5G dataset and UNSW NB-15 dataset). The CNN algorithm has successfully classified the traffic in the acquired dataset from our proposed testbed. The same approach was taken for the classification of UNSW NB-15 dataset. The algorithm successfully classified traffic into malicious and non-malicious traffic. Fig.6b, presents the loss for the two datasets during training and validation of the CNN algorithm. Table III presents the accuracy, precision, recall and F1-score resulting from the two testing datasets. An algorithm trained without underlying 5G network traffic (Fig 3a-f) classified as normal, may contribute towards a higher false positive rate when applied to a realistic 5G network for intrusion detection.

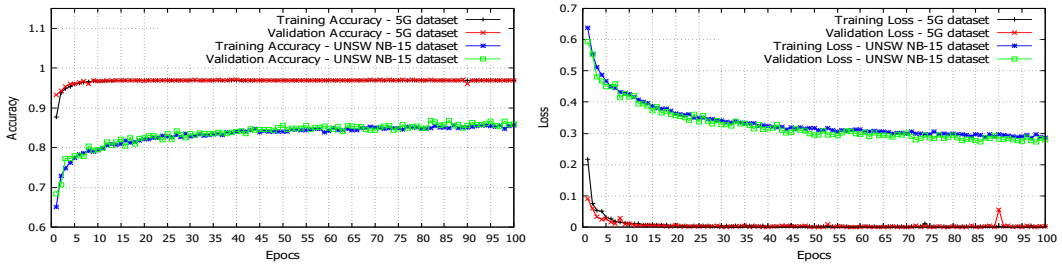
V. Conclusion and Future Work

In this study, we developed a 5G mobile telecommunication testbed to produce 5G datasets that can be used to study 5G traffic malicious attacks and their characteristics. We conducted this research to advance the effectiveness of security based research for 5G and beyond, using 5G datasets instead of simulators or emulators. Our UDP server has been instantiated using iPerf but the testbed is otherwise free from simulation or emulation. We collected traffic unique to a 5G mobile network, conducted malicious attacks on an MEC node, studied the service disruption and presented the associated data for the above. Details regarding the CNN, encoding/decoding algorithm employed in the 2D RGB image generation together with discussion on the computational complexity of the algorithms employed will follow shortly. We are currently working on mitigation techniques for malicious traffic using Deep Learning for 5G-MEC Security. The results and 5G-MEC datasets will follow in our future publication.

References

[1] Sami Kekki, Walter Featherstone, Yonggang Fang, et al. "MEC in 5G networks". In: *ETSI white paper* 28 (2018), pp. 1–28.

[2] *KDD Cup'99*. URL: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.



(a) Training and Validation Accuracy for the Datasets

(b) Training and Validation Loss for the Datasets

Figure 6: Training and Validation Data from the Proposed dataset and UNSW NB-15 dataset

[3] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, et al. "A detailed analysis of the KDD CUP 99 data set". In: *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE. 2009, pp. 1–6.

[4] *The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic*. URL: https://www.stratosphereips.org/datasets-ctu13.

[5] Nour Moustafa and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015, pp. 1–6. DOI: 10.1109/MilCIS.2015.7348942.

[6] Jiaqi Li, Zhifeng Zhao, and Rongpeng Li. "A machine learning based intrusion detection system for software defined 5G network". In: *arXiv preprint arXiv:1708.04571* (2017).

[7] Reeta Devi, Rakesh Kumar Jha, Akhil Gupta, et al. "Implementation of intrusion detection system using adaptive neuro-fuzzy inference system for 5G wireless communication network". In: *AEU-International Journal of Electronics and Communications* 74 (2017), pp. 94–106.

[8] Lorenzo Fernández Maimó, Ángel Luis Perales Gómez, Felix J Garcia Clemente, et al. "A self-adaptive deep learning-based system for anomaly detection in 5G networks". In: *IEEE Access* 6 (2018), pp. 7700–7712.

[9] Hichem Sedjelmaci, Sidi Mohammed Senouci, Nirwan Ansari, et al. "A Trusted Hybrid Learning Approach to Secure Edge Computing". In: *IEEE Consumer Electronics Magazine* (2021).

[10] Ibrahim Alrashdi, Ali Alqazzaz, Esam Aloufi, et al. "Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE. 2019, pp. 0305–0310.

[11] Jong-Hyouk Lee and Hyoungshick Kim. "Security and privacy challenges in the internet of things [security and privacy matters]". In: *IEEE Consumer Electronics Magazine* 6.3 (2017), pp. 134–136.

[12] Hongji Huang, Song Guo, Guan Gui, et al. "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions". In: *IEEE Wireless Communications* 27.1 (2019), pp. 214–222.

[13] Mohamed Amine Ferrag, Leandros Maglaras, Sotiris Moschoyiannis, et al. "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study". In: *Journal of Information Security and Applications* 50 (2020), p. 102419.

[14] Pat Bosshart, Dan Daly, Glen Gibb, et al. "P4: Programming protocol-independent packet processors". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 87–95.

[15] *5G CORE NETWORK*. URL: https://openairinterface.org/oai-5g-core-network-project/.

[16] Navid Nikaein, Mahesh K Marina, Saravana Manickam, et al. "OpenAirInterface: A flexible platform for 5G research". In: *ACM SIGCOMM Computer Communication Review* 44.5 (2014), pp. 33–38.

[17] *OpenCells-Programmable SIM Cards*. URL: https://open-cells.com/.

[18] *oai / openairinterface5G*. URL: https://gitlab.eurecom.fr/oai/openairinterface5g.

[19] *USRP SDR x310*. URL: https://files.ettus.com/manual/page_usrp_x3x0.html.

[20] P4 Language Consortium et al. "Behavioral model (bmv2)". In: *URL: https://github. com/p4lang/behavioral-model [cited 2020-01-21]* (2018).

[21] OA Fernando, Hannan Xiao, and Xianhui Che. "Evaluation of Underlying Switching Mechanism for Future Networks with P4 and SDN (Workshop Paper)". In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer. 2019, pp. 549–568.

[22] *Virtualization - uvt*. URL: https://ubuntu.com/server/docs/virtualization-uvt.

[23] Ji Li, Hui Gao, Tiejun Lv, et al. "Deep reinforcement learning based computation offloading and resource allocation for MEC". In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2018, pp. 1–6.

[24] Peng Liu, Bozhao Qi, and Suman Banerjee. "Edgeeye: An edge service framework for real-time intelligent video analytics". In: *Proceedings of the 1st international workshop on edge systems, analytics and networking*. 2018, pp. 1–6.

[25] Chien-Chun Hung, Ganesh Ananthanarayanan, Peter Bodik, et al. "Videoedge: Processing camera streams using hierarchical clusters". In: *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE. 2018, pp. 115–131.

[26] Angel Martin, Roberto Viola, Mikel Zorrilla, et al. "MEC for fair, reliable and efficient media streaming in mobile networks". In: *IEEE Transactions on Broadcasting* 66.2 (2019), pp. 264–278.

[27] Xing Xu, Jie Li, Yang Yang, et al. "Toward Effective Intrusion Detection Using Log-Cosh Conditional Variational Autoencoder". In: *IEEE Internet of Things Journal* 8.8 (2020), pp. 6187–6196.

[28] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[29] Yalei Ding and Yuqing Zhai. "Intrusion detection system for NSL-KDD dataset using convolutional neural networks". In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. 2018, pp. 81–85.

[30] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, et al. "A survey of deep learning and its applications: a new paradigm to machine learning". In: *Archives of Computational Methods in Engineering* 27.4 (2020), pp. 1071–1092.

[31] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, et al. "A deep learning approach for intrusion detection using recurrent neural networks". In: *Ieee Access* 5 (2017), pp. 21954–21961.