IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-9427-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOMWKSHPS57453.2023.10225868

Developing xApps for Rogue Base Station Detection in SDR-Enabled O-RAN

Jun-Hong Huang*, Shin-Ming Cheng*†, Rafael Kaliski‡, and Cheng-Feng Hung*
*Dept. of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan
Email: {m10915007, smcheng, and d10915002}@mail.ntust.edu.tw
† Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan.
‡Dept. of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, Email: rkaliski@ieee.org

***Abstract*—In order to support the diverse requirements of 5G communications, a multitude of RAN components are required. To enable multiple vendor support for 5G, each of whom can independently choose components, Open-RAN (O-RAN) defined a set of standards to which the components must adhere. In addition, O-RAN defines the management elements used to manage each component to secure the 5G networks. While the proposed architecture can manage both 4G and 5G environments, including 5G NSA (Non-Standalone), it inherently suffers from the same vulnerabilities found in 4G LTE. Consequently, an attacker can use unprotected signaling and a low-cost Software Defined Radio (SDR) to launch rogue base station (RBS) attacks on the user equipment (UE), even in O-RAN architectures. In this paper, we consider the stability of signals collected from high-quality operational BSs versus cheap RBSs. Using signal stability features, we develop a machine learning (ML) based RBS detector located on the UE. With the aid of an O-RAN xAPP, ML models can be retrained using the data collected from multiple UEs, and the updated model can be delivered to UEs to enable higher detection accuracy. We conduct extensive experiments by implementing an RBS using a USRP B210, enabling O-RAN using E-Release, and data collected from operational BSs. Moreover, the detector is implemented as an Android APP, which realizes the connection to the O-RAN xAPP. The experimental results show that our detector can achieve more than 99% accuracy, precision, recall, and F1 score.**

***Index Terms*—5G Non-Standalone (NSA), Attack Detection, RF Signature, Rogue Base Station (RBS) Attacks, Software-Defined Radio (SDR), Supervised Machine Learning**

I. Introduction

In addition to enabling a flexible 5G network architecture, O-RAN enables AI on the network and virtualizes the Radio Access Network (RAN) Intelligent Controller (RIC) and associated functions to enable interoperation between different vendors' components and devices [1], [2]. The RIC is further subdivided into different latencies so as to enable targeted toward specific latency Machine Learning (ML) models for more efficient radio resource management. Long-term policies are directed by the Non-Real Time (Non-RT) RIC's rAPPs, implemented by the Near-RT RIC's xAPPS [3].

With the aid of various APPs, O-RAN can utilize 5G to build upon the original 4G infrastructure while enabling high-speed connectivity. However, as 5G services are still in a nascent stage [4], they often lack the requisite security. To address this, 5G network security remains a key issue to ensure reliable connectivity [5], [6].

The recent innovation of low-cost Software Defined Radio (SDR) with open-source 5G software (such as srsRAN [7], and OpenAirInterface (OAI) [8]) enables the development of experimental base stations (BSs). Unfortunately, attackers can easily establish rogue BSs (RBSs) that trick victim UEs into connecting to themselves by sending a stronger signal strength than the operational BS [9]. When the victim UEs follow the signaling of an RBS, the RBS may perform harmful actions or leak a UE's private information [10]–[14].

To minimize the damage caused by RBSs, it is necessary to implement protection or detection mechanisms [15]–[17]. The detection parameters can be divided into the following four layers - physical (PHY) layer (e.g., the signal of received message [18]–[20]), Radio Resource Control (RRC) layer (e.g., cell, operator, or country identity of BS [18], [19], [21]), Non-Access Stratum (NAS) layer (e.g., signaling message flow [17]), and application layer (e.g., the real positions of operational BSs [19], [22]). A detector [19], [20] can calculate the physical location of the target BS and check whether it is secure by comparing the location to that of known operational BS locations, which are publicly available. Another detector, PHOENIX [17], checks the NAS messages received by the UE to determine whether the sequence is correct, and the content is expected to determine whether the connection is secure.

Under the assumption that RBS could forge all broadcasted RRC, NAS, and application messages, the detectors located at those layers might not have good detection performance. On the other hand, although the detectors using the PHY layer could distinguish the signal behaviors transmitted from RBS more clearly, the information collected by signal UE might introduce deviation. This paper proposes an xAPP located at Near-RT RIC to resolve the above challenges by gathering all PHY layer behavior sent from multiple UEs, training an ML-based classifier, and updating the model to the detectors located at UE. In particular, we choose signal stability as the main PHY layer feature for identification since the low-cost implementation of RBSs may exhibit unstable signaling.

Three famous ML models are developed, that is, Random Forest (RF), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), to validate the performance of the proposed detector. We conduct extensive experiments to evaluate the

978-1-6654-9427-4/23/$31.00 ©2023 IEEE

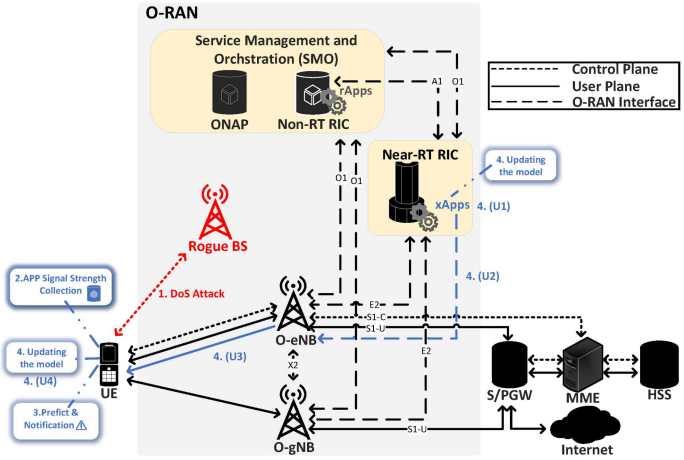


Fig. 1. System Architecture of SDR-enabled O-RAN.

RBS detector. We develop an SDR-enabled O-RAN environment where interfaces between BSs and O-RAN components are carefully implemented. Moreover, we build an RBS via a USRP B210 with the open-source software srsRAN and realize DoS attacks to paralyze victim UEs. The experimental results show that the proposed method can achieve an accuracy rate of nearly 100%. The results also demonstrate that the signal stability can reflect the behavior of RBSs, and with the aid of xAPP, malicious BS attacks can be more accurately identified.

II. Background

A. O-RAN and xApps

The Near-RT RIC xApp manages the resources and traffic of O-RAN network nodes to improve the QoE and QoS of the UE and the users. In existing research, there are many research works on how to help O-RAN network nodes optimize by combining ML models with xAPPs, four common categories are:

1) KPI Monitor xAPP: The main function of Key Performance Indicator (KPI) Monitor xAPP [23] is to collect various KPI metrics, such as throughput, from the current execution status of E2 nodes. The main purpose of this xAPP is to perform initial data collection and provide other xAPPs with the required KPI information for subsequent optimization control decisions.

2) Intelligent Optimization xApp: Cao *et al.* [24] deployed Federated Deep Reinforcement ML models between the RIC and E2 nodes to reduce UE handover frequency and to improve throughput. Orhan *et al.* [25] proposed a graph neural network and reinforcement learning application to optimize user association and load balancing.

3) QoE Optimization xApp: Kumar *et al.* [26] optimized Automatic Neighbour Relation with xApp, improved gNodeB handoff technology, reduced call drop rates, and increased the overall number of successful handoffs. Agarwal *et al.* [27] proposed a new QoE Enhancement Function xApp and used a new innovative Adaptive Genetic Algorithm to optimize the quality of users in the video service. Baldesi *et al.* [28] proposed a spectrum sharing application via an xAPP. Filali *et al.* [29] optimized resource allocation using deep reinforcement learning. Thaliath et al. [30] optimized different slices, service, and resource allocation with an xAPP.

4) Anomaly Detection xApp: Using the advantages of O-RAN architecture, the O-eNB can collect UE information and transmit it to xApp in the Near-RT RIC through the E2 interface for analysis and calculation to determine whether a malicious attack is occurring and consequently block the problematic UE via the Control-Plane.

B. RBS Attacks and Detection

Typically, the adversary utilizes RBS to enable attacks by initiating a stronger signal strength (Reference Symbol Received Power; RSRP), thereby causing the victim UEs to connect to the RBS. UEs recognize the attached RBSs as operational and legitimate ones and believe anything they deliver. In this case, RBS could easily cause the attached UE to voluntarily reveal sensitive information and disconnect all BSs so privacy and DoS attacks may be achieved.

Regarding the RBS detectors, FBSleuth [31] collected RSRP, and Radio Frequency (RF) features to collect evidence for RBS criminal attacks accurately. Murat [21] determined that any RSRP or (Reference Signal Received Quality; RSRQ) above a certain value is a fake base station. PHOENIX [17] received lots of attention since it first realized NAS signaling analysis. YAICD [16] could identify rogue BS attacks on the device using the existing detectors leveraging parameters. FBS-Radar [18] identified fake base stations by collecting suspicious SMS messages on end-user devices. Karacay *et al.* [20] measured in the simulator via USRP and then positioned the fake base station. Crocodile Hunter [19] also measured in the simulator via USRP and then positioned the fake base station but collected more information than [20], including BS's ID and location information for analysis.

III. Methodology

As we can see in Fig. 1, before the UE is attacked, it collects the signal strength from the neighboring BSs with the strongest signal strength and records the prediction result. If the predicted BS is connected, the UE is notified of the result. In the O-RAN architecture, we also use xApp for model training when we update the dataset and pass the latest model to the O-eNB through the E2 Interface and then to the UE through the User Plane for the model update on UE APP. In this section, we introduce how to collect data, train the ML model, and how to deploy/update the model into the UE app to detect RBSs.

A. System Model

We distinguish an RBS from a malicious one by continuously collecting data from the Synchronization Signal Block (SSB) and picking features from the RBS based on the sensitivity and stability of the received signals. The model for detecting RBSs is divided into four steps for analysis, namely: raw signal collection, feature extraction, model training, and

model evaluation. In the original signal collection, we used different telecommunication providers to collect data both indoors and outdoors. In the feature extraction step, we use judgmentally stable standard deviations as our features. In the model training step, the feature vectors are put into the model of our classification algorithm to generate a predictive model. Finally, in the model evaluation step, the model's performance is evaluated using different test samples.

Data Collection: The handset calculates the signal strength of the SSB signals of the connected BS and the neighboring BSs. We can make calls through the Android API and record the RSRP, PCI, and E-UTRA Absolute Radio Frequency Channel Number (EARFCN) of the connected and neighboring BSs. We finally chose to collect the signal strength every 10ms because the fastest response time of the UE we tested was 10ms.

The signal of the isolated RBS is stronger than the surrounding environment. While waiting for the UE to connect, the UE can find a strong BS from the neighboring BSs. Based on this feature, we can collect the signal strength of neighboring BSs, which have a stronger signal strength than the connected BS, and collect the signal strength of operational BSs both indoors and outdoors to determine whether the signal is affected by the difference in the environments. Finally, we collect the signal strength of sensed RBSs.

Feature Extraction: We selected 2000 data points as a dataset by feature extraction and picked a set of signal strength data of different operational base stations and different RBSs as shown in Figure 2. We found that the signal strength of the operational base station is more stable and consistent than the signal strength of the RBS. After collecting the signal strength datasets, we calculate the standard deviation to assist in evaluating signal stability. When the standard deviation value is larger, the difference between the value and the average value is larger, and vice versa, i.e. we can determine the stability of signal strength by these characteristics. As such, we use signal standard deviation as the main feature. The datasets are converted to standard deviation and subjected to Min-Max Normalization, which scales the data equally to the [0,1] region, thereby improving the speed and accuracy of the detector's convergence.

Model Training: In the training part of the model, we mainly verify whether the same model can determine the BSs of different operators and the equipment of different RBSs regardless of indoor or outdoor. In particular, The signal strength collected by one operational BS is used to test other operational BSs. In the RBS datasets section, the datasets of RBS devices are used for training and testing. Because there is no dependency between each dataset, we choose to use a supervised learning classification model to classify each set of signal strength features and select five different classifiers, namely RF, KNN, and SVM, for validation and analysis and finally adjust the parameters to make the model more accurate when learning.

Model Evaluation: We cross-validate the model five times and select the best model with the best parameters. The trained model is used to predict the test datasets and to evaluate whether the same model can discriminate between different operational and RBS devices, both indoors and outdoors. The best-performing classifiers were selected and the trained models were placed in the UE APP.

B. Procedures

We now describe how the detector detects RBSs and how the model is updated in the O-RAN environment.

Detection Process: We determine if the target BS is a RBS via the below steps.

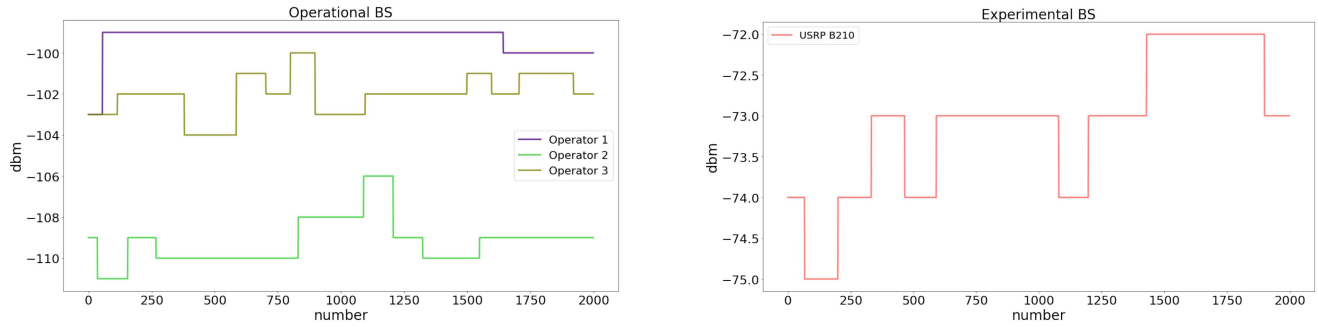
- Step D1: Find if any of the neighboring BSs have a signal strength higher than the strength of the BS that the UE is connected to.
- Step D2: If the signal strength of a neighboring BS is higher than that of the connected BS, sets of data will be collected continuously for the signal strength of this neighboring BS.
- Step D3: After the standard deviation calculation of the collected dataset, the values are normalized to the maximum and minimum.
- Step D4: Use the model to perform predictions of this dataset.
- Step D5: The results of the model prediction are kept for one minute, and if the UE connects to our predicted neighboring BS within one minute, the predicted results will be displayed. If there is no connection to the predicted neighboring base for more than one minute, the result will be discarded and retrieved.

Model Updates: We will place the initial model in the detectors located at UE and xApp located at Near-RT RIC. As shown in Fig. 1, the model will be updated in the following steps.

- Step U1: Put the latest datasets into xApp, and then let xApp perform model training.
- Step U2: The trained model is sent to O-eNB via E2 Interface.
- Step U3: The O-eNB is transmitted via the User Plane to the detector located at UEs.
- Step U4: Finally, the UE APP will replace the original model with the received updated model.

IV. Experimental Results

In this section, we set up the RBS and evaluate the performance of training classifiers on different datasets. First, we introduce the devices used in RBSs and launch the attacks. Then, in model training, we introduce the collection of datasets and evaluate the performance of different ML algorithms based on our selected features. Finally, we apply the proposed method to classifiers trained in different contexts to compare their performance.



(a) Operational BSs Signal. (b) RBSs Signal.

Fig. 2. (a) Signal strength of base stations for different operator (b) RBS signal strength for different devices

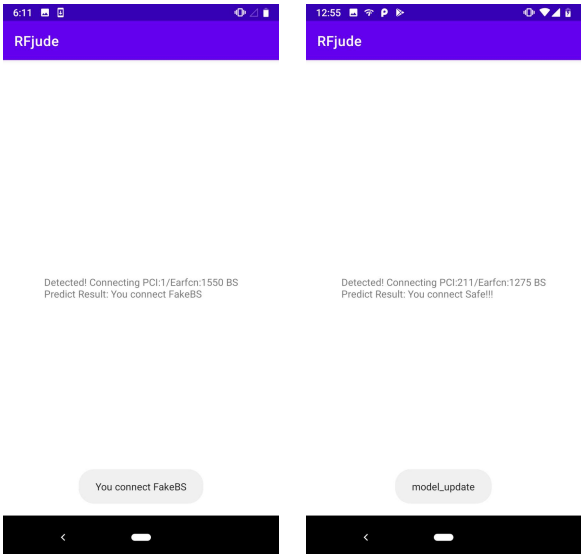
A. Environment Setup

RBS Side: The NUC is configured with an Intel Core i5 6500 CPU, four internal cores, four 3.2GHz threads, 24GB RAM, and a USRP B210 with a VERT900 antenna. First, we install the open-source software srsRAN on the device and listen to the BS's Mobile Country Code, Mobile Network Code, and Tracking Area Code in the environment. By modifying the srsRAN parameters, a BS similar to the operational BS is launched. The signal strength is higher than the operational BS to attract the victim UE. Finally, malicious code is implanted in srsRAN and sent to the victim UE, which causes the victim UE to be unable to connect with the operational BS after receiving the signaling, realizing a DoS attack.

UE Side: As described in III-A, a Google Pixel3 is adopted as a UE to collect signal strength. For the operational BS datasets, we use the operator sim card and the existing detector PHOENIX and Opensignal to confirm whether the connected BS is operational and then collect the signal strength. We collected the BS signal strengths from three operators at a location approximately 300 meters away from the legal BS, both indoors and outdoors. In the RBS datasets, we found that the attack success rate of the RBS is higher when the UE is about 300 meters away from the operational BS, which is consistent with the attack method. Therefore, we collect the signal strengths at a distance of 1 meter.

O-RAN: We use a server-class computer with Intel core i7-12700 CPU, 16GB RAM, ITB hard disk storage, Near-RT RIC based on Kubernetes cluster deployment, and E2 interface under the O-RAN alliance's E-Release version of Ubuntu 20.04 as the operating system. Due to the containerized construction, it can be flexibly and quickly deployed.

ML-Based Models for Detector: Three kinds of famous ML models are applied in this experiment: RF, KNN, and SVM. To make the model learning more accurate, we use the parameter finding method and tune the model parameters. When the model is trained, we train by parameter search to find the parameter that allows the highest accuracy and use this parameter to test the prediction dataset. Regarding the training and testing datasets, we collect 500 sets from the operational BSs of 3 different carriers in the same administrative area. Only one BS in every 500 sets was collected for signal strength, so the EARFCN and PCI are the same. Five different EARFCN and PCI signal strengths are collected in each of the 500 sets. In addition, we collected 500 sets of signals received from RBS for testing.



(a) Detection Results (b) Model Update

Fig. 3. Common figure caption.

B. Detector and xAPP Implementation

The detector located at UE determines if there is a neighboring BS whose signal strength is greater than the operational BS. If so, it collects the signal strength from the neighboring BS. If the neighboring BS with the strongest signal does not change during the collection of 2000 data samples, the collected 2000 data samples will be used to calculate the standard deviation, and then the trained model will make a prediction. If not, the search will begin again for a neighboring

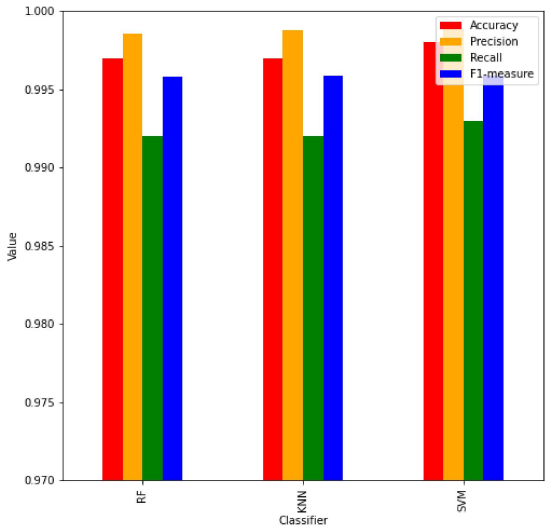


Fig. 4. Evaluation of ML models for the classifier.

BS with a stronger signal than the one you are connected to. The result will be displayed if the UE connects to the predicted BS within one minute. If the prediction is successful, the RBS will be displayed as shown in Fig. 3(a).

By combining the signal strength collected from detectors at the UEs located in different geographical areas, xAPPs could realize a complete prediction at a higher detection rate. When a UE collects new data from surrounding BSs, the data is updated to the xAPP to train a new model. Then the updated model is delivered to all O-eNBs through the E2 Interface. Each O-eNB will transmit the model through the User Plane to the UEs, and a notification will pop up on the UE to inform them that the model has been updated as shown in Fig. 3(b).

C. Performance Evaluation

Fig. 4 evaluates the performance of the ML models applied in the detector by considering all signals collected from all operators in both indoor and outdoor scenarios. All models can obtain 99% accuracy. The SVM model outperforms all other models.

We bow analyze the impacts of operators on signal stability. In particular, we trained the ML-based detectors using signals collected from operator 1 to see if the trained model works well in detecting signals collected from operators 2 and 3. The experimental results in both indoor and outdoor scenarios and shown in Table I and Table II, respectively. As shown in Table I, we found that the model trained by signals collected from Operator 2 could not detect signals collected from Operator 1 well in indoor scenarios. This is because more BSs are located in the experimental area, and the signal condition is much more complicated than that for Operator 2. Therefore, using the Operator 2 trained model to determine the signals from other operational BSs is less desirable. On the contrary, the detector trained by complicated signals from Operator 1 obtains higher accuracy in detecting RBS. Regarding the outdoor scenario shown in Table II, the signal condition of Operator 2 becomes much more complicated, which introduces better performance than the indoor scenario.

V. Conclusion

We use the signal strength stability feature to train our ML model. The datasets we collected demonstrate our model achieves high than 99% accuracy. We also deployed our ML model in the UE to predict the RBS to achieve the effect of real-time detection. Our detector can be used on any Android UE without the need for root permissions. With the help of the xApp in the O-RAN architecture, not only can the training process be transferred to the xApp to reduce the computational load on the UE, but also, the model update from the xApp can update the UE of the whole O-RAN network. To achieve higher deployment efficiency. Although the accuracy rate of the model is as high as 99%, it is not always successful in identifying RBSs. Therefore, in the future, if we can add the NAS layer detection proposed by PHOENIX, we can further reduce the errors in the model prediction and produce a more accurate APP to identify RBSs.

References

[1] A. Garcia-Saavedra and X. Costa-Perez, "O-RAN: Disrupting the virtualized RAN ecosystem," *IEEE Commun. Stds. Mag.*, vol. 5, no. 4, pp. 96–103, Dec. 2021.

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *arXiv preprint arXiv:2202.01032*, Feb. 2022.

[3] A. S. Abdalla *et al.*, "Toward next generation open radio access network – what O-RAN can and cannot do!" *arXiv preprint arXiv:2111.13754*, Nov. 2021.

[4] J. Feng, B.-K. Hong, , and S.-M. Cheng, "DDoS attacks in experimental LTE networks," in *Proc. IEEE AINA 2020*, Apr. 2020.

[5] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3GPP 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 170–195, 1stquarter 2020.

[6] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 196–248, 1stquarter 2020.

[7] "srsLTE: a free and open-source 4G LTE software suite," https://docs.srslte.com/en/latest/, [Online; accessed 19-July-2008].

[8] "OpenAirInterface: a flexible platform towards an open LTE ecosystem," https://www.openairinterface.org/?page_id=2762, [Online; accessed].

[9] 3GPP TS 33.501, "Security architecture and procedures for 5G system," Tech. Rep., Aug. 2020, version 16.3.0.

[10] S.-M. Cheng, B.-K. Hong, and C.-F. Hung, "Attack detection and mitigation in MEC-enabled 5G networks for AIoT," *IEEE IoT Mag.*, Sept. 2022.

[11] A. Shaik, R. Borgaonkar, J. Seifert, N. Asokan, and V. Niemi, "Practical attacks against privacy and availability in 4G/LTE mobile communication systems," in *Proc. NDSS 2016*, Feb. 2016.

[12] W.-L. Heish, B.-K. Hong, and S.-M. Cheng, "Toward large-scale rogue base station attacks using container-based virtualization," in *Proc. IEEE VTC 2019-Fall*, Aug. 2019.

[13] D. Rupprecht, K. Kohls, T. Holz, and C. Poepper, "IMP4GT: IMPersonation attacks in 4G NeTworks," in *Proc. NDSS 2020*, Feb. 2020.

[14] G. Lee *et al.*, "This is your president speaking: Spoofing alerts in 4G LTE networks," in *Proc. ACM MobiSys 2019*, June 2019, pp. 404–416.

[15] "Snoopsnitch," https://opensource.srlabs.de/projects/snoopsnitch, 2019, accessed: 2021-04-04.

[16] P. Ziayi, S.-M. Farmanbar, and M. Rezvani, "YAICD: Yet another IMSI catcher detector in GSM," *Security and Communication Networks*, vol. 2021, Jan. 2021.

[17] M. Echeverria, Z. Ahmed, B. Wang, M. F. Arif, S. R. Hussain, and O. Chowdhury, "PHOENIX: Device-centric cellular network protocol monitoring using runtime verification," in *Proc. NDSS 2021*, Jan. 2021.

TABLE I
PERFORMANCE OF SPECIAL OPERATIONAL BS FOR INDOOR UE.

	Testing Set	Operator 1				Operator 2				Operator 3			
Training Set	Classifier	Accuracy	Precision	Recall	F1-measure	Accuracy	Precision	Recall	F1-measure	Accuracy	Precision	Recall	F1-measure
Operator 1	RF	-	-	-	-	0.993	0.99309	0.993	0.99299	0.993	0.99309	0.993	0.99299
	KNN	-	-	-	-	0.993	0.99309	0.993	0.99299	0.993	0.99309	0.993	0.99299
	SVM	-	-	-	-	0.993	0.99309	0.993	0.99299	0.993	0.99309	0.993	0.99299
Operator 2	RF	0.883	0.89997	0.883	0.88174	-	-	-	-	0.986	0.986	0.986	0.986
	KNN	0.883	0.89997	0.883	0.88174	-	-	-	-	0.986	0.986	0.986	0.986
	SVM	0.986	0.986	0.986	0.986	-	-	-	-	0.993	0.99309	0.993	0.99299
Operator 3	RF	0.977	0.97715	0.977	0.97699	0.993	0.99309	0.993	0.99299	-	-	-	-
	KNN	0.993	0.99309	0.993	0.99299	0.993	0.99309	0.993	0.99299	-	-	-	-
	SVM	0.991	0.99104	0.991	0.99099	0.993	0.99309	0.993	0.99299	-	-	-	-

TABLE II
PERFORMANCE OF DETECTOR LOCATED OUTDOOR SPECIAL OPERATIONAL BS FOR OUTDOOR UE.

	Testing Set	Operator 1				Operator 2				Operator 3			
Training Set	Classifier	Accuracy	Precision	Recall	F1-measure	Accuracy	Precision	Recall	F1-measure	Accuracy	Precision	Recall	F1-measure
Operator 1	RF	-	-	-	-	0.993	0.9931	0.993	0.9929	0.993	0.9931	0.993	0.9929
	KNN	-	-	-	-	0.993	0.9931	0.993	0.9929	0.993	0.9931	0.993	0.9929
	SVM	-	-	-	-	0.993	0.9931	0.993	0.9929	0.993	0.9931	0.993	0.9929
Operator 2	RF	0.96	0.96124	0.96	0.95997	-	-	-	-	0.993	0.99309	0.993	0.99299
	KNN	0.984	0.984	0.984	0.98399	-	-	-	-	0.993	0.99309	0.993	0.99299
	SVM	0.986	0.986	0.986	0.986	-	-	-	-	0.993	0.99309	0.993	0.99299
Operator 3	RF	0.992	0.99207	0.992	0.99199	0.993	0.99309	0.993	0.99299	-	-	-	-
	KNN	0.992	0.99207	0.992	0.99199	0.993	0.99309	0.993	0.99299	-	-	-	-
	SVM	0.992	0.99207	0.992	0.99199	0.993	0.99309	0.993	0.99299	-	-	-	-

[18] Z. Li, W. Wang, C. Wilson, J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, and Y. Liu, "FBS-Radar: Uncovering fake base stations at scale in the wild," in *Proc. NDSS 2017*, Jan. 2017.

[19] C. Quintin, "Detecting fake 4G LTE base stations in real time," in *Proc. USENIX Enigma 2021*, Feb. 2021.

[20] L. Karaçay *et al.*, "A network-based positioning method to locate false base stations," *IEEE Access*, vol. 8, pp. 111 368–111 382, Aug. 2021.

[21] P. K. Nakarmi, M. A. Ersoy, E. U. Soykan, and K. Norrman, "Murat: Multi-RAT false base station detector," *arXiv preprint arXiv:2102.08780*, Feb. 2021.

[22] R. Borgaonkar, A. Martin, S. Park, A. Shaik, and J.-P. Seifert, "White-Stingray: Evaluating IMSI catchers detection applications," in *Proc. WOOT 2017*, Aug. 2017.

[23] "O-RAN OSC Onboarding and Deployment of xApps," https://wiki.o-ran-sc.org/display/IAT/Traffic+Steering+Flows, [Online; accessed].

[24] Y. Cao, S.-Y. Lien, Y.-C. Liang, K.-C. Chen, and X. Shen, "User access control in open radio access networks: A federated deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, 2021.

[25] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. Talwar, "Connection management xAPP for O-RAN RIC: A graph neural network and reinforcement learning approach," in *Proc. IEEE ICMLA 2021*, Dec. 2021, pp. 936–941.

[26] H. Kumar, V. Sapru, and S. K. Jaisawal, "O-RAN based proactive ANR optimization," in *Proc. IEEE GLOBECOM Workshop 2020*, Dec. 2020, pp. 1–4.

[27] B. Agarwal, M. A. Togou, M. Ruffini, and G.-M. Muntean, "QoE-driven optimization in 5G O-RAN enabled HetNets for enhanced video service quality," *IEEE Communications Magazine*, Sept. 2022.

[28] L. Baldesi, F. Restuccia, and T. Melodia, "ChARM: NextG spectrum sharing through data-driven real-time O-RAN dynamic control," *arXiv preprint arXiv:2201.06326*, Jan. 2022.

[29] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation O-RAN resource slicing for URLLC services using deep reinforcement learning," *arXiv preprint arXiv:2202.06439*, Feb. 2022.

[30] J. Thaliath, S. Niknam, S. Singh, R. Banerji, N. Saxena, H. S. Dhillon, J. H. Reed, A. K. Bashir, A. Bhat, and A. Roy, "Predictive closed-loop service automation in O-RAN based network slicing," *arXiv preprint arXiv:2202.01966*, Feb. 2022.

[31] Z. Zhuang, X. Ji, *et al.*, "FBSleuth: Fake base station forensics via radio frequency fingerprinting," in *Proc. ACM AsiaCCS 2018*, June 2018, p. 261–272.