RF Jamming Dataset: A Wireless Spectral Scan Approach for Malicious Interference Detection

Abubakar S. Ali, Govind Singh, Willian T. Lunardi, Lina Bariah, Michael Baddeley, Martin Andreoni Lopez, Jean-Pierre Giacalone and Sami Muhaidat

***Abstract*—The evolution of next-generation communication systems demands that wireless networks possess the attributes of awareness, adaptability, and intelligence. Wireless sensing techniques provide valuable information about the radio signals in the environment. However, hostile threats, such as jamming, eavesdropping, and manipulation, pose significant challenges to these networks. This paper presents a comprehensive study on an innovative RF-jamming detection testbed designed to combat these threats. The testbed leverages the spectral scan capability of wireless network interfaces and the jamming toolkit, JamRF, to accurately detect and mitigate jamming attacks. This study outlines the methodology used to develop the testbed, along with a detailed discussion of the rationales behind the design decisions. The accompanying RF jamming dataset, which includes experimentally measured data, is expected to promote the development and evaluation of jamming detection and avoidance systems. As a proof-of-concept, we trained five different machine learning algorithms and achieved a jamming detection accuracy of over 90% for all algorithms. The proposed RF jamming dataset and testbed represent a significant advancement in the fight against malicious interference in wireless networks.**

***Index Terms*—jamming; dataset; spectral scan; machine learning; software defined radio; experimental data.**

I. Introduction

Ensuring the security of communication networks is of utmost importance. While wired networks have been targeted by various types of attacks, the widespread adoption of wireless networks in recent years has made them a prime target for malicious activities. However, advances in technology have made wireless networks more affordable and easier to deploy, making them a popular choice for many organizations. Despite their popularity, wireless networks are known to be more susceptible to security attacks compared to wired networks due to the nature of wireless links [1]. The openness of the wireless medium makes it susceptible to both intentional and unintentional interference, with interference from neighboring cells being a prevalent form of unintentional interference in a wireless communication system. On the other hand, intentional interference refers to malicious attacks on a victim receiver that is not equipped to defend itself [2]. One such attack is the jamming attack, which actively transmits high energy to disrupt reliable data transmission or reception and can severely impact system performance [3].

A.S. Ali, L. Bariah, S. Muhaidat are with the KU Center for Cyber-Physical Systems, Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi 127788, UAE, (e-mails: asali,lina.bariah, muhaidat@ieee.org). M. Baddeley, M Andreoni, W.T, Lunardi and J.P. Giacalone are with the Secure Systems Research Center (SSRC) at the Technology Innovation Institute (TII), Abu Dhabi, UAE (e-mails: govind, willian, michael, willian, jean-pierre@ssrc.tii.ae).

To mitigate the impact of jamming attacks, researchers in academia, industry, and government, have dedicated significant effort to developing jamming detection and avoidance techniques. To facilitate these efforts, various datasets in different formats have been made available to the public to aid in the creation of jamming detection and avoidance systems. Puñal et al. [4] created a comprehensive dataset that includes multiple trace sets of 802.11p communications under different Radio Frequency (RF) jamming conditions. The RF jammer's operation patterns were analyzed, including constant, reactive, and pilot jamming. The observations were conducted in an anechoic chamber and outside in two key outdoor environments: an open area with a straight road and a densely populated building environment. Whelan et al. [5] collected a comprehensive dataset that comprises logs from a regular flight of an unmanned aerial vehicle (UAV) as well as one in which the UAV is subjected to global positioning system (GPS) spoofing and jamming. The experiment utilized a signal generator to precisely locate the UAV in Shanghai, China. GPS spoofing was achieved using the HackRF software-defined radio (SDR) and the GPS-SDR-SIM application to transmit the UAV's coordinates. GPS jamming was accomplished by broadcasting white Gaussian noise using the HackRF.

Despite the usefulness of WiFi traces data for gaining insight into the state of network channels, it does not furnish a comprehensive depiction of the utilization and status of the entire spectrum. Moreover, the traces obtained are in packet form, representing samples at the network layer, necessitating a packet sniffer for analysis. To address the limitation of WiFi traces data in providing a comprehensive view of spectrum utilization and conditions, various monitoring systems have been proposed and made accessible in the literature. Prominent among these datasets include the Google Spectrum [6] for television white-space measurements, the IBM Horizon project [7] that presents a decentralized architecture for sharing Internet of Things (IoT) data, and Microsoft's Spectrum Observatory [8], which enables spectrum sensing through the use of high-end sensors. The focus of Google's Spectrum and Blue Horizon on specific use cases results in their limited scope, while the high cost of the necessary sensing nodes impedes widespread deployment of Microsoft's Spectrum Observatory. In [9, 10], ElectroSense was proposed as a flexible and cost-effective testbed that leverages low-cost sensors to collect and analyze spectrum data through a crowd-sourcing paradigm. The primary goal of the initiative is to sense the full spectrum in diverse locations worldwide and provide processed spectrum data to users seeking a comprehensive understanding of spectrum utilization.

In contrast to earlier efforts, in this paper, we present

an experimental testbed for performing spectrum scanning using the in-built Wireless Local Area Network (WLAN) Interface Cards (NICs) of communication devices to collect data in different environments. Furthermore, using the testbed, we employ JamRF a jamming toolkit developed in [3], to synthesize different jamming scenarios and generate an RF jamming dataset. We outline the methodology used for developing the testbed and discuss the reasons for the choices made during its development to facilitate future improvements in the experimental exploration of jamming dataset production based on spectral scans. Our measurement data is neatly labeled into categories, which can be utilized in RF jamming analysis. This dataset can assist researchers in wireless security to conduct experimental evaluations of existing and future jamming detection and avoidance systems. Additionally, we provide an example scenario that can be used to construct experiment-driven jamming and avoidance systems and suggest avenues for further study using this dataset.

The remainder of this paper is structured as follows. The design of the proposed experimental setup is outlined in Section II, including the underlying principles and practical implementation of the testbed. The sample dataset obtained from the testbed is presented in Section III. An example application of the dataset is demonstrated in Section IV. The paper concludes in Section V.

II. Testbed Design and Implementation

In this section, we present the design and implementation of the testbed used for the measurement and analysis of the jamming signal generated by the JamRF toolkit [3]. Section II-A provides a comprehensive discussion of the testbed design, while Section II-B details the implementation of the design, including the usage of a Raspberry Pi Compute Module 4, a WiFi Radio for Spectral Scan, and a HackRF Jammer.

A. Testbed Design Based on JamRF

The proposed experimental design utilizes the JamRF toolkit [3] to conduct a jamming attack on all available 2.4/5GHz channels. The constant jammer configuration with Gaussian noise jamming signal is employed. The experiments are performed in three environments: an RF isolation chamber, a laboratory, and an office. To avoid disrupting the transmission activities of other users, the jamming attack is carried out inside the RF isolation chamber. The attack is executed using a HackRF with JamRF, and the captured signals are recorded at the receiver side using a Compute Module 4 (CM4) with a mounted Qualcomm Atheros device (QC9880) in background mode scanning. The receiver is positioned at different distances in $\{20, 40, 60\}$ cm from the jammer, and the jamming transmit power varies at $\{0, 5, 10\}$ dBm. For each distance and power combination, Fast Fourier Transform (FFT) samples are collected for approximately three seconds, and the process is repeated ten times with a 10-second pause between each iteration. In three scenarios, no jamming attack is conducted. These scenarios are low activity in the laboratory, high activity in the office, and no activity in the RF isolation chamber. The collected measurement data is organized and labeled into categories for ease of RF jamming analysis.

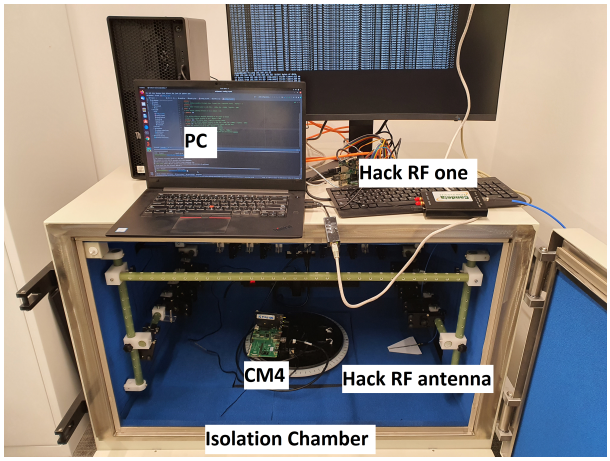


Fig. 1: Overview of the Spectral Scan Testbed.

B. Implementation of the Testbed

The implementation process of the testbed is depicted in Fig. 1. This section presents a comprehensive overview of the experimental details and measurement methods involved in our implementation. In particular, we describe the types of WLAN interfaces considered in the experiments and specify the parameters of the spectral scan testbed. Furthermore, we discuss the hardware and software components utilized in constructing the testbed.

1) Raspberry Pi Compute Module 4: The CM4 is a raspberry Pi 4 compact form factor primarily designed for embedded applications. It features a quad-core ARM Cortex-A72 processor and dual video output, among other interfaces. For this experiment, we utilize the CM4 Input-Output (IO) board, which serves as a development system for the CM4 and an embedded board for end products. The IO board enables the construction of systems using off-the-shelf components such as HATs and PCIe cards, including those for NVMe, SATA, networking, or USB. The major user connectors are conveniently located on one side for ease of enclosure design.

2) WiFi Radio for Spectral Scan: In our experimental testbed, we utilized two commercially available wireless modules, namely, the Qualcomm Atheros QCA9880 and Doodle Labs NM-DB-3U radio. The Doodle Labs NM-DB-3U is based on the Qualcomm AR958x chipset and supports IEEE 802.11n and 3x3 MIMO. It is an industrial-grade module that interfaces via mini PCIe and is supplied by Doodle Labs. The Qualcomm Atheros QCA9880, on the other hand, is a dual-band 3x3 MIMO 802.11ac/abgn chipset that is also interfaced via mini PCIe. Both of these modules are capable of conducting spectral scans, as they are equipped with the ATH10k (drivers/net/wireless/ath/ath10k/spectral.c) and ATH9k (drivers/net/wireless/ath/ath9k/common-spectral.c) wireless drivers, respectively, which are based on the mac80211 softmac architecture.

3) HackRF Jammer: In our experimental setup, we utilized the HackRF One, a wideband SDR half-duplex transceiver developed and manufactured by Great Scott Gadgets [11]. With the ability to both receive and transmit signals, this device supports frequencies ranging from 1 MHz to 6 GHz, with a

maximum output power of up to 15 dBm, depending on the band. The HackRF One includes a sub-miniature version A (SMA) antenna port, SMA clock input and output ports, and a USB 2.0 port, making it compatible with popular software-defined radio applications such as GNU Radio and SDR. As outlined in Section II, we employed JamRF, a jamming toolkit that implements various types of jammers and jamming strategies using the HackRF One and GNU radio [3].

Note that the ATH10k and ATH9k driver configurations do not automatically enable spectral scan by default. This required the specific activation of the CONFIG_ATH10K_SPECTRAL and CONFIG_ATH9K_COMMON_SPECTRAL features in the kernel configuration. To capture spectral data, an open-source tool (https://github.com/govindsi/utilities/blob/main/scripts/spectral_scan.sh) was utilized under various configurations (https://github.com/govindsi/utilities/tree/main/config/AP).

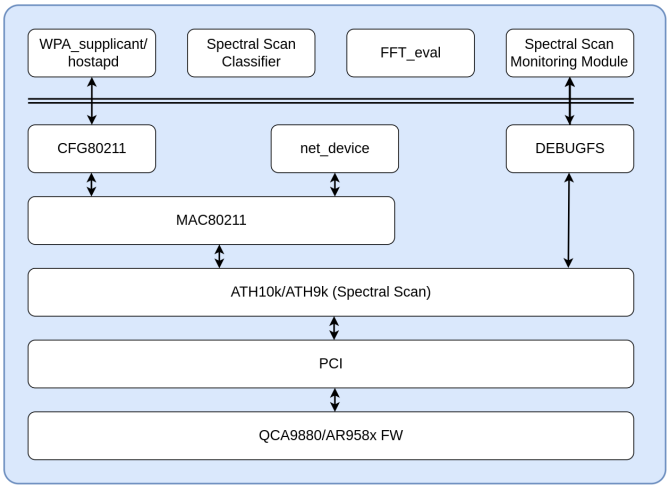


Fig. 2: Overall Architecture of the Spectral Scan System.

III. Data Set Organization and Characteristics

In this section, we provide a comprehensive overview of the data set accompanying this article. Section III-A presents the architecture of the Spectral Scan system used to generate the data set. Section III-B elaborates on the features obtained from the FFT data. Section III-C provides a visual representation of the Spectral Scan results, including an illustration of the impact of jamming and jammer configuration on the RF spectrum. Finally, Section III-D categorizes the data set based on the type of measurement and the parameters used in each experiment.

A. Spectral Scan System Architecture

The architecture of the spectral scan system is presented in Fig. 2. This system integrates multiple communication layers to facilitate spectral scanning functionality. The WPA_supplicant and hostapd components are utilized to configure the User Media Access Control (UMAC) mode, which can be set to access point, mesh, station, or independent basic service set modes. The spectral scan classifier is employed to classify the spectrum conditions, while the FFT_eval block is based on an open-source spectral scan pre-processing tool (https://github.com/simonwunderlich/FFT_eval). The tool's userspace program provides a graphical representation of the Fast Fourier Transform (FFT) samples collected from Atheros NICs, thereby facilitating the development of open-source spectrum analyzers for Qualcomm Atheros AR92xx and AR93xx-based chipsets. The ATH10k/ATH9k SPECTRAL_SCAN_CTL driver is used for spectral scan configuration, with the spectral data being captured via the DEBUGFS interface and transferred to the WiFi firmware through the Peripheral Component Interconnect (PCI) transport.

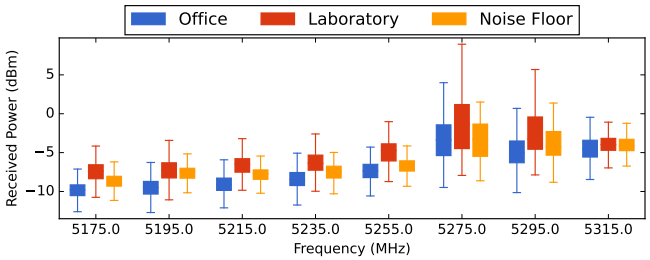
B. Spectral Scan Features

The Spectral Scan is a feature offered by some commercial off-the-shelf (COTS) wireless chipset products, which enables the collection of FFT data from the physical layer through software-controlled means. The Spectral Scan can be divided into two categories: high-latency and low-latency scans. The FFT data collected from the spectrum can be stored in a binary file format, which can then be post-processed to create an open-source spectrum analyzer or interference classifier. The binary data file contains eight primary features: frequency, noise, max magnitude, total gain in dB, base power in dB, relative power in dB, average power in dB, and received power in dBm. These features can be extracted from the Spectral Scan datagram header. The received power in dBm feature is calculated using the received power equation specified in the Qualcomm Atheros AR92xx and AR93xx chipset documentation. Following the parsing process, the data is stored as a comma-separated values (CSV) file, which can then be utilized for training machine learning (ML) algorithms. The CSV file provides time-series data derived from the in-phase-quadrature (IQ) samples binary file. The binary and CSV data files have been preserved and made accessible for reference in conjunction with this paper.

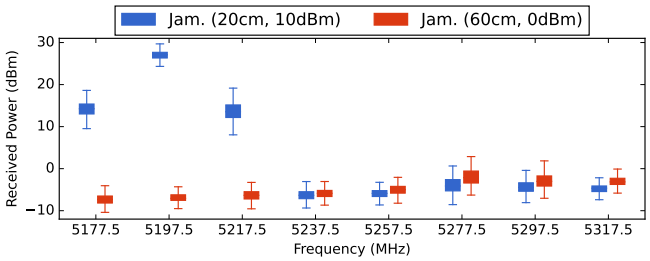
C. Visualization of Spectral Scan Results

In Figs. 3a and 3b, we present visualizations of the *received power (dBm)* plotted against *frequency*. Figure 3a shows that when the spectral scan is conducted in an isolation chamber, the highest received signal power is 1 dBm. In an office environment, due to the router's distance from the receiver, moderate ambient RF is observed with a maximum received power of 4 dBm. In a laboratory setting with high RF ambiance near the router, a high-intensity received power of 10 dBm is observed.

The impact of jamming on RF spectrum is presented in Fig. 3b. The graph shows the relationship between the *received power (dBm)* and *frequency* when different jamming configurations are in place. When the jamming is performed at 5200 MHz with a jamming power of 10 dBm and a distance of 20 cm between the jammer and the receiver, it is evident that the received power reaches around 30 dBm at 5200 MHz. However, when the jammed frequency is set to 5280 MHz, the jammer is positioned 60 cm away from the receiver, and the jamming power is decreased to 0 dBm, the received power reaches a maximum of approximately 0 dBm at 5280 MHz. The figure also highlights that the jamming effects can be seen



(a) Received power for the normal environments.



(b) Received power for two instances of jamming power/distance. The blue boxes represent jamming occurring at 5200MHz, while the red represent jamming occurring at 5280MHz.

Fig. 3: Visualization of received power (dBM) over frequency for jamming and normal (non-jammed) scenarios in Unlicensed National Information Infrastructure (U-NII)-1 and U-NII-2 portions of the spectrum.

at surrounding frequencies and decreases as the distance from the jammed frequency increases.

D. Categorization of Dataset

This section presents the categorization of the dataset accompanying the article. Based on the type of measurement, such as device type, device bandwidth, and spectral scan method, experiments from approximately 30 different configurations have been selected and grouped into four categories as described in Section II. The dataset comprises five sub-directories, each named according to three parameters in the format of spectral_scans_A_B_C, where A represents the device type (either QCA9880 or doodlelabs), B represents the scan bandwidth (either ht20, ht40, or vht80), and C represents the mode of scan (background, chanscan, or manual).

Each sub-directory contains over a thousand samples, with filenames in the format of samples_A_B_C_D_E, where A represents the environment in which the data was collected (either chamber, lab, or office), B represents the jammed frequency, C represents the distance between the jammer and receiver (either 20 cm, 40 cm, or 60 cm), D represents the jammer transmit power (0 dBm, 5 dBm, or 10 dBm), and E represents the transmission number, starting from 1 and indicating the temporal order of the transmissions. For example, the file "samples_chamber_2412MHz_40cm_5dbm_3.bin" indicates the third transmission with a jamming power of 5 dBm, a distance of 40 cm between the jammer and receiver, a jammed frequency of 2412 MHz, and data collected in an RF isolation chamber. For each configuration, ten transmissions were conducted.

IV. Jamming Detection

In this section, we present a machine learning-based approach for determining the exposure of a transmitter and a receiver to RF jamming attacks. This jamming detection problem is framed as a binary classification task, with samples classified as either normal or jamming. Normal samples are acquired from laboratory, office, and isolation chamber environments without jamming, while jamming samples are collected from the isolation chamber with the JamRF toolkit turned on. To achieve high detection accuracy, it is crucial to carefully consider various aspects, such as the selection of appropriate input features, measurement and collection of data, generation of a large dataset, and application of efficient algorithms for training, validation, and testing of the model. In this paper, we evaluate the performance of five different classifiers. Although our approach to the problem is supervised, it may be valuable to investigate the application of unsupervised anomaly detection approaches, such as ARCADE [12]. This possibility will be the focus of future research efforts.

The five classifiers evaluated in this paper were Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), RAndom Forest (RAF), eXtreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM). MLP is a feedforward neural network with at least three node layers, including an input layer, hidden layer, and output layer, and utilizes the supervised learning approach of backpropagation. SVM is a supervised learning model that classifies fresh samples into two categories through a non-probabilistic binary linear classifier and is capable of performing non-linear classification via the kernel trick. RF is an ensemble learning approach that utilizes many decision trees for classification, regression, and other tasks and has improved performance but decreased interpretability compared to a single decision tree. XGBoost is an optimized gradient-boosted decision tree solution that prioritizes speed and performance and often outperforms a single decision tree in terms of accuracy while compromising interpretability. LightGBM is a scalable, distributed gradient boosting system that supports multiple algorithms, including RAF, and differs in tree construction compared to XGBoost.

A. Pre-processing

Here we describe the steps to prepare the data before training the classifiers. Firstly, the feature "freq" was dropped as it is deemed irrelevant to the task of jamming detection on a particular (isolated channel). Additionally, due to missing and noisy data, the feature "rcvpwr_dBm" was also dropped. The time series data for each channel was then transformed into a single row by applying seven descriptive statistics, including minimum, maximum, mean, standard deviation, 75th percentile, 50th percentile, and 25th percentile, resulting in 49 features. Finally, the data were separated into two groups; one consisting of jamming data and the other consisting of data from isolated chambers (low interference), offices (moderate interference), and laboratories (high interference). These groups were used to train a binary classifier. However, the four independent classes were kept separate for the purpose of training a multi-class classifier.

TABLE I: Jamming detection performance in binary classification with normal vs. jamming, where normal class comprises low, moderate, and high interference samples. The results are in the format *mean* (± *std.*) obtained over 10-folds.

Algorithm	Number of Samples	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Inference Speed (KHz)
MLP	629	94.33 (±0.47)	96.00 (±0.00)	95.67 (±0.94)	97.00 (±0.00)	414.52 ± (21.0)
SVM	629	99.0 (±0.00)	99.00 (±0.00)	99.00 (±0.00)	100.00 (±0.00)	107.68 ± (4.30)
RAF	629	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)	54.95 ± (5.33)
XGBoost	629	98.67 (±1.25)	99.67 (±0.47)	99.00 (±0.00)	99.67 (±0.47)	277.97 ± (37.4)
LGBM	629	99.00 (±0.00)	100.00 (±0.00)	99.00 (±0.00)	99.67 (±0.47)	409.60 ± (4.60)

TABLE II: Performance comparison of jamming detection for multi-class classification. The results are present in the format of *mean* (± *std.*) obtained from 10-folds.

Algorithm	Interference Type	Number of Samples	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Inference Speed (KHz)
MLP	Low Interference	43	84.00 (±11.3)	56.67 (±11.8)	65.67 (±6.13)	90.33 (±0.94)	382.63 (±10.9)
	Moderate Interference	46	67.67 (±1.89)	88.67 (±6.13)	77.33 (±3.77)		
	High Interference	51	96.67 (±2.36)	37.67 (±4.71)	56.67 (±4.71)		
	Jamming	489	93.00 (±1.41)	98.67 (±0.94)	95.67 (±0.47)		
	Macro Average	629	82.67 (±8.26)	67.00 (±3.27)	70.67 (±3.77)		
SVM	Low Interference	43	89.00 (±0.00)	95.00 (±0.00)	92.00 (±0.00)	98.00 (±0.00)	32.82 (±0.96)
	Moderate Interference	46	93.00 (±0.00)	91.00 (±0.00)	92.00 (±0.00)		
	High Interference	51	96.00 (±0.00)	94.00 (±0.00)	95.00 (±0.00)		
	Jamming	489	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)		
	Macro Average	629	95.00 (±0.00)	95.00 (±0.00)	95.00 (±0.00)		
RAF	Low Interference	43	93.67 (±0.94)	97.00 (±11.8)	95.33 (±1.24)	99.00 (±0.00)	44.08 (±11.6)
	Moderate Interference	46	95.33 (±0.94)	96.00 (±0.00)	95.67 (±0.47)		
	High Interference	51	98.00 (±0.00)	94.67 (±0.94)	96.33 (±0.47)		
	Jamming	489	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)		
	Macro Average	629	96.33 (±0.47)	96.33 (±0.47)	96.33 (±0.47)		
XGBoost	Low Interference	43	93.00 (±1.64)	94.33 (±0.94)	93.67 (±0.94)	98.67 (±0.47)	53.09 (±3.73)
	Moderate Interference	46	93.33 (±2.49)	94.00 (±1.41)	93.67 (±1.89)		
	High Interference	51	98.00 (±0.00)	95.33 (±3.40)	96.67 (±1.70)		
	Jamming	489	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)		
	Macro Average	629	95.33 (±0.47)	95.67 (±0.47)	95.33 (±0.47)		
LGBM	Low Interference	43	91.00 (±2.83)	95.00 (±0.00)	93.00 (±1.41)	98.67 (±0.47)	265.43 (±38.1)
	Moderate Interference	46	92.00 (±2.83)	96.00 (±0.00)	94.00 (±1.41)		
	High Interference	51	98.00 (±0.00)	93.33 (±4.11)	95.67 (±2.05)		
	Jamming	489	100.00 (±0.00)	100.00 (±0.00)	100.00 (±0.00)		
	Macro Average	629	95.67 (±1.25)	96.33 (±0.94)	95.67 (±1.25)		

-

B. Training and Tuning

The ML-based classification algorithms are trained and evaluated using a measured dataset. After the pre-processing step, the training split of the processed dataset is utilized to train and fit the models. The performance of the models is then tested and presented using the testing dataset, which contains normal/interference and jamming data. To achieve high performance, a random search hyper-parameter tuning technique with 10-fold cross-validation is employed.

The hyperparameters of the MLP classifier are optimized to obtain the optimum model. In the case of binary classification, the best hyperparameters include Adam solver, an initial learning rate of 0.0001, a batch size of 128, l_2 regularization factor of 0.001, a maximum iteration of 300, and two hidden layers with thirty and fifteen units respectively. For multi-class classification, the solver, initial learning rate, and maximum iteration are similar to those for binary classification. However, the batch size is 128, the l_2 regularization factor is 0.0001, and there are two hidden layers with 30 and 15 units, respectively. In the SVM classifier, three major hyperparameters must be tuned for optimal performance: kernel, C, and γ. The optimal hyperparameters for binary classification were an RBF kernel, $C = 20$, and $\gamma = 0.0001$. For multi-class classification, the kernel is also RBF with $C = 35$ and $\gamma = 0.001$. For the random forest classifier, six hyperparameters were adjusted. The ideal hyperparameters for binary classification include 200 estimators, minimum samples leaf of 1, maximum depth of 5, minimum samples split of 2, maximum features set to $\sqrt{n}$ where n is the number of features, and bootstrap set to false. In the case of multi-class classification, the maximum features and bootstrap values are identical to those for binary classification. However, the number of estimators is 100, the minimum leaf samples is 2, the maximum depth samples is 30, and the minimum split samples is 4.

C. Results and Discussions

Table I compares the performance of different machine learning algorithms (MLP, SVM, RAF, XGboost, and LGBM)

for binary-class classification of jamming. The evaluation metrics used are precision, recall, F1-score, accuracy, and inference speed (KHz). To handle imbalanced datasets, it is important to calculate these metrics separately for each class. This allows for a more nuanced understanding of the performance of the algorithms, as the imbalance in the dataset can affect the overall accuracy and make it misleading. Moreover, although in our data, the majority class is jamming, but in reality it is the minority. This means that it's crucial to classify jamming activity correctly, as classifying it as normal interference has serious consequences. The precision, recall, and F1-score metrics provide a more complete picture of the performance of the algorithms, highlighting the trade-off between correctly classifying the samples of each class and the number of false positive or false negative predictions. For binary-class classification, the results showed that all algorithms performed well, with precision and recall ranging from 94% to 100% and F1-scores ranging from 95% to 100% for all classes. In terms of accuracy, RAF and SVM exhibit the highest performance achieving an accuracy of 100%. On the other hand, Table II compares the performance of the same machine learning algorithms for multi-class classification of jamming and low, moderate, and high interference. The results showed that RAF performed best with a 96.33% F1-score and an accuracy of 99%. MLP showed a lower performance, with F1-scores ranging from 56% to 96% and an accuracy of 90.33%.

In addition to accuracy, the speed of inference has also been studied. To this end, over 2000 samples from the dataset were generated and the average inference speed per second was calculated. When the algorithms were run on a 32GB RAM CPU with a dual-core process Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz, the MLP classifier was the fastest among the five classifiers, with an average inference speed of 414.52 KHz, and 382.63 KHz for the binary and multi-classification scenarios respectively. The inference speed of the algorithms varied similarly to the binary-class results, with MLP being the fastest and RAF the slowest. In conclusion, all algorithms tested showed good performance in jamming detection for both binary and multi-class classifications, with RAF showing the best results in terms of accuracy and F1-score. The inference speed of the algorithms also varied, with MLP being the fastest. Overall, the LGBM offers the best trade-off between accuracy and speed for both binary and multi-class classification scenarios.

V. Conclusions

In this study, we described the design and implementation of a radio frequency jamming detection testbed using the Wireless Spectral Scan dataset. The testbed design encompasses various technical and practical considerations, and we have provided a detailed overview of these considerations in the article. Furthermore, we presented a set of experimental results and analyzed the performance of five machine learning-based classifiers for jamming detection. The results indicate that the random forest classifier offers high accuracy in detecting jamming attacks, making it a promising approach for anti-jamming techniques. The findings of this study have important implications for the research community. Firstly, the presented dataset and results can inspire further research in developing new anti-jamming techniques. Secondly, the practical insights gained from the implementation of the testbed can assist in the development of new radio frequency jamming datasets and generation testbeds for future applications. In future work, we plan to demonstrate additional use cases for the dataset, including deep anomaly detection and deep reinforcement learning methods. These efforts are aimed at advancing state of the art in jamming detection and mitigating the negative impact of jamming attacks in radio frequency communication systems.

References

[1] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surv. Tutor.*, vol. 13, no. 2, pp. 245–257, 2011.

[2] S. D. Amuru and R. M. Buehrer, "Optimal jamming against digital modulation," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 10, pp. 2212–2224, 2015.

[3] A. S. Ali, M. Baddeley, L. Bariah, M. A. Lopez, W. T. Lunardi, J.-P. Giacalone, and S. Muhaidat, "Jamrf: Performance analysis, evaluation, and implementation of rf jamming over wi-fi," *IEEE Access*, vol. 10, pp. 133 370–133 384, 2022.

[4] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "CRAWDAD dataset uportorwthaachen/vanetjamming2012 (v. 2014-05-12)," Downloaded from https://crawdad.org/uportorwthaachen/vanetjamming2012/20140512, May 2014.

[5] J. Whelan, T. Sangarapillai, O. Minawi, A. Almehmadi, and K. El-Khatib, "Uav attack dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/00dg-0d12

[6] Google, "Spectrum Sharing," google.com. https://www.google.com/get/spectrumdatabase/ (accessed Oct. 10, 2022).

[7] IBM, "Software Defined Edge Processing," ibm.com. https://www.ibm.com/docs/en/eam/4.4?topic=examples-software-defined-radio-edge-processing (accessed Oct. 10, 2022).

[8] M. Zheleva, R. Chandra, A. Chowdhery, P. Garnett, A. Gupta, A. Kapoor, and M. Valerio, "Enabling a nationwide radio frequency inventory using the spectrum observatory," *IEEE Trans. Mob. Comput.*, vol. 17, February 2018.

[9] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van Den Bergh, H. Cordobes, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and Big Spectrum Data," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 210–217, 2018.

[10] S. Rajendran, V. Lenders, W. Meert, and S. Pollin, "Crowdsourced Wireless Spectrum Anomaly Detection," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 694–703, 2020.

[11] M. Ossmann, "HackRF One," greatscottgadgets.com. https://greatscottgadgets.com/hackrf/ (accessed Oct. 10, 2022).

[12] W. T. Lunardi, M. A. Lopez, and J.-P. Giacalone, "Arcade: Adversarially regularized convolutional autoencoder for network anomaly detection," *IEEE Trans. Netw. Serv. Manag.*, pp. 1–1, 2022.

Abubakar S. Ali received the B.S. degree in electrical engineering from Bayero University Kano, Kano, Nigeria, in 2014 and the M.S. degree in communications and signal processing from University of Leeds, Leeds, UK, in 2015. From 2018 to 2019, he was a Lecturer with the Department of Electrical Engineering, Bayero University Kano, Kano, Nigeria. He is currently pursuing the Ph.D. degree in electrical and computer engineering at Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include low power wireless communications, machine learning, artificial intelligence and optimization for communications and networking, and security in intelligent communications and networking.

Govind Sing is a Lead Engineer with the Secure Systems Research Centre (SSRC) at the Technology Innovation Institute (TII) in Abu Dhabi, UAE. Govind is a conversant embedded professional having 16 year of industry experience in the area of embedded software development in the area of mobile computing , IOT devices. He has research interest in wireless system design and development in different computing platform.

Willian Tessaro Lunardi is a Senior Researcher at the Secure Systems Research Centre, Technology Innovation Institute, Abu Dhabi, UAE. He has a Ph.D. in computer science from the University of Luxembourg. His main area of research is machine learning and combinatorial optimization. He is currently working on machine learning for anomaly detection, network security, physical layer security, and jamming detection. He has published over 25 research papers in scientific international journals, conferences, and book chapters.

Lina Bariah (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in communications engineering from Khalifa University, Abu Dhabi, UAE, in 2015 and 2018, respectively. She was a Visiting Researcher with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, in 2019, and an affiliate research fellow, James Watt School of Engineering, University of Glasgow, UK. She is currently a Senior Researcher at the technology Innovation institute, a visiting research scientist at Khalifa University, and an affiliate researcher in the University at Albany, USA.

Michael Baddeley received the MEng in Computer and Electronic Systems from the University of Strathclyde, Glasgow, UK, in 2010 and the PhD in "Software Defined Networking for the Industrial Internet of Things" from the University of Bristol, UK, in 2020. He is currently a Lead Wireless Researcher with the Secure Systems Research Centre (SSRC) at the Technology Innovation Institute (TII) in Abu Dhabi, UAE. His current research focuses on robust wireless communication for ad-hoc and infrastructure-less mesh networks: such as interference mitigation, spectral coexistence, and end-to-end dependability across a mesh-cloud continuum.

Martin Andreoni Lopez is a Network Security Researcher at the Secure System Research Center of the Technology Innovation Institute in Abu Dhabi, United Arab Emirates. He was a Researcher at Samsung R&D Institute Brazil. Graduated as an Electronic Engineer from the Universidad Nacional de San Juan (UNSJ), Argentina in 2011. Master in Electrical Engineering from the Federal University of Rio de Janeiro (COPPE / UFRJ), in 2014. He has a Ph.D. from the Federal University of Rio de Janeiro (COPPE / UFRJ) in Teleinformatics and Automation Group (GTA) and by Sorbonne University in the Phare team of Laboratoire d'Informatique Paris VI (LIP6), France, in 2018.

Jean-Pierre Giacalone is Vice President of Secure Communications Engineering at Secure Systems Research Centre, Technology Innovation Institute, Abu Dhabi, UAE. He is responsible for carrying out research on secure communications, with a focus on improving resilience of cyber-physical and autonomous systems. He has worked as an expert in software architecture for advanced driving assistance systems at Renault, and as principal engineer and architect within the mobile systems technologies group at Intel. He has an engineering degree from the from the Ecole nationale sup ´ erieure d' ´ electrotechnique, d' ´ electronique, ´ d'informatique, d'hydraulique et des tel ´ ecommunications (ENSEEIHT) in ´ Toulouse, France.

Sami Muhaidat. (Senior Member, IEEE) received the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, Ontario, in 2006. From 2007 to 2008, he was an NSERC postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Toronto, Canada. From 2008 to 2012, he was an Assistant Professor in the School of Engineering Science, Simon Fraser University, BC, Canada. He is currently a Professor at Khalifa University, and an Adjunct Professor with the Department of Systems and Computer Engineering, Carleton University, Canada. His research focuses on wireless communications, optical communications, IoT with emphasis on battery-less devices, and machine learning.