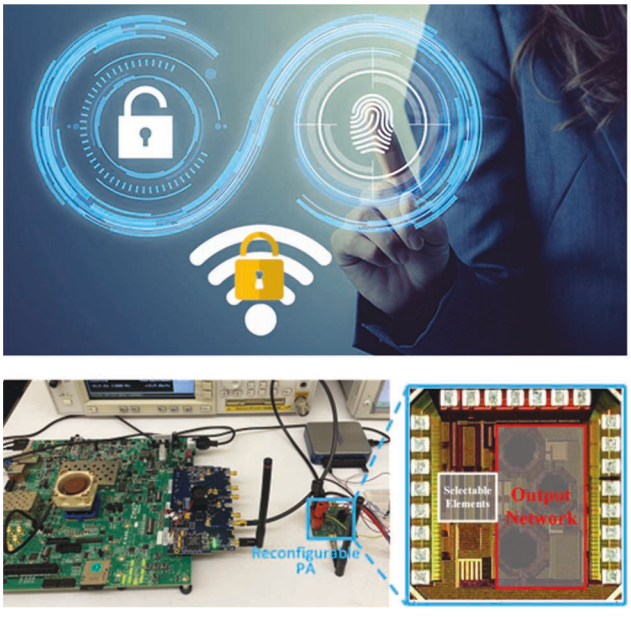
Vanessa Chen, Jiachen Xu, Yuyi Shen, and Ethan Chen

RF Fingerprint Classification With Combinatorial-Randomness-Based Power Amplifiers and Convolutional Neural Networks

Secure analog/ RF electronics and electromagnetics



TOP FIGURE: HTTPS://WWW.ECE.CMU.EDU/NEWS-AND-EVENTS/STORY/2022/02/FINGERPRINTING-THE-IOT.HTML

The growth of the IoT requires more comprehensive security measures than ever. RF fingerprinting (RFF) utilizes features in the signals and waveforms from transmitters' physical-layer imperfections to classify and authenticate devices. To prevent attacks from impersonators, combinatorial randomness is exploited to augment the RF fingerprints with a high-efficiency PA for IoT applications. By enabling different subsets of thinly sliced PA elements, the transmitter can be reconfigured with 220 subsets that exhibit distinctive RF fingerprints for signal analysis at the edge. In this work, a combinatorial-randomness-based PA was implemented in a BLE system. The BLE packets' in-phase and quadrature samples transmitted from each configuration are collected with different SNRs to emulate the

Digital Object Identifier 10.1109/MSSC.2022.3200302
Date of current version: 11 November 2022

1943-0582/22©2022IEEE

environmental changes in communication channels. A lightweight convolutional neural network (CNN) classifier demonstrates the possibility of accurate and fast inference of unique features in the IoT environment, which our approach exploits to enable on-chip time-varying RF fingerprints.

The State of the Art

With the growth of the IoT, network security is becoming an increasingly significant concern. In addition to the ever-higher stakes at play due to the increasing integration of the IoT into critical infrastructure, industrial systems, and health-care devices via sensors, power meters, and so on [1], the expansive nature of the IoT makes it possible for attackers to collect large amounts of information. This information can be used for the purpose of defeating security protocols, enabling attackers to observe or guess at security parameters used in conventional authentication protocols, such as Wi-Fi Protected Access [2]. This points to a need for security protocols that serve to authenticate actual devices within a network, as opposed to the data being transmitted from them.

RFF has been investigated as a software-based device authentication mechanism that may be implemented without requiring the redeployment of physical IoT devices with better security systems [3], [4], [5], [6]. RF fingerprints consist of signal features imprinted upon a radio's transmit waveform by its inherent hardware characteristics [e.g., PA nonlinearity, carrier frequency offset, in-phase/quadrature (I/Q) imbalance, and so on] [6] and can serve as unique physical signatures for their associated radio when extracted. Approaches that require some form of data preprocessing, such as a Gabor transform (GT) or fast Fourier transform to explicitly extract RF fingerprints prior to feeding them into a machine-learning (ML) classifier [3], [4], [6], and approaches that simply feed raw I/Q samples to an ML block for directly distinguishing radios [5] have both been reported in the literature.

Many works have contributed to the RFF literature. These include a CNN-based RFF system tested on a dataset of IEEE 802.11a frames collected from Universal Software Radio Peripheral SDR transmitters [7], a drone detection and identification system using a database of RF signals collected from drones [8], and a database of recorded Bluetooth signals for testing RFF methods [9]. In these studies, the hardware impairments associated with each recorded radio that yield RF fingerprints remain fixed over time.

However, a mixture of security and user capacity concerns motivates the study of configurable and time-varying RFF systems. For instance, it is possible for attackers to replay signals from legitimate radios to impersonate trusted devices and thus penetrate the RFF authentication [10]. The introduction of a time-varying aspect to an RFF system would add another dimension of complexity to the measures required to successfully engage in such a replay attack. Furthermore, the configurability could be applied to enhance the variability of RF fingerprints within the system and thus improve user capacity, such as in the case of [11].

This work presents an ML-assisted RF fingerprint classification with a transmit-side reconfigurable PA for BLE IoT applications, as illustrated in Figure 1. PA configurability is achieved by selecting combinations of sliced PA transistor elements, altering the RF fingerprints imparted upon the transmitted signal in correspondence with the transistor parameters associated with the selected elements. Data recordings are included for different SNRs to facilitate the analysis of the impact of noise on RFF system performance.

The recorded data represent a far greater assortment of distinct hardware impairment-induced RF fingerprints than those presented in the datasets published in [7], [8], and [9], the largest of which encompassed oscilloscope samples from 86 different Bluetooth-enabled smartphones. Measurements were conducted across 220 PA configurations, each producing a distinct RF fingerprint. Furthermore, as a result of the RF fingerprints originating from within the same device, our dataset represents a more challenging classification task and can serve as a stricter validation tool for RFF systems. Unlike the works of [7] and [8], a direct RF sampling transceiver with digital downconversion (DDC) to minimize the impact of receiver impairments has been investigated, so the impact of custom receiver nonidealities using

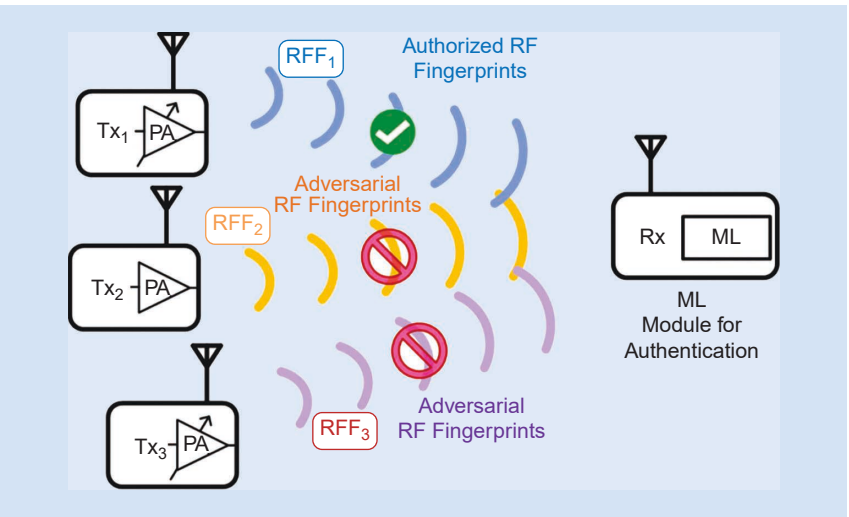


FIGURE 1: The RF fingerprint classification with the ML module for authentication. The RF fingerprints are augmented with a configurable PA in the transmitter.

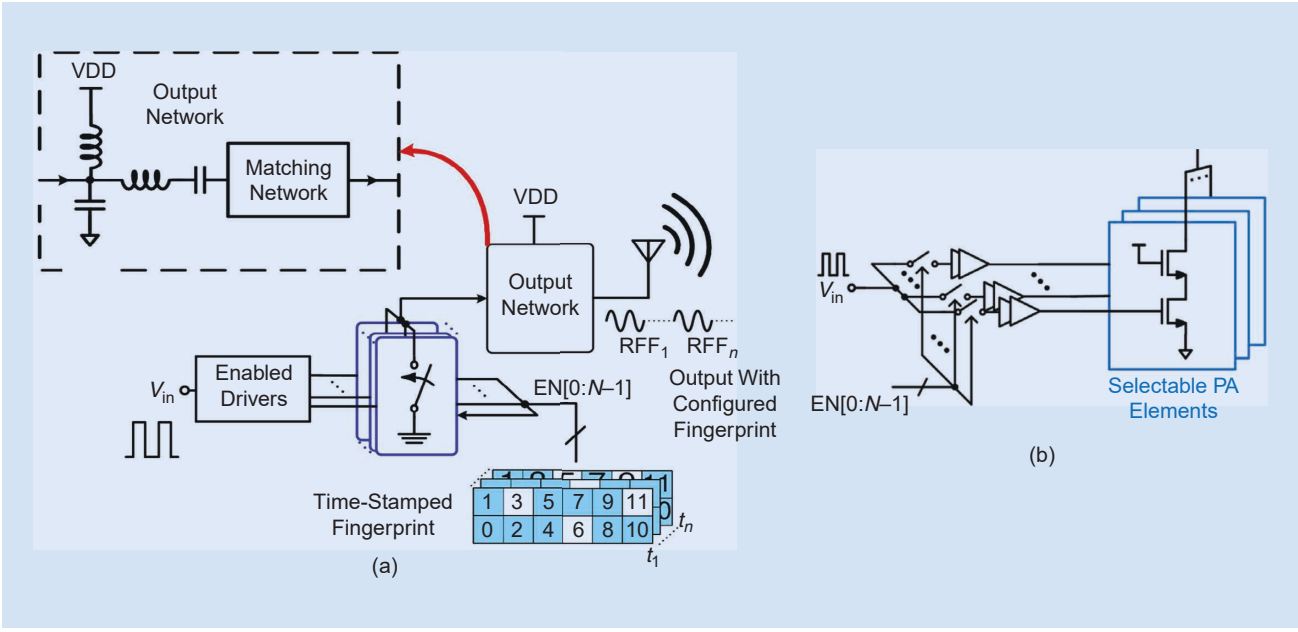


FIGURE 2: (a) The high-level structure of the reconfigurable PA and (b) the sliced transistor elements that are used to imprint configurable RF fingerprints on BLE packets.

baseband models can be evaluated. The work was validated using a lightweight CNN classifier, which demonstrates the preservation of relevant RF fingerprint features within it.

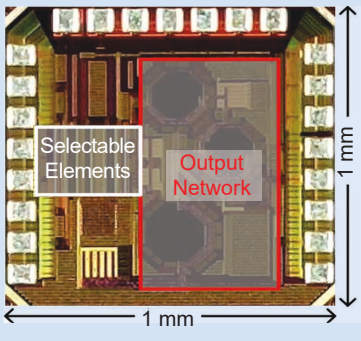
The following section describes the structure of the reconfigurable PA and the measurement setup used for collecting the dataset. The section "Recorded Data Analysis" analyzes the recorded signals, and the section "CNNs for PA Configuration Classification" presents the implementation of the CNN classifier. The section "Environmental Changes" discusses the impacts of the environmental changes, and the final section concludes the article.

Reconfigurable PA and Dataset Collection

Reconfigurable PA

The high-level structure of the reconfigurable PA used for this study is illustrated in Figure 2(a). The primary PA transistor is sliced into several parallel devices that may be enabled independently of one another through their driver circuitry, as shown in Figure 2(b). Each transistor is operated as a switch and is connected to a single Class E PA-style output network that blocks higher order harmonic components from reaching the output while shaping the transistor current and voltage waveforms to lower power dissipation. The total off-capacitance present at the output network as a result of the transistor parasitics remains constant irrespective of PA configuration, permitting configuration-independent operation.

At the time of fabrication, each selectable slice's transistor parameters (V_{TH}, β) are affected by random process variations in accordance with Pelgrom's law [11]. Because these transistor parameters ultimately give rise to the PA's RF fingerprint, selecting different combinations of PA device slices results in distinct RF fingerprints. A design with 12 total selectable slices was designed for a frequency of 2.4 GHz and taped out. A die photo of a fabricated chip sample is shown in Figure 3. To maintain a balance of output power and configurability, it was decided to enable nine slices at a time to allow a combinatorial search space of C (12,9) = 220 possible PA configurations.



Technology	65-nm CMOS
Chip Area	1 mm^2
Core Area	0.267 mm^2
Set Number of Choosing 9 From 12	220

FIGURE 3: A die photo of the reconfigurable PA used to collect the data.

Bluetooth Low-Energy Packet Parameters

A physical-layer waveform corresponding to a nonconnectable advertising packet, such as those used for transporting BLE Mesh protocol data units, was generated using

Matlab's Communications Toolbox and repeatedly transmitted through the PA test chip using the AD9082 RF transceiver board. Each BLE packet recorded in the presented dataset was transmitted over BLE channel 37 (2,402 MHz) using the LE1M physical-layer mode.

Measurement Setup and Data Processing

The dataset was captured using an AD9082-FMCA-EBZ direct RF sampling transceiver board as a receiver, with the onboard RF DAC used to drive the reconfigurable PA. Baseband I/Q samples were directly fetched at a sampling rate of 250 MSPS from the AD9082-FMCA-EBZ using the Analog Devices High Speed Converter Toolbox for Matlab, after being sampled at 6 GSPS by the RF ADC and passing through the on-chip digital downconverter and decimation filters. These were further decimated to a sampling rate of 25 MSPS in Matlab to lower the required storage space. A USB-6001 NI DAQ was used to generate the control signals necessary to switch the PA between configurations. Sufficient samples were collected for each PA configuration to contain more than 1,000 received BLE packets. An image of the measurement setup is shown in Figure 4(a). Figure 4(b) illustrates the connections of the different components used in the measurement setup. The power spectra of the PA output while transmitting the BLE packets are plotted in Figure 5 across 220 PA configurations. It can be seen from the figure that there is a ±2-dB variation across all 220 configurations.

The I/Q samples corresponding to actual BLE packets were extracted from the fetched data using cross correlation with the known BLE preamble sequence to determine the starting indices of the packet. Afterward, demodulation was performed to verify the integrity of each packet in the dataset.

Recorded Data Analysis

Recordings were captured of the transmitted BLE packets across all 220 PA configurations using the setup described in the section "Reconfigurable PA and Dataset Collection." Enough I/Q samples were taken to collect more than 1,200 packets for each PA configuration at a baseline SNR of 35 dB. Later, an attenuator in the measurement setup was used to lower the signal power, and sufficient samples were taken to collect more

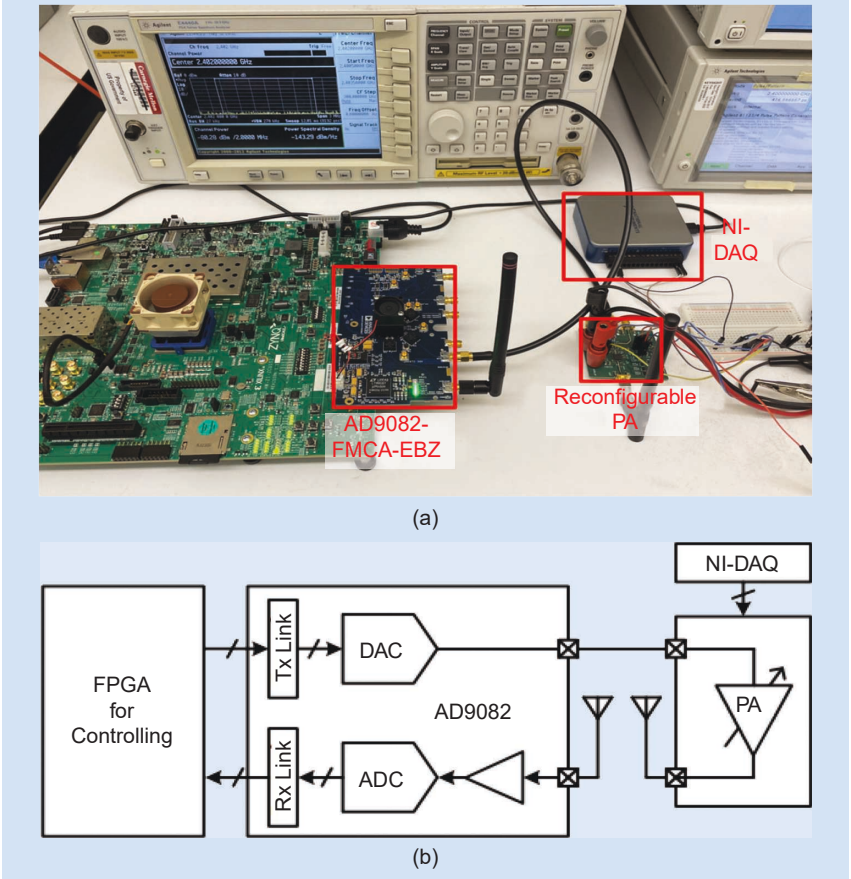


FIGURE 4: (a) The AD9082-FMCA-EBZ board was used as a direct conversion receiver with DDC and as a transmitter to drive the PA. An NI DAQ USB-6001 is used to generate the control signals for sweeping across the PA configurations. (b) The block diagram of the connections of the different components used in the measurement setup.

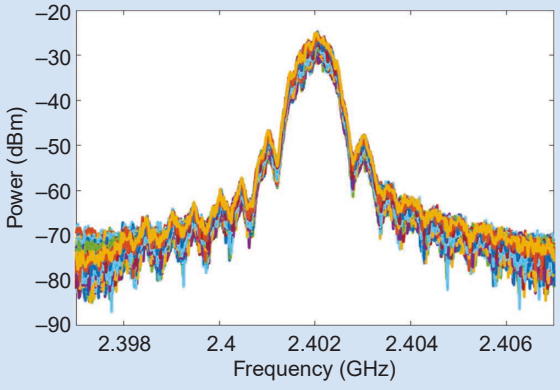


FIGURE 5: The power spectra of the PA output while transmitting the BLE packets were measured across PA configurations with a spectrum analyzer. The different colored lines are used to indicate the power spectrum for each of the 220 combinations.

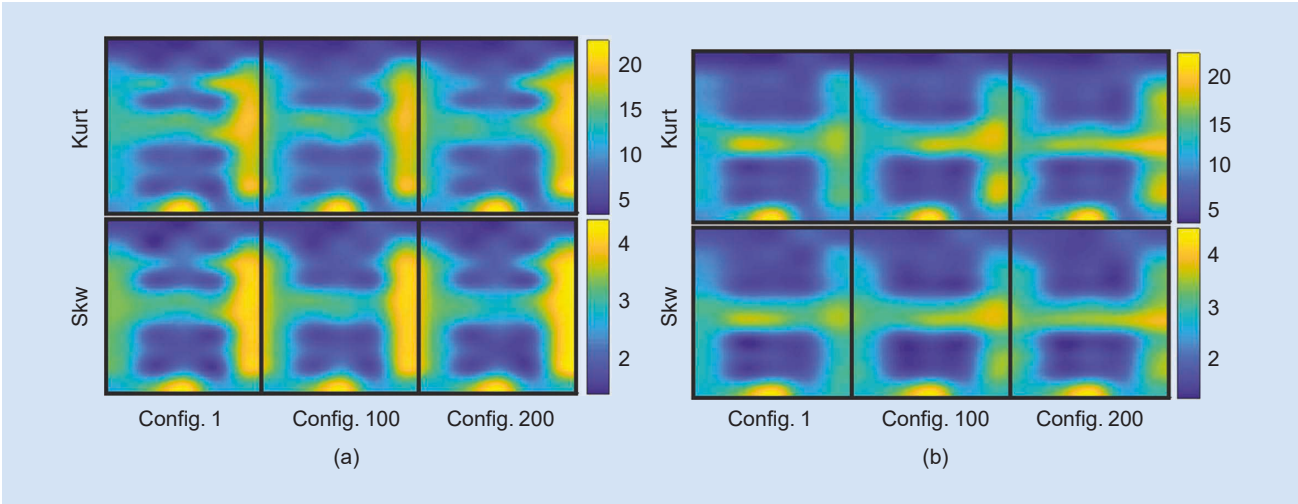


FIGURE 6: The average RF-DNA footprint for three separate PA configurations is visualized as an image for (a) the baseline SNR of 35 dB and (b) a moderate SNR of 15 dB. Config.: configuration.

than 500 packets per configuration at SNRs of 25 dB and 15 dB. It can be seen from the analysis of this section that the diminishing RF fingerprints at the low SNR scheme would lead to a more difficult classification problem.

RF Fingerprint Visualization

The distinct RF fingerprints produced by separate PA configurations may be visualized through the calculation of RF distinct native attribute (RF-DNA) fingerprints using a discrete Gabor transform (DGT), as in [4]. For these purposes, the first 1,000 I/Q points making up the fixed BLE preamble and access address of each captured packet were taken. The DGT was used to calculate the time-varying frequency content of these 1,000-point vectors for 100 frequency bins across 100 segments of the input vector, and the normalized squared magnitude of the resultant 100 × 100 matrix was calculated. These DGT settings were chosen to yield a large number of data points for computing the kurtosis (Kurt) and skewness (Skw) of $N_T = 20 \times N_F = 10$ patches taken from the DGT output.

The average RF-DNA fingerprint for three different PA configurations was computed across the corresponding recorded BLE packets for the baseline SNR of 35 dB and a moderate SNR of 15 dB. The resultant matrices were interpolated and visualized in Figure 6 using the color legend on the right. Visual differences between the fingerprints calculated for distinct PA configurations can be clearly seen, although some features disappear at the lower SNR setting.

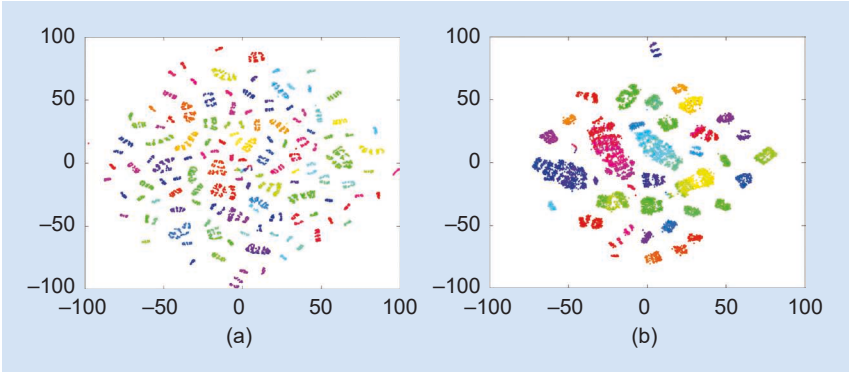


FIGURE 7: t-SNE was used to visualize the distribution of RFFs within the presented dataset for (a) the baseline SNR of 35 dB and (b) a moderate SNR of 15 dB.

Clustering Analysis

The full distribution of recorded RF fingerprints across PA configurations can be visualized through the usage of t-distributed stochastic neighbor embedding (t-SNE) to project the high-dimensional recorded data to two dimensions for plotting. Unlike principal component analysis, which tries to separate dissimilar data points, t-SNE seeks to group similar data points, making it more suited for visualizing some datasets [12]. After truncating the captured packets to the first 1,000 I/Q points to isolate the preambles and access addresses and interleaving the I and Q samples into the same vector, t-SNE was applied to cluster a set of data containing 60 randomly chosen BLE packets recorded from each PA configuration. The clustered low-dimensional results are plotted in Figure 7 for data recorded with SNRs of 35 and 15 dB. Although the impact of noise is visible through the reduced number of clusters in the plot corresponding to the lower SNR, RF fingerprint clusters corresponding to different PA configurations in the dataset are clearly present.

CNNs for PA Configuration Classification

A CNN model was utilized to classify up to 220 PA configurations from their transmitted packets'

raw I/Q data. The recorded packets were truncated to the first 1,000 I/Q points (40 bits) to isolate the fixed BLE preambles and access addresses. The recorded data were further processed to simulate the impact of quantization and receiver sampling rate on the RFF system. The lightweight CNN model was optimized with post-training quantization to speed up the inference on IoT edge devices, and the power usage for RFF classification was reported on a Raspberry Pi to estimate the energy and resource overhead that the RFF system brings.

Dataset Normalization

Two major receiver specifications considered in this example are the ADC sampling rate and the bit resolution. To demonstrate the required ADC sampling rate in the receiver for sufficient classification accuracy, we decimate the original raw data to sampling rates of 1, 2, 5, 10, and 15 MSPS. A higher sampling rate indicates a higher ADC speed requirement and a higher ML classifier input data length, which could preserve more RFF features from the signal with a cost of consuming more power. The bit resolution sets the precision of the digital domain representations of the analog signal and is also critical for determining the RFF features available to the classifier. In this case, the data are quantized to 6 bits, 8 bits, 10 bits, 12 bits, 14 bits, and 16 bits to test their impact on the classifier accuracy. Because the ADC sampling rate and bit resolution are directly related to the energy consumption and cost of radios, achieving high classification accuracy at a low sampling rate and low bit resolution is desirable for energy-efficient RFF usage.

TABLE 1. THE CNN ARCHITECTURE.

LAYER	OUTPUT DIMENSIONS
Input	2 × input size (In)
16 Ch 1 × 5 Conv, stride = 2	16 × In/2
16 Ch 1 × 5 Conv, stride = 2	16 × In/2
Fully connected	128
Fully connected	220 (for 220 PA configurations)

The bit resolution sets the precision of the digital domain representations of the analog signal and is also critical for determining the RFF features available to the classifier.

The data normalization is done by linearly scaling and shifting the raw packets:

$$S'[i] = \frac{\text{round}\left(\frac{S[i]}{\max(abs(S))} \cdot Q\right)}{Q},$$

where S is all of the recorded signals in the dataset, $S[i]$ is a packet in the recorded signal, $S'[i]$ is the corresponding normalized packet, and $Q = 2^N - 1$, which scales and shifts the data to a range of $[-1, 1]$ with a desired quantization level of N bits to become the input to the CNN.

CNNs for RFF Classification

The structure of the CNN model is shown in Table 1. The first 1,000 packets were selected to transmit from each PA configuration from the data recorded at the baseline SNR of 35 dB to form the CNN dataset with a total size of 220,000. The packets from each configuration were allocated with a ratio of 4:3:3 to form training, validation, and test sets, respectively. Real and imaginary samples are taken in by the CNN as separate channels. The length of the input vector N is 40*SPS (samples per BLE symbol), as a result of each packet being truncated to the 40-symbol-long preamble/access address sequence. The raw dataset is processed and normalized using the aforementioned bit resolutions and sampling rates. An Adam optimizer with a learning rate of 0.001 and binary cross-entropy loss function was chosen from our experimental trials for training the CNN model, with a batch size of 128. The training process was done with an Nvidia V100 GPU. Each training epoch took around 2.7–4 s, depending on the input vector length. After training for 300 epochs, the model parameters with the highest validation accuracy were selected to be tested with the test set, and the results are shown in Figure 8.

The classification accuracy approaches an asymptote for bit resolutions

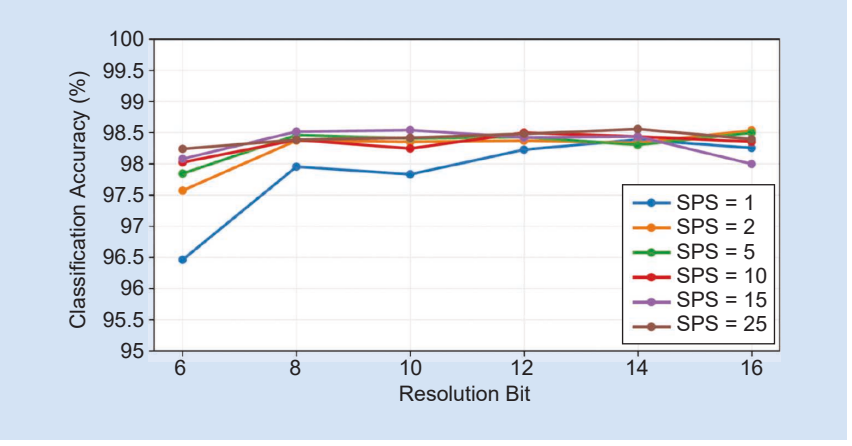


FIGURE 8: The CNN model's 220-configuration RFF classification accuracy with different receiver sampling rates and receiver bit resolutions.

of larger than 8, below which the sampling rate becomes more impactful on accuracy. An SPS of 5 and a bit resolution of 10 were used for the model to achieve 98.53% accuracy with 220 classes to carry on further analysis of the dataset for its moderate hardware requirements on the radio receiver.

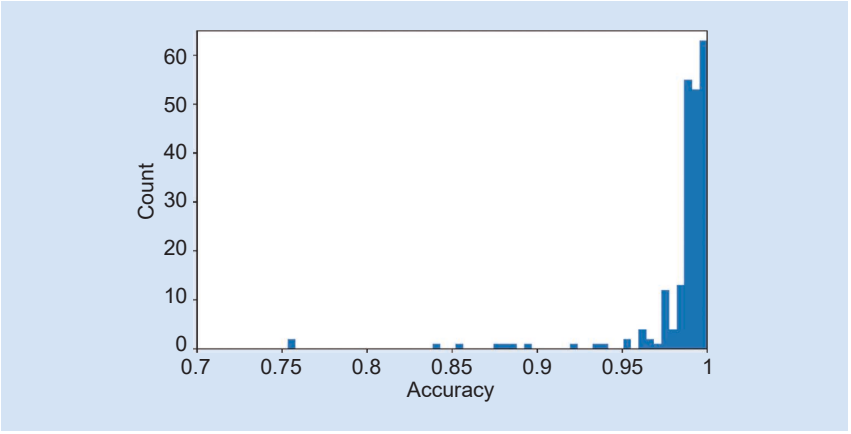


FIGURE 9: The distribution plot of the classification accuracies of 220 configurations with SPS = 5 and bit resolution = 10.

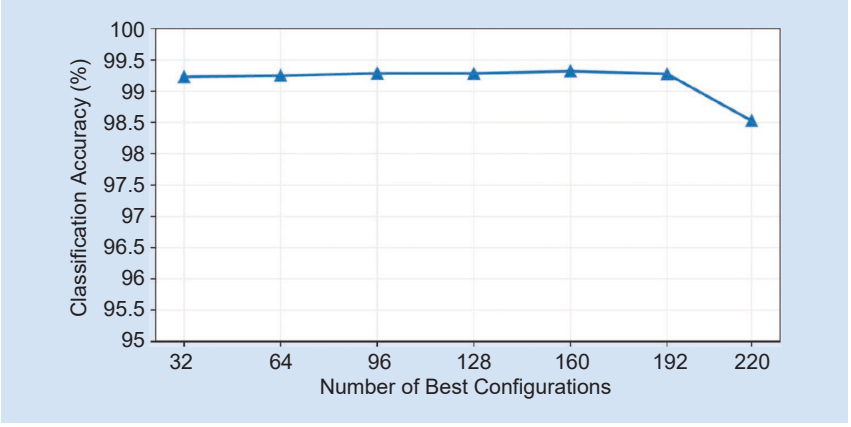


FIGURE 10: Testing the trained model's accuracy with only selected numbers of configurations in the test set. The configurations being used are selected using the validation set's confusion matrix.

TABLE 2. THE CNN PERFORMANCE ON A RASPBERRY PI 3B+.

MODEL TYPE	ACCURACY	MODEL SIZE	INFERENCE TIME	DYNAMIC POWER
FP16	98.53 %	267 KB	1.32 ms	0.21 W
INT8	95.09 %	138 KB	0.38 ms	0.21 W

TABLE 3. PERFORMANCE COMPARISON.

MODEL	TASK	ACCURACY	PLATFORM	DYNAMIC POWER
CNN	220-class RFF classification	95.09 %	Raspberry Pi	0.21 W
Bayesian neural network [14]	Six-class RFF classification	89.5%	Xilinx ZCU102	0.19 W
Feedforward neural network [15]	Six-class modulation classification	94 %	Xilinx XCZU9EG	0.5 W

The PA's RFF reconfigurability depends on the random process variations on the PA elements to form up to 220 RF fingerprints, and it is expected that some configurations in the PA would exhibit similar RF fingerprints, thus affecting the classifier's ability to distinguish among these configurations. By examining the confusion matrix, the less-distinct configurations could be identified and excluded from usage. Instead of attempting to visualize the large 220×220 confusion matrix, the distribution of the prediction accuracies on the datasets is plotted in Figure 9. This plot provides information about the quantity of the configurations that are exhibiting more/less distinguishable RF fingerprints. The trained models were tested with the test set using only a specific number of configurations, where the configurations are selected by ranking their accuracies in the validation set's confusion matrix, and the results are shown in Figure 10. The results show that the RF fingerprints exhibited by different PA configurations have varied performances. The classification accuracy can be improved while the number of configurations is reduced to keep the ones with better distinguishability.

To verify the overhead of deploying RFF to a real system, the performance of the CNN model was tested on a Raspberry Pi 3B+. The trained models were also processed with post-training quantization with TensorFlow Lite to compress the model and speed up the inference, and the specifications are shown in Table 2. As indicated in [13], the CNN model suffers from different levels of accuracy loss from the original model because of its small size. Because the small model is already adequately fast in floating point, trading off accuracy by fully quantizing the model may not be necessary, depending on the case. The power consumption on the Raspberry Pi was measured during idle state and inference. The dynamic power is used as a metric, which is calculated by subtracting the Raspberry Pi's idle power consumption

from the peak power consumption during inference to estimate the overhead brought to the system by classifying the RFFs. Table 3 compares the CNN model with other models implemented on FPGAs using a Bayesian neural network [14] and feedforward neural network [15]. The presented model can classify more classes with a Raspberry Pi.

The size of the training set was also explored to determine how it affects the classification accuracy to consider the case where only a limited number of training samples is available during the device's deployment. Instead of using 400 samples from each configuration in the training set, the number of training samples was limited to $N = 20, 40, 60, 80, 100, 200,$ and 300 from each configuration to observe the model's performance. The same validation set and test set (300 signals from each configuration) were used across these experiments. After the validation accuracy converged, the models' performance was tested on the test set, and the results are plotted in Figure 11. In general, a larger training set results in better accuracy, and the accuracy approaches an asymptote beyond $N = 100$.

Environmental Changes

To provide an example of validating the RFF classifier's performance under worse noise conditions with the presented results, the proposed CNN model was tested by drawing the data from packets recorded at lower SNRs. When the original classifier model, which trained with the baseline 35-dB-SNR dataset, was tested against the 25-dB-SNR and 15-dB-SNR noisy datasets, the classification accuracy was unsatisfactory and fell under 2% for all of the models with noisy signals.

For the model to be adequate for classifying noisy data, the training set needed to include noisy data as well. The CNN models were trained with evenly mixed data across the three available SNR levels (200 packets for each SNR from each configuration), and Figure 12 shows the classification accuracy for the signals at each SNR. The classifier showed 97.8% accuracy on the 35-dB data set, which is slightly lower than the accuracy of the classifier only trained with the 35-dB dataset (98.5%). However, the newly trained classifier has an accuracy of 92.7% and 88.4% on 25 dB and 15 dB, respectively, which is a much better performance compared to that of the original classifier, which had less than 2% accuracy on the noisy data. The result indicates that the fingerprint features are kept and can be identified even when data are noisy. However, it would be more desirable if the trained model could be robust to the incoming signals with features from environmental changes that are not seen during the training to build a more reliable RFF system. Thus, in this manner a dataset with different SNRs could be used to test the adaptability of edge computing models by selecting different mixtures of noise levels for training and testing.

Conclusion

In this work, we presented an ML-assisted RF fingerprint classification with a combinatorial PA featuring augmented RF fingerprints. The recorded dataset includes more than 1,200 BLE packets for each PA configuration across all 220 PA configurations at a baseline SNR of 35 dB. Moreover, more than 500 packets per configuration at SNRs of 25 dB and 15 dB were also collected to evaluate the impacts of environmental changes. The scheme of a single device with multi-RFFs is beneficial for IoT security with RFF through the multiple configurations. We also evaluated how a lightweight

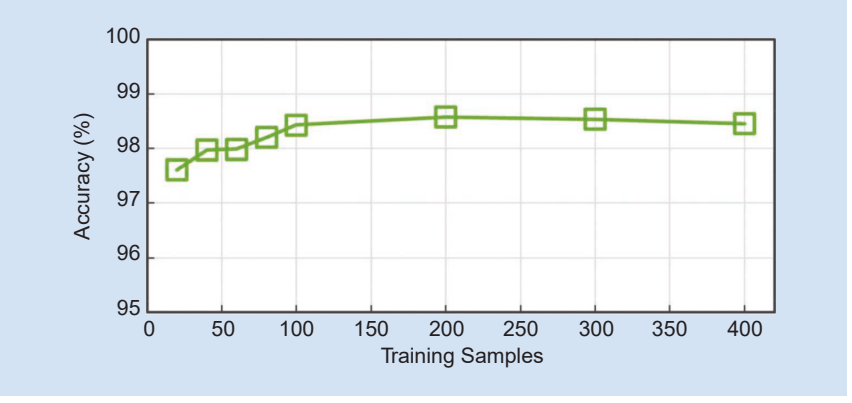


FIGURE 11: Classification versus training data size. N is the number of packets the training set had for each PA configuration.

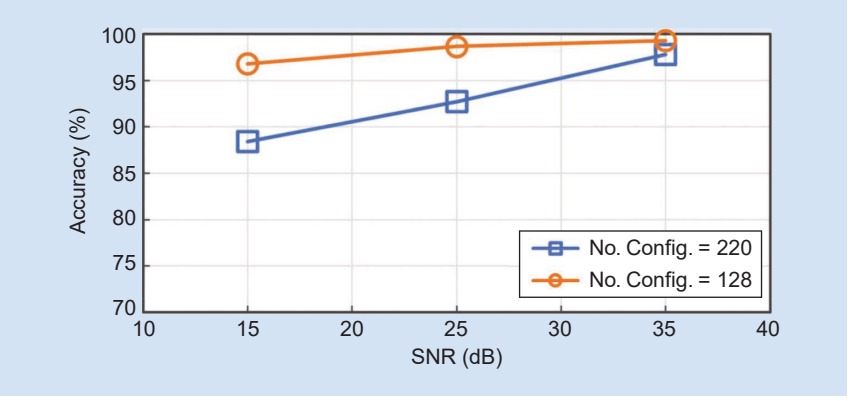


FIGURE 12: The figure of classification accuracy versus SNR shows cases for choosing both 128 and 220 configurations. The tested model was trained by data with SNRs of 15, 25, and 35 dB.

CNN model achieves different performances on the data with various system-level considerations, including receiver cost and noise analysis, which empowers the possibility of adding another layer of protection to wireless edge devices for secure IoT communication.

Acknowledgment

This work was supported by the National Science Foundation under Grants 1952907, 1953801, and 2028893.

References

[1] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, Apr. 2019, doi: 10.1109/COMST.2019.2910750.

[2] J. M. McGinthy, L. J. Wong, and A. J. Michaels, "Groundwork for neural network-based specific emitter identification authentication for IoT," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6429–6440, Aug. 2019, doi: 10.1109/JIOT.2019.2908759.

[3] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 349–360, Feb. 2019, doi: 10.1109/JIOT.2018.2838071.

[4] D. Reising, J. Cancelleri, T. D. Loveless, F. Kandah, and A. Skjellum, "Radio identity verification-based IoT security using RF-DNA fingerprints and SVM," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8356–8371, May 2021, doi: 10.1109/JIOT.2020.3045305.

[5] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 146–152, Sep. 2018, doi: 10.1109/MCOM.2018.1800153.

[6] W. Wu, S. Hu, D. Lin, and Z. Liu, "DSLN: Securing Internet of Things through RF fingerprint recognition in low-SNR settings," *IEEE Internet Things J.,* vol. 9, no. 5, pp. 3838–3849, Mar. 2021, doi: 10.1109/JIOT.2021.3100398.

[7] K. Sankhe et al., "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 165–178, Mar. 2020, doi: 10.1109/TCCN.2019.2949308.

[8] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Future Gener. Comput. Syst.*, vol. 100, pp. 86–97, Nov. 2019, doi: 10.1016/j.future.2019.05.007.

[9] E. Uzundurukan, Y. Dalveren, and A. Kara, "A database for the radio frequency fingerprinting of Bluetooth devices," *Data*, vol. 5, no. 2, p. 55, Jun. 2020, doi: 10.3390/data5020055.

[10] C. S. U. Rehman, K. W. Sowerby, and C. Coghill, "Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers," *J. Comput. Syst. Sci*, vol. 80, no. 3, pp. 591–601, 2014, doi: 10.1016/j.jcss.2013.06.013.

[11] S. Rajendran, Z. Sun, F. Lin, and K. Ren, "Injecting reliable radio frequency fingerprints using metasurface for the Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1896–1911, Dec. 2021, doi: 10.1109/TIFS.2020.3045318.

[12] K. Y. Wong and F.-l. Chung, "Visualizing time series data with temporal matching based t-SNE," in *Proc. 2019 Int. Joint Conf. Neural Netw. (IJCNN)*, pp. 1–8, doi: 10.1109/IJCNN.2019.8851847.

[13] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.

[14] J. Xu, Y. Shen, E. Chen, and V. Chen, "Bayesian neural networks for identification and classification of radio frequency transmitters using power amplifiers' nonlinearity signatures," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 457–471, Jul. 2021, doi: 10.1109/OJCAS.2021.3089499.

[15] S. Soltani, Y. E. Sagduyu, R. Hasan, K. Davaslioglu, H. Deng and T. Erpek, "Real-time and embedded deep learning on FPGA for RF signal classification," in *Proc. MILCOM 2019 - 2019 IEEE Military Commun. Conf. (MILCOM)*, pp. 1–6, doi: 10.1109/MILCOM47813.2019.9021098.

About the Authors

Vanessa Chen (vanessachen@cmu.edu) received her Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2013. She was with Qualcomm, San Diego, CA, USA, working on energy-efficient data-acquisition systems for mobile devices. From 2010 to 2013, she was with Carnegie Mellon, focusing her research on self-healing systems and high-speed ADCs, and she held a research internship position with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, in 2012. She was an assistant professor with The Ohio State University, Columbus, OH, USA. She is currently an assistant professor of electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA 15213 USA. Her research interests focus on data conversion interfaces for machine learning, RF/analog hardware security, ubiquitous sensing, and communication systems.

She was a recipient of the National Science Foundation CAREER Award in 2019, the Analog Devices Outstanding Student Designer Award in 2013, and an IBM Ph.D. Fellowship in 2012. She is also an associate editor of *IEEE Transactions on Biomedical Circuits and Systems* and *IEEE Open Journal of Circuits and Systems* and a guest editor of *Association for Computing Machinery (ACM) Journal on Emerging Technologies in Computing Systems*. She is a Member of IEEE and a Technical Program Committee member of the IEEE Symposium on VLSI Circuits, the IEEE Custom Integrated Circuits Conference, the IEEE Asian Solid-State Circuits Conference, and the IEEE/ACM Design Automation Conference.

Jiachen Xu (jxu3@andrew.cmu.edu) received his B.S. degree in computer engineering from Purdue University in 2020. He is currently pursuing his Ph.D. degree at Carnegie Mellon University, Pittsburgh, PA 15213 USA. His interests lie in brain-inspired machine-learning algorithms and embedded system design for wireless applications. He is a recipient of the 2022 International Solid-State Circuits Conference Analog Devices Outstanding Student Designer Award. He is a Graduate Student Member of IEEE.

Yuyi Shen (yuyis1@andrew.cmu.edu) received her B.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA 15213 USA, in 2020, where she is currently pursuing her Ph.D. degree. She held an internship position with Apple Inc. in 2020, and she is primarily interested in RFIC design with a focus on the application of RF circuits to security and device identification. She was a recipient of the International Solid-State Circuits Conference Analog Devices Outstanding Student Designer Award in 2021 and the Ben Cook Graduate Fellowship in 2022. She is a Graduate Student Member of IEEE.

Ethan Chen (ethanchen@cmu.edu) is a research scientist with the Energy-Efficient Circuits and Systems Lab at Carnegie Mellon University, Pittsburgh, PA 15213 USA. His research interests include neuromorphic computing, hardware security, and biomedical interfaces.