Radio Frequency Fingerprinting With Siamese Network

Raju Dhakal, Bhawana Poudel Devkota, Laxima Niure Kandel
Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University
Email: {dhakalr, poudelb2}@my.erau.edu, niurekal@erau.edu

***Abstract*—Cryptographic security measures are typically unsuitable for IoT and UAV networks because of their inherent computational and energy limitations. As a result, lightweight alternatives like radio frequency fingerprinting (RFF) have recently gained significant attention. RFF utilizes deep learning (DL) techniques, such as Convolutional Neural Networks (CNNs), to identify and detect unique fingerprints. However, CNNs present several challenges: they require large training sets, struggle to detect malicious devices not included in the training process, and require retraining the entire model when new devices are added to the network. To overcome these limitations, in this research we propose the use of a Siamese network for device identification. A Siamese network can measure the similarity or difference between the outputs of two input data points. It requires a smaller training set, can easily detect outliers, and does not need retraining when new devices are added to the network. The transmission of wireless signals from 25 different devices is simulated, collecting 600 MB of I/Q samples using MATLAB. The simulation introduces three physical layer imperfections in the transmitter devices: phase noise, DC offset, and frequency offset. Also, the Similarity-Based Device Identification and Outlier Detection Algorithm (SDIODA) is developed to identify known devices, detect outliers, and utilize the collected dataset and the designed Siamese network to verify the performance of the algorithm. The results demonstrate that the Siamese network achieves nearly 100% accuracy in identifying known devices and detecting outliers with the simulated datasets.**

***Index Terms*—Radio Frequency Fingerprinting (RFF), Siamese network, Identification and Detection**

I. Introduction

The rapid developments of wireless communication technologies have resulted in the progress of 5G and beyond networks, Unmanned Aerial Vehicle (UAV) technology, the Internet of Things (IoT), smart cities, wearable technologies, autonomous vehicles, remote healthcare, industrial automation, and so on [1]–[3]. These technologies extensively rely on radio transmissions, encompassing various standards such as Wi-Fi, Bluetooth, and cellular communication. Nevertheless, significant security threats can be encountered due to the broadcast nature of radio transmission. These security threats can include eavesdropping, spoofing, man-in-the-middle, unauthorized access, jamming, denial of service, and many more attacks [4]. To protect these wireless devices against these threats, a proper authentication mechanism may be required so that the system can verify legitimate users and block suspicious users. Traditional cryptographic authentication methods like password-based authentication, certificate-based authentication, and so on often require a substantial demand of computational and energy resources and the complex task of managing secret keys, which may not always be applicable in resource-constrained environments such as IoT, UAV networks, etc. [5]. Thus, designing a lightweight approach to identify devices with low computational and energy demands is often desirable.

Radio Frequency Fingerprint (RFF) based identification has proven to be an efficient and lightweight method for device authentication, making it especially suitable for IoT environments [6]. All wireless communication systems involve processing message signals through various signal processing components to generate the Radio Frequency (RF) signal to be transmitted. These components inside transmitter systems may include an Analog-to-Digital Converter (ADC), Digital-to-Analog Converter (DAC), oscillators, power amplifiers, mixers, filters, etc. Unique imperfections such as the non-linearity of power amplifiers, oscillator phase noise, the imbalance in the In-phase (I) and Quadrature (Q) components of signals, and so on are often inherent in these electronic components [7]. These imperfections are often tolerable, distort the signals slightly, do not deteriorate the quality of signals, and hence do not impact the functionality of the communication system. The use of these hardware imperfections in the aggregate to identify devices is called RFF. RFF is very robust regarding environmental factors, resulting in long-term stability. In addition, RFF is device-specific, which makes it tough for adversary devices to imitate those imperfections. These properties of RFF make it suitable for identifying devices.

Identifying these RFFs often requires training complex models, such as CNNs, to distinguish subtle signal variations. In the classification stage, devices are classified based on the extracted features. However, using complex models like CNNs has some drawbacks. These models can struggle to identify outliers (devices not involved in the training process), and if a new device is added to the library, the entire model needs to be retrained from scratch. To address these issues, Siamese networks have become very popular. They can be trained with a small amount of data, easily detect outliers, and do not require retraining when new devices are added to the library [8]. This architecture allows for more efficient and flexible device identification, making it a suitable alternative to traditional CNN-based RFF authentication methods.

The key contributions of this paper are as follows:

- We collected 600 MB of I/Q samples from 25 devices through a MATLAB simulation. Each device contributed 10,000 frames, with each frame containing 160 I/Q samples. The entire dataset is made publicly available

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

via the GitHub repository [1] .

- We implemented Siamese network and trained it using I/Q samples from 20 known devices and 5 unknown devices for outlier detection.
- We developed a new algorithm called Similarity-Based Device Identification and Outlier Detection Algorithm (SDIODA) to identify known devices, detect outliers and verify the proposed algorithm using our dataset and trained model.
- We demonstrate that the trained Siamese network can effectively identify the in-library (known) and out-of-library devices (outliers) with an accuracy of 100%.

II. Related Work

In recent years, the research community has increasingly focused on using Deep Learning (DL) methods to identify devices through RF fingerprints. Numerous studies have demonstrated the effectiveness and accuracy of DL techniques, particularly with the advancements in powerful tools like CNNs and Recurrent Neural Networks (RNNs), for identifying transmitter fingerprints. This section will explore some of the recent works in this field.

Tong et al. [9] applied machine learning to RF fingerprinting using a 400 GB dataset of I/Q signals from 10,000 devices. They compared a baseline model with an enhanced ResNet-50-1D, analyzing WiFi and ADS-B signals. ADS-B signals were found to be more accurate and easier to classify. Surprisingly, the baseline model sometimes outperformed ResNet-50-1D and was six times faster in processing. However, both models struggled to identify unseen devices and required retraining to include new devices due to the limitations of the conventional CNN approach. Kandel et al. [10] used the relative phase differences of oscillators in MIMO transmitters to create unique device fingerprints. Testing involved 17 identical devices in indoor non-line-of-sight settings, with the receiver moved manually to simulate static and mobile scenarios. A deep neural network with one input layer, one output layer, and five hidden layers processed the fingerprints, achieving 97% accuracy for static devices and 92% for mobile ones. However, the system struggled to identify outlier devices. Li et al. in [11] used the concept of transfer learning to identify RF fingerprints of samples collected from HACKRF ONE SDR by varying preprocessing techniques, learning methods, and evaluation metrics. This still lacks detecting outlier devices. Additionally, the reliance on a traditional deep learning network poses challenges in adapting to new devices, thereby affecting the overall robustness and scalability of the system. There are several works, including [12]–[16], that work in RF fingerprinting to identify wireless devices by using CNNs and have made significant contributions in this area. However, many of them share different limitations. Most of these systems are not capable of detecting outliers. Also, they are not scalable enough to include new devices within the systems and require retraining of the model from scratch when a new device is added to the system.

[1] https://github.com/POUDELB2/RF_Fingerprinting_Dataset

Recent research has explored using Siamese networks to identify RF fingerprints. Louis et al. [17] implemented a Siamese network to analyze I/Q samples collected from 16 identical USRPX310 Software Defined Radios (SDRs) in two configurations: over the air and the cable. Their approach achieved an accuracy of up to 0.99. However, their work was limited to training and testing the Siamese network with the same and different inputs. They did not extend their model to multi-class classification or outlier detection. Another work by Guomin Sun [18] proposed to use Combined Siamese networks for Transmitter Identification (CSNTI) to identify 12 radio devices (8 known and 4 unknown) whose I/Q samples were captured by NI USRP-2974. This method consisted of a series of Siamese network classifiers, where each classifier differentiates one transmitter from the others. The output of all classifiers is normalized using the softmax function. They achieved an overall accuracy of 0.87 in identifying known devices and detecting unknown devices. However, this system is not robust enough to handle the increased number of transmitters. For example, if we have 100 known devices, we should train 100 Siamese network classifiers, ultimately increasing the computational complexity. Also, when the new device is included in the system, the model should be retrained by increasing the number of classifiers each time.

Upon a comprehensive review of recent advancements in RF fingerprinting, we found that CNNs are predominantly employed to capture subtle variations in device fingerprints. Although CNNs are effective, they require retraining whenever new devices are introduced, limiting their scalability. Additionally, emerging research has begun exploring using Siamese networks for RF fingerprinting. However, these efforts often overlook the crucial aspect of distinguishing between known devices and identifying outliers. To address these challenges, we propose a novel approach utilizing Siamese networks to recognize known devices and effectively detect outliers. This approach aims to enhance the system's ability to adapt to new devices without extensive retraining, improving its robustness and scalability in a changing environment.

III. Methodology

This section describes the MATLAB dataset simulation process, explains the Siamese network model implementation with the CNN base model and outlines the working of the SDIODA algorithm for identifying known and unknown IoT devices.

A. Dataset simulation

To generate a comprehensive dataset for RF fingerprinting, transmission and reception of WLAN beacon frames across multiple IoT devices with hardware impairments namely phase noise, DC offset, and frequency offset are simulated using MATLAB. The generated WLAN beacon uses frequency of 5 GHz with a 20 MHz bandwidth. A total of 25 IoT devices are considered. Out of 25 IoT devices considered, 20 are known or authentic and are labeled with corresponding MAC addresses. The remaining five are considered unknown or malicious (not involved in the training process) and are labeled

as "Unknown". Each impairment varies slightly for example DC offset variation ranges from -50 to -32 *mV*, Phase noise variation ranges from 0.01 to 0.3 *radian*, and frequency offset variation ranges from -4 to 4 MHz. Random values within these ranges are assigned to each IoT device to simulate realistic impairments. Each IoT device is assigned a unique MAC address that generates beacon frames. These frames are transmitted through a channel with Additive White Gaussian Noise (AWGN) and a Rayleigh fading multi-path channel with a Signal-to-Noise Ratio (SNR) of 20 dB. Then, the received signal is processed by the receiver, where Legacy-Long Training Field (L-LTF) sequences of 160 I/Q samples are extracted. For each IoT device, 10,000 frames are collected, resulting in 250,000 frames. The extracted sequences are divided into training, validation, and test sets in the ratio of 7:1:2. All data samples with "Unknown" labels are excluded and saved separately to prevent them from being used in the model's training process.

B. Base model for Siamese network

The CNN illustrated in figure 1 serves as the base model for the implemented Siamese network. The CNN model begins with an input layer of size (160, 2) and applies batch normalization to standardize the data, ensuring mean (μ) of zero and a standard deviation (σ) of one is maintained. For each of the frames, I and Q components are arranged into a matrix of size [`frame_length`, 2], where each row corresponds to a sample and the two columns represent the I and Q values. This matrix is then expanded to [`frame_length`, 2, 1], resembling an image with two channels, one for the I component and one for the Q component. A CNN can then process these "image-like" frames for RF fingerprinting. The value of frame_length is set to 160.

Zero padding of size two is then applied to preserve the edges for the subsequent convolutional layers. The architecture includes four convolutional layers with ReLU activation functions, each followed by max-pooling layers with a pool size of one for dimension reduction. The first and second convolutional layers have 64 filters with sizes of 2 and 4, respectively. The third and fourth convolutional layers each have 32 filters, with sizes of 16 and 32, respectively. After these layers, a flattening layer prepares the data for the dense layer. The dense layer comprises 256 neurons with a sigmoid activation function. Finally, the output layer of size 256 provides the final output of the CNN. The hyperparameters used in our model are based on a previously published article by Birnbach et al. [19].

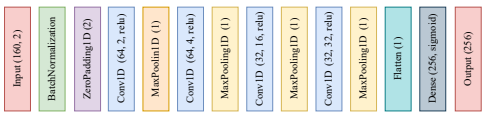


Fig. 1: Layers in CNN.

C. Siamese network

Figure 2 illustrates the structure of the Siamese network. This network takes two inputs, which are processed by two identical CNNs. The outputs of these subnetworks, referred to as fingerprints, are then used to compute the Euclidean distance, as shown in equation 1. This distance is referred to as the similarity score. The Siamese network is trained such that similar inputs yield a similarity score close to one, while different inputs result in a score near zero. The CNN model described in Section III-B is the foundation for the Siamese network we implemented.

$$d(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{\sum_{i=1}^{n} (f_{1,i} - f_{2,i})^2} \tag{1}$$

where:

- $\mathbf{f}_1$ and $\mathbf{f}_2$ represents the fingerprint vectors from the two identical CNNs,
- $d(\mathbf{f}_1, \mathbf{f}_2)$ is the Euclidean distance between $\mathbf{f}_1$ and $\mathbf{f}_2$,
- The indices i range from 1 to n, where n is the dimensionality of the fingerprint vectors.

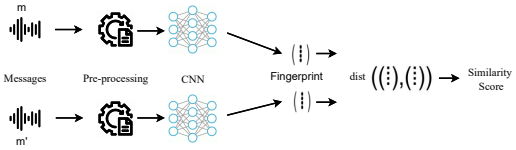


Fig. 2: Structure of Siamese Network.

D. Similarity-Based Device Identification and Outlier Detection Algorithm (SDIODA)

Among the 25 devices in our dataset, 20 are known devices (used to train the model), and the remaining 5 are assumed to be unknown devices (not incorporated in the training session). Figure 3 shows the flowchart of SDIODA to identify known and unknown devices for each sample in the test set with data points from known and unknown devices. First, we randomly select a reference sample from a train set from each of the 20 known devices to identify the device for each data point. The similarity score is calculated for each reference sample, generating 20 similarity scores. The highest similarity and corresponding index are selected and compared against the threshold value of the similarity score (i.e., 0.57). If the highest similarity score is less than the threshold, the sample is determined to be from an unknown device (outlier). Otherwise, the sample is from one of the 20 known devices. The known devices are identified by the corresponding MAC address of the device.

IV. Results

This section describes the proposed Siamese network's performance in detecting and identifying known and unknown IoT devices.

A. Learning curve

Our training set consists of 140,000 data samples, with 7,000 samples from each of the 20 known devices. Positive pairs are created by randomly selecting two different samples

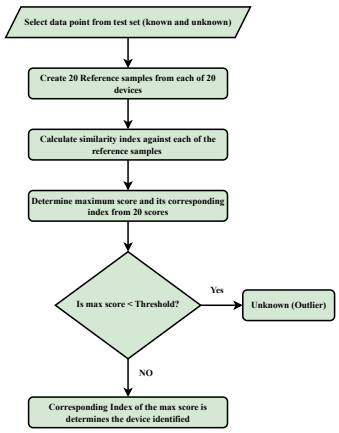


Fig. 3: Flowchart of SDIODA to identify known and detect unknown devices.

from the same class. Negative pairs are formed by randomly selecting one sample from one class and another from any of the remaining 19 classes. In our labeling scheme, pairs of similar samples (from the same class) are labeled with 1, and pairs of dissimilar samples (from different classes) are labeled with 0. We trained our model with training sets for 15 epochs with a batch size of 256 and a learning rate of 0.0001. The learning curve in figure 4 swiftly reaches low loss and higher accuracy on training and validation data, indicating that the Siamese network is learning efficiently. Also, we can observe that validation loss and accuracy are very close and stable with training loss and accuracy, which suggests that the model generalizes well to the unseen data. Furthermore, the model performs without overfitting since there is no noteworthy gap between the training and validation curves.

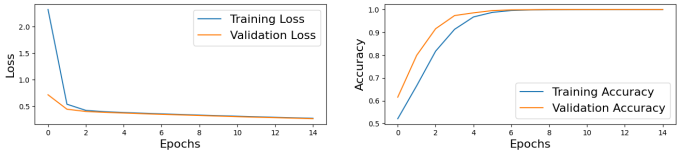


Fig. 4: Learning curve of trained Siamese network.

B. Performance of Siamese network to distinguish similar and dissimilar pairs

To examine the performance of the trained Siamese network, we first tested the test data set with known devices (data from devices involved in the training process). This test dataset consists of 20% of the dataset from each known device. To generate the pair of similar and dissimilar devices, each data point in the test set is paired with one random data point from the same device and another random data point from the different devices, thereby assigning label 1 for the data pair from the same device and label 0 for the data pairs from different devices. Hence, the test set, consisting of a total of 40,000 data points (2,000 data points from each of 20 devices), yields 80,000 pairs of data points. The confusion matrix in figure 6 indicates that all the similar and dissimilar pairs are

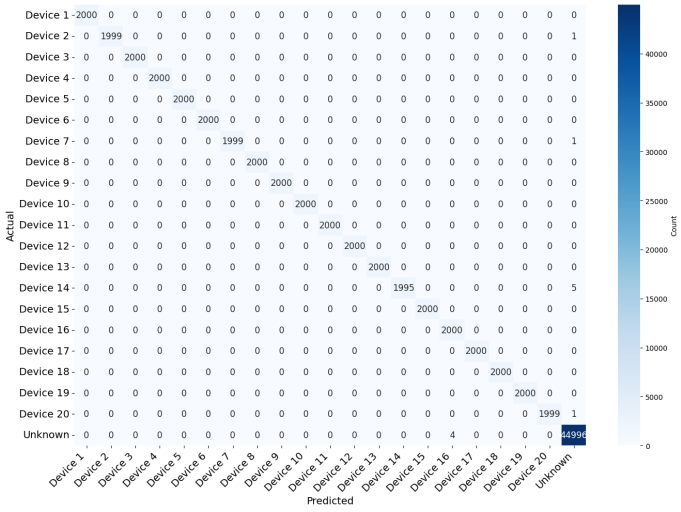


Fig. 5: Confusion matrix for test set with known (device 1-20) and unknown devices (device 21-25).

classified with an accuracy of 100%. This proves that the Siamese network perfectly differentiates similar and dissimilar pairs.

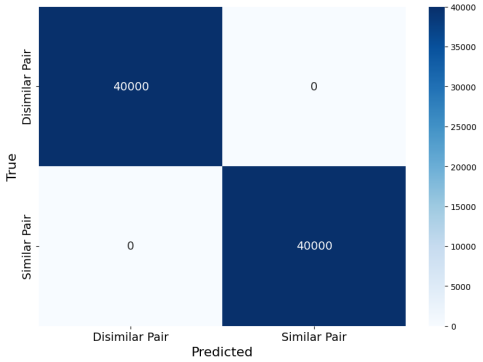
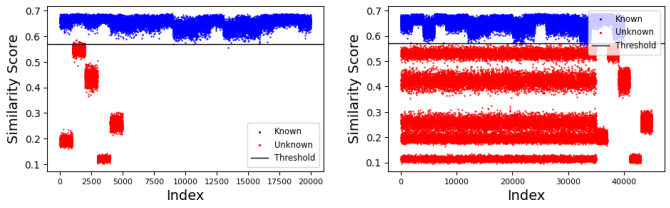


Fig. 6: Confusion matrix to demonstrate the performance of Siamese network.

C. Identification of known devices and detection of unknown devices

Furthermore, to demonstrate the performance of the designed model in identifying known devices and outliers (devices that are not involved in the training process), we created another dataset that consists of data points from the previous test set and all data points from unknown devices and used the algorithm discussed in the section III-D. This yields datasets consisting of a total of 90000 data points (40000 from known devices and 10000 from each of five unknown devices). The confusion matrix in figure 5 demonstrates the model's performance in identifying known and unknown devices. We can observe that known and unknown MAC addresses are identified with an accuracy nearly equal to 100%. Very few samples from known and unknown devices are misclassified. This proves the significant performance of the Siamese network in identifying in-library devices and outliers.

D. Scatter plot of similarity scores for positive and negative pairs in the test set with known and unknown devices



(a) (b)

Fig. 7: Scatter plots showing similarity scores for 20 known (device 1-20) and 5 unknown (device 21-25) devices. (a) Validation set. (b) Test set.

Figure 7(a) and 7(b) show the scatter plot of the similarity scores of each data point in the validation set and test set, respectively. The red points in the scatter plot are the similarity scores from unknown devices, and the blue points are the similarity scores for data points from known devices. We can see the clear distinction between known and unknown devices, making it easy to determine the threshold value.

V. Limitations and Future Work

Although we demonstrated that the Siamese network could effectively identify wireless transmitters and detect outliers with the I/Q samples, this work has some shortcomings. Firstly, the dataset used in this experiment was collected from a MATLAB simulation, which may not be enough to mimic the transmission of signals in a real-time environment. In addition, the work is limited to processing data from 25 identical devices. Furthermore, the data collection considers only three types of hardware impairments, namely phase noise, DC offset, and frequency offset, whereas, in real communication systems, many other impairments exist, such as I/Q imbalance, impedance mismatch, clock jitter, and so on. To overcome these limitations, real-world experiments involving diverse wireless devices that account for a wide range of imperfections are necessary.

VI. Conclusion

We propose a lightweight RF fingerprinting-based authentication approach designed for computation and energy-constrained systems, such as IoT devices. To enhance scalability, we introduced the use of a Siamese network for device identification, eliminating the need to retrain the model when new devices are added to the network. Additionally, we collected 600 MB of I/Q samples from 25 different devices using MATLAB simulations. To further improve device identification, we developed an algorithm called SDIODA, capable of identifying known devices and detecting unknown ones. The effectiveness of the SDIODA algorithm was validated using a simulated dataset within the framework of a designed Siamese network. The results demonstrated that the proposed approach achieved 100% accuracy in both identifying known devices and detecting unknown devices. In summary, our research demonstrates that the integration of Siamese networks and the SDIODA algorithm provides a lightweight, scalable, and reliable solution for device identification in resource-constrained environments.

References

[1] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, 2017.

[2] W. Ejaz, A. Anpalagan, M. A. Imran, M. Jo, M. Naeem, S. B. Qaisar, and W. Wang, "Internet of things (iot) in 5g wireless communications," *IEEE Access*, vol. 4, pp. 10 310–10 314, 2016.

[3] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 112–120, 2017.

[4] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions," *Electronics*, vol. 12, no. 6, p. 1333, 2023.

[5] M. I. Bhat and K. J. Giri, "Impact of computational power on cryptography," *Multimedia Security: Algorithm Development, Analysis and Applications*, pp. 45–88, 2021.

[6] J. Zhang, G. Shen, W. Saad, and K. Chowdhury, "Radio frequency fingerprint identification for device authentication in the internet of things," *IEEE Communications Magazine*, 2023.

[7] B. Chatterjee, D. Das, S. Maity, and S. Sen, "Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning," *IEEE internet of things journal*, vol. 6, no. 1, pp. 388–398, 2018.

[8] D. Chicco, "Siamese neural networks: An overview," *Artificial neural networks*, pp. 73–94, 2021.

[9] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for rf fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[10] L. N. Kandel, Z. Zhang, and S. Yu, "Exploiting csi-mimo for accurate and efficient device identification," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[11] H. Li, K. Gupta, C. Wang, N. Ghose, and B. Wang, "Radionet: Robust deep-learning based radio fingerprinting," in *2022 IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 190–198.

[12] S. Wang, L. Peng, H. Fu, A. Hu, and X. Zhou, "A convolutional neural network-based rf fingerprinting identification scheme for mobile phones," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 115–120.

[13] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2604–2616, 2021.

[14] D. Huang, A. Al-Hourani, K. Sithamparanathan, W. S. Rowe, L. Bulot, and A. Thompson, "Deep learning methods for device authentication using rf fingerprinting," in *2021 15th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2021, pp. 1–7.

[15] J. Yu, A. Hu, G. Li, and L. Peng, "A multi-sampling convolutional neural network-based rf fingerprinting approach for low-power devices," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 1–6.

[16] J. Yang, H. Gu, C. Hu, X. Zhang, G. Gui, and H. Gacanin, "Deep complex-valued convolutional neural network for drone recognition based on rf fingerprinting," *Drones*, vol. 6, no. 12, p. 374, 2022.

[17] L. Morge-Rollet, F. Le Roy, D. Le Jeune, and R. Gautier, "Siamese network on i/q signal for rf fingerprinting," in *Conference on Artificial Intelligence for Defense (CAID) 2020*, 2020.

[18] G. Sun, "Rf transmitter identification using combined siamese networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2021.

[19] S. Birnbach, J. Smailes, R. Baker, and I. Martinovic, "Adaptable hardware fingerprinting for radio data links and avionics buses in adversarial settings," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. IEEE, 2023, pp. 1–10.