Radio Frequency Fingerprint Identification for Device Authentication in the Internet of Things

Junqing Zhang, Guanxiong Shen, Walid Saad, *Fellow, IEEE*, and Kaushik Chowdhury, *Senior Member, IEEE*

***Abstract*—Device authentication of wireless devices at the physical layer could augment security enforcement before fully decoding packets. At the upper layers of the stack, this is conventionally handled by cryptographic schemes. However, the associated computing overhead may make such regular approaches unsuitable for the emerging class of Internet of Things devices, which are typically resource-constrained and embedded in areas that make them difficult to retrieve and reprogram. In contrast, radio frequency fingerprint identification (RFFI) exploits the unique hardware features as device identifiers at the physical layer. This article reviews both the state-of-the-art in engineered feature-based RFFI protocol design and advances in recent deep learning-based protocols, as well as a hybrid protocol that combines their advantages. Specifically, the hybrid approach leverages two methods: a more versatile distance-based classifier and an automatic feature extractor. This article also summarizes the goals of identification, verification and classification as applicable to RFFI, and how they can be achieved by the above protocols.**

***Index Terms*—Internet of Things, device authentication, deep learning, radio frequency fingerprint identification**

I. Introduction

The pervasiveness of wireless networks has made it necessary to secure them against a plethora of threats. In particular, proper authentication of device identities that access a network is crucial, and it constitutes a major security challenge for emerging networks such as the Internet of Things (IoT) [1]. Conventional device authentication schemes mainly rely on cryptographic algorithms and protocols as well as unique device identifiers. The former is usually used to design a challenge-response protocol, which requires a common key shared between two devices. The latter is represented by software addresses, such as MAC addresses.

Cryptographic authentication schemes face some challenges when applied to an IoT environment. First, it is difficult for some IoT applications to regularly refresh their key. Key distribution is usually done by cryptography but IoT devices may not be able to afford the costly computation. Instead, they have to use a constant key that is vulnerable to several types of threats. Second, the IoT will encompass legacy devices that do not have support for firmware updates. Meanwhile, software addresses can be tampered with quite easily, hence, they cannot be considered unique.

Therefore, there is a clear and urgent need for unique and stable device identifiers for IoT devices. Radio frequency fingerprint (RFF) has emerged as a promising candidate [2]. For all wireless devices, there exist inevitable differences in the hardware components due to variations in the manufacturing process, even when those devices are produced from the same manufacturing product line. These hardware impairments typically include oscillator imperfection, mixer imbalances, and power amplifier (PA) non-linearity [2]. There are unique and stable impairments and they can be exploited as device fingerprints to represent wireless devices.

RFF identification (RFFI) is the protocol that leverages RFF for authentication. In particular, radio devices transmit as normal and the wireless waveform will be distorted by hardware impairments. The receiver will classify the identity of the transmitter solely based on the received signal. In other words, RFFI protocols are typically deployed at the receiver side, and no modification or interaction is required from transmitters. Hence, RFFI is applicable to all IoT techniques such as WiFi [3]–[6], IEEE 802.15.4/ZigBee [7], [8], and LoRa [9]–[11].

The majority of the existing RFFI works can be categorized into engineered feature-based protocols and deep learning-based protocols. Early engineered feature-based RFFI works such as [3] and [8] design algorithms to extract a subset of hardware features manually, and, thus, they require expertise for an understanding of the adopted communication protocol. Meanwhile, there has been a recent surge of research works that employ deep learning to perform RFFI [5]–[7], [9]–[12]. In particular, these prior works leverage the inherent ability of deep learning to perform automatic feature extraction.

RFFI can enable the identification, verification and classification of radio devices. These RFFI processes are used, respectively, to identify device legitimacy, verify the asserted identity, and classify device labels. While these tasks have been investigated individually, a comprehensive overview and comparison among them are missing. The main contribution of this paper is, thus, a holistic review of conventional engineered feature-based protocols and deep learning-based approaches. In particular, we discuss how their implementations achieve the above three tasks and their limitations. Then, we present a hybrid protocol to leverage the automatic feature extraction capability of the deep learning model and the versatile distance-based classifier. This paper provides the first tutorial on how RFFI can achieve identification, verification, and classification for radio devices.

J. Zhang and G. Shen are with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom. (E-mail: junqing.zhang@liverpool.ac.uk; guanxiong.shen@liverpool.ac.uk)

W. Saad is with Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, VA 24061 USA (E-mail: walids@vt.edu)

K. Chowdhury is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. (E-mail: krc@ece.neu.edu)

Digital Object Identifier xxx

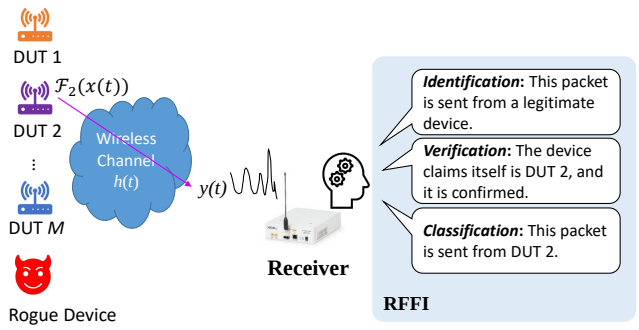

Fig. 1. The overview of an RFFI protocol.

RFF is extracted from RF signals, which are impacted by wireless channel propagation effects. In addition, as a device identifier, it is also important to ensure their uniqueness and stability over time and environmental conditions such as temperature variations. These remaining challenges are discussed to inspire future research on RFFI, which are essential for RFFI to be successfully applied in the real world.

II. RFFI Overview

As shown in Fig. 1, a typical authentication problem involves M legitimate devices under test (DUTs) communicating with a receiver responsible to perform device authentication. The received signal from a DUT m can be given as

$$y(t) = h(t) * \mathcal{F}_m(x(t)) + n(t), \tag{1}$$

where $h(t)$ is the wireless channel, $*$ denotes the convolution operation, $x(t)$ is the transmitted signal, $\mathcal{F}_m(\cdot)$ represents hardware effects of the m-th DUT, and $n(t)$ is the additive white Gaussian noise (AWGN). The transmitted signal consists of a header and payload. A software address (e.g., a MAC address) will be present in the header, and is often used as the device identifier in conventional authentication schemes.

The transmitted signal passes through a series of hardware blocks, including the modulator, digital-to-analog converter (DAC), mixer, oscillator, PA, and antenna. Due to the manufacturing variation, these components are subject to hardware impairments, including DAC sampling errors, e.g., quantization errors and integral nonlinearity (INL), mixer imbalance (both gain and phase), PA nonlinearity, oscillator imperfections such as carrier frequency offset (CFO) and phase noise, as well as antenna characteristics [2]. These impairments slightly distort the signal and their effects are collectively represented by the $\mathcal{F}_m(\cdot)$ in Equation (1). They are unique and difficult to be tampered with, and, hence, they can be leveraged as device identities.

RFFI aims to exploit these unique hardware impairments for device authentication and provide an additional security mechanism. It can be implemented on top of a classical and existing receiver architecture. In practice, RFFI is an open-set recognition problem as there will always be unknown devices and/or rogue attackers [6]. As illustrated in Fig. 1, the tasks in the RFFI literature can be categorized as follows:

- *Identification* distinguishes legitimate devices from any potential rogue devices, which is a binary classification task.
- *Classification* infers the index/label of legitimate devices under a closed-set assumption.
- *Verification* ascertains whether the packet is sent from the asserted device with an algorithmically derived ID.

It should be noted that the terminology used in the RFFI community differs from that of the classification literature. For example, identification usually refers to identifying/classifying classes but in RFFI literature identification has been used to detect rogue devices. Many RFFI classification studies only consider closed-set recognition, focusing on designing different approaches to achieve higher classification accuracy. It may not be considered a security concept though as only known devices are involved. However, we used these task definitions to align with the RFFI literature.

RFFI typically involves extracting RFF features embedded in RF signals (feature extraction) and determines whether the fingerprint closely matches any of those of a known set of legitimate transmitters (classifier design). Conventional RFFI approaches extract low-dimensional features manually, as detailed in Section III. Thanks to deep learning, recent RFFI works leverage the automatic feature extraction capability of deep learning and significantly extended the RFFI realm, as explained in Section IV. Section V then presents a hybrid RFFI algorithm by combining the deep learning-based feature extractor and distance-based classifier. A comparison of the works adopting these approaches is given in Table I, which will be elaborated in the following sections.

III. Engineered Feature-based RFFI

As shown in Fig. 2, there are two types of engineered feature-based protocols, namely distance-based solutions (Section III-B) and machine learning (ML)-based solutions (Section III-C), depending on different classification designs. Both rely on engineered feature extractors (Section III-A) and consist of training and test stages.

A. Feature Extractor

Engineered feature-based protocols need to extract hardware features manually. In practice, we may not be able to estimate all of the features. Instead, only a subset of features can be extracted for any given DUT m, captured by $f_m(\cdot)$. For example, the PARADIS designed in [3] is one of the seminal works in this area that extracts CFO, SYNC correlation, IQ offset, magnitude error and phase error from WiFi devices. The extraction relies on the underlying communication protocols and elegant mathematical models. For instance, the CFO can be estimated using repeated preambles in each packet. Hence, this approach can be considered model-based.

Engineered feature extraction requires an expert understanding of the adopted communication protocol and transmitter architecture. However, it is challenging to extract individual features accurately as some of them are interrelated. For example, phase imbalance of the mixer, CFO, phase noise, and PA nonlinearity all contribute to phase rotations. It is difficult

TABLE I
COMPARISON OF RFFI ALGORITHMS

Category	Paper	IoT Technique	RFF Feature/Signal Representation	Classifier	Task
Engineered Feature-Based RFFI	[3]	WiFi	Frequency error, SYNC correlation, IQ offset, magnitude error, phase error	kNN, SVM	Identification and Classification
	[8]	IEEE 802.15.4/ZigBee	Differential Constellation Trace Figure (DCTF), frequency offset, Constellation Trace Figure Features (CTF)	Distance-based	Classification
Deep Learning-Based RFFI	[9]	LoRa	IQ samples, FFT coefficients	MLP, CNN, SVM	Classification
	[7]	IEEE 802.15.4/ZigBee	IQ samples	CNN	Classification and Verification
	[10]	LoRa	IQ samples, FFT coefficients, spectrogram	MLP, CNN, LSTM	Classification
	[6]	WiFi	IQ samples	CNN, autoencoder	Identification and Classification
Hybrid RFFI	[11]	LoRa	A CNN-based feature extractor with spectrogram	kNN	Identification and Classification

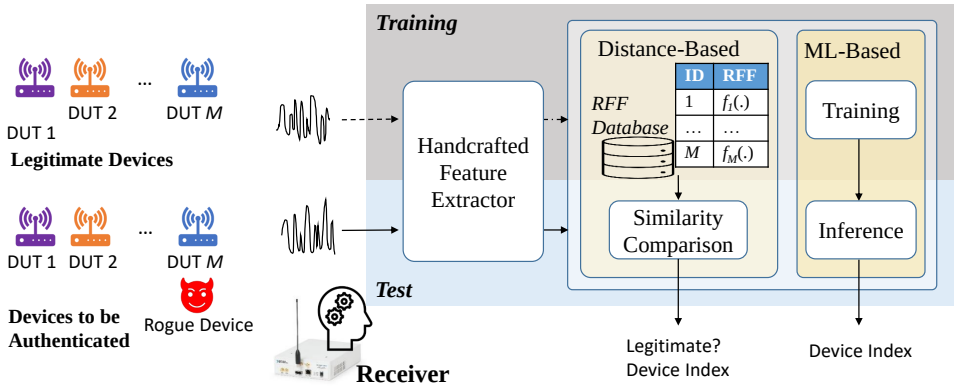

Fig. 2. Conventional engineered feature-based RFFI protocol. Both distance-based and ML-based solutions are shown.

to separate them and estimate their individual contribution. In addition, using only a subset of hardware features potentially limits the classification capacity.

Engineered feature-based RFFI protocols are vulnerable to impersonation attacks. Due to the broadcast nature of wireless communications, transmissions from legitimate users to receivers can be captured by any rogue device within the range. As engineered feature-based approaches rely on explicit features such as CFO and their extraction algorithms are publicly known, attackers can eavesdrop on legitimate transmissions, carry out the same feature extraction procedure and build an RFF database. They can then change their hardware features to masquerade legitimate devices. Such an attack is reported in [13] where the CFO-based RFFI for IEEE 802.11 is subverted. A possible solution is to hide the RFF with noise [13], but this would require hardware access that will not be applicable to IoT devices.

B. Distance-based Solutions

1) Training: Distance-based solutions need to maintain a database that stores the RFFs of each legitimate device. The receiver will collect several signals from each DUT, extract the hardware features, and save them in an RFF database. There might be multiple records of RFF mapping for a given individual device because the feature extraction process is subject to noise and interference.

2) Test: When the receiver captures a packet, it estimates the same type of features used in the training stage and obtains $\hat{f}$. It will then compare the estimated features against the records in the RFF database. The similarity can be measured by distance metrics or cosine similarity. For example, the comparison of the Euclidean distance can be given as

$$\arg\min_m d_m = |\hat{f} - f_m| < \eta, \tag{2}$$

where η is a predefined threshold.

Distance-based RFFI protocols can achieve all three tasks:

- *Identification*: When the signal comes from a legitimate device, a valid decision will be returned by Equation (2). Otherwise, if the signal comes from a rogue device, its hardware impairments will not be the same as or similar to any one of the legitimate devices; no valid result will be returned in this case.
- *Classification*: The receiver will extract the RFF feature of the incoming signal and infer the device index, which is returned by Equation (2).
- *Verification*: The receiver will first decode the claimed identity, i.e., the index m', from the received signal and obtain the relevant RFF from the database, $f_{m'}$, as shown in Equation (2). It will then compare $f_{m'}$ with the estimated RFF of the incoming signal, $\hat{f}$. If the similarity is within the threshold, the verification is successful.

The similarity comparison can be completed by the k-nearest-neighbor (kNN) algorithm [3] or simple distance matching. The work in [8] designed a Euclidean distance-based hybrid classifier that combines multiple features with weights adjusted based on the signal-to-noise ratio (SNR).

C. ML-based Solutions

Classical machine learning models, e.g., support vector machine (SVM) and random forest, are exploited because of their classification capability. We take the widely used SVM [3] as an example.

1) Training: We first need to collect some labelled packets from legitimate devices and then extract their features to form a training set. After that, the SVM is trained and served for future authentication. Different from distance-based solutions, an RFF database is not required.

Since engineered feature-based RFFI protocols are model-driven, their training overhead is quite low. For example, the work in [3] found that 20 packets from each DUT are sufficient to train an SVM model and achieve a good performance.

2) Test: SVM is suitable for classification tasks. The receiver first captures a packet, estimates the features, $\hat{f}$, and feeds it into the well-trained SVM model. Then a predicted label will be returned.

However, the SVM model cannot tell whether the packet is from unknown devices as they are inaccessible during the training stage. This makes it not suitable for verification and identification tasks.

IV. Deep Learning-based RFFI

In the area of deep learning-based RFFI, to our best knowledge, the work in [9] was the first one to apply convolutional neural networks (CNN) and multilayer perceptrons (MLP). Following this work, there have been several deep learning-based approaches that include the use of CNN [2], [5]–[7], [10], MLP [10] and Long Short-Term Memory (LSTM) [10], [12].

An example of a deep learning-based RFFI classification protocol is illustrated in Fig. 3 with a CNN as a basis. Similar to deep learning applied to other domains such as image recognition, the protocol consists of two stages, namely training and test. During the training stage, the receiver will collect signals from all the legitimate DUTs to train a capable model. It can automatically extract all the underlying features, which can fully exploit the hardware impairments and avoid the extraction overhead. At the test stage, an inference is made based on the received signal and the pre-trained neural network model.

Deep learning-based RFFI is data-driven and heavily relies on the amount of training data. The training of a deep learning solution usually requires extensive computational resources that may not be available to low-cost IoT devices. This can be potentially solved by training a neural network at the cloud and deploying the trained model at the devices, as the inference is computationally light.

A. Signal Representation

Deep learning has excellent capability of automatic feature extraction, which can mitigate the disadvantage of manually extracting engineered features. Many RFFI works directly use the time domain RF signals, i.e., IQ samples, as the deep learning input [2], [4]–[6], [10]. RFF might be not evident in the time domain, and transferring time domain signals to FFT coefficients (frequency domain) [4], [9], [10] and spectrogram (time-frequency domain) [10], [11] can expose hidden RFF features, which can be learned by deep learning easier.

It should be noted converting RF signals to various domains is different from manual feature extraction. After the conversion, the signal is still high-dimensional and RFF features are still not unveiled.

B. Identification and Verification

As the natures of the identification and verification processes are similar, they can be processed by the same deep learning techniques with different configurations.

1) Binary Classification: The use of deep learning for identification and verification will rely on binary classification but with different class/label categories.

Some unauthorized devices are introduced during training to emulate rogue devices. A binary classifier is trained in a supervised manner.

- For *identification*, only one neural network is needed; the two classes include legitimate and rogue devices. For example, the work in [6] proposed a so-called Disc. model for identification.
- For *verification*, a separate verification neural network should be trained for each legitimate device m [7]. The two classes are the DUT m and all other devices (could be legitimate or rogue). During the test stage, the system first extracts the claimed identity, m'. It then calls its corresponding verification model to determine whether the claimed identity is true.

It is impossible to access all the potential rogue devices in practice. Whenever a rogue device with similar hardware impairments to a legitimate device is present but not included in the training, misclassification may occur.

2) Anomaly/Outlier Detection: We can consider both identification and verification as an outlier/anomaly detection problem that aims to confirm whether a newly received signal belongs to known distributions. Hence, those two tasks can be implemented using the same deep learning techniques but with a different outlier definition, as follows:

- *Identification* determines whether the transmitter belongs to a group of legitimate devices. In this case, the outliers refer to rogue devices.
- *Verification* is used to confirm whether the transmitter is claiming its true identity. Therefore, the outliers' identities are different from their claimed ones. They can be either legitimate or rogue.

Autoencoder is popular for anomaly/outlier detection [6]. It first encodes the input data to a smaller dimension, and, then uses a decoder to reconstruct the input. The training data

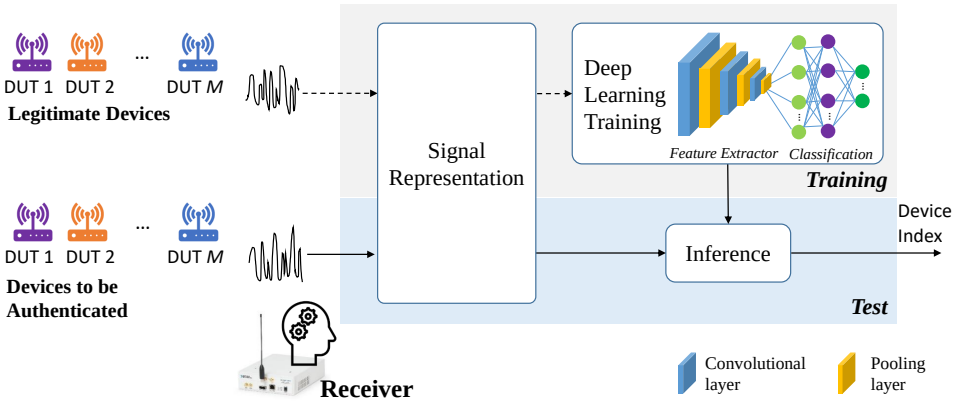

Fig. 3. Deep learning-based RFFI classification protocol. CNN is shown as an example.

consists of signals from legitimate devices for the identification task or signals from the asserted device for the verification task. When the test data is from an outlier, a reconstruction error will be returned. Autoencoder works in an unsupervised learning manner. There is no need to collect data from outliers in advance, which is more realistic and accurate than the binary classifier-based approach. For verification, an individual anomaly detection model should be trained for each legitimate device, the same as the binary classifier-based approach.

C. Classification

Most of the existing deep learning-based RFFI work focuses on the classification task [2], [6], [7], [10], [12]. It is typically cast as a multi-class classification problem in the deep learning context. Hence, state-of-the-art supervised deep learning techniques can be exploited. In particular, MLP [10], CNN [2], [5]–[7], [10], and LSTM [10], [12] are the commonly adopted architectures for RFFI. As shown in Fig. 3, a deep learning classification model typically consists of an automatic feature extraction part and a classification part. The classification is usually carried out by the softmax function that returns a list of probabilities representing the confidence levels for each label. The element with the maximum probability is chosen as the predicted label.

Deep learning classification is a closed-set classification problem because models are trained on a fixed number of known classes. The model cannot distinguish unknown devices, which will instead be classified as legitimate devices whose RFF features are closest. This will cause a significant security loophole since we cannot prevent rogue devices.

D. Joint Identification and Classification

Some deep learning-based approaches can achieve both identification and classification. These solutions will first determine whether the candidate IoT device is legitimate or rogue. If it is a rogue device, its access is denied. If it is legitimate, the protocol will further infer its device index.

A common and straightforward way will be combining two deep learning models to achieve identification and classification separately. For example, the work in [12] used generative adversarial networks (GANs) to detect rogue devices for identification and CNN as well as LSTM for classification. The approach is however cumbersome as the two models should be trained separately.

There are efforts to achieve joint identification and classification using a single deep learning model. For example, the authors in [6] borrowed the open set recognition concept and adopted the OpenMax architecture to classify both known and unknown devices. Specifically, an additional class representing unknown devices is added to the trained neural network model by adjusting the softmax outputs. Meanwhile, the work in [5] cleverly exploits the softmax output information to achieve rogue device detection and legitimate device classification simultaneously. Specifically, when a rogue device arrives, the outputs of the softmax function will not have an obvious winner, which can be inferred there might be a rogue device.

V. Hybrid RFFI Protocol

A. Overview

In IoT applications, devices may join and leave the network frequently, which leads to a variable number of IoT devices. This poses challenges to RFFI design. Deep learning is excellent at automatic feature engineering but it is not scalable in terms of changing the number of classes/labels. In contrast, distance-based classifiers can be easily adjusted to different device numbers. Therefore, a hybrid protocol can synergistically leverage their advantages [11], specifically, the automatic feature extraction capability of deep learning and the versatile authentication functionality of a distance-based classifier.

B. Protocol Design

The proposed hybrid architecture is illustrated in Fig. 4 and is exemplified with CNN as the deep learning feature extractor, which consists of training, enrollment, and authentication stages.

1) Training: A deep learning classification model is trained, following the procedure described in Section IV-C. Once it is successfully trained, the dense layers will be taken out and the remaining layers will serve as a feature extractor. More

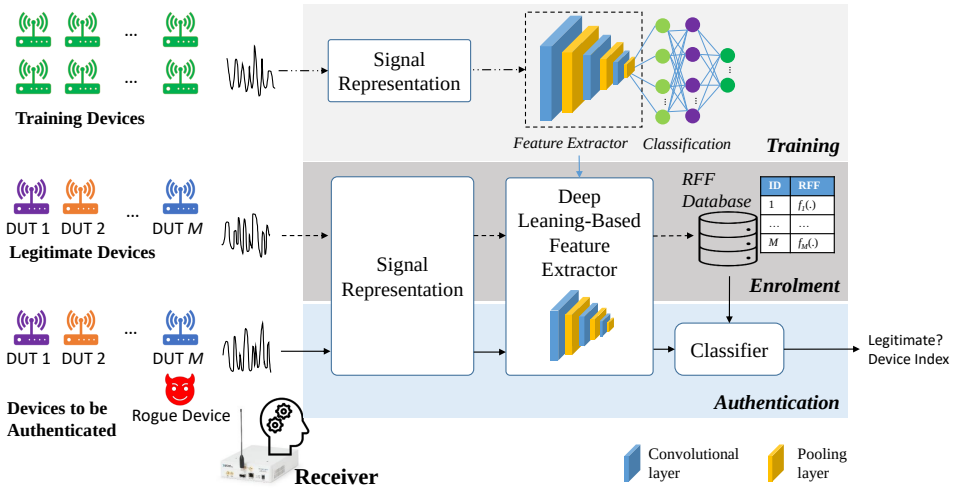

Fig. 4. Hybrid RFFI protocol. CNN is shown as an example of training a feature extractor.

training devices are preferred to produce a deep learning model with a better generalization capability.

2) Enrollment: A deep learning-based feature extractor is used to automatically obtain hardware features. The automatic feature extraction capability of deep learning models alleviates the difficulties of extracting engineered features and can exploit all the underlying hardware impairments. For each DUT, several packets will be transmitted and their RFF will be extracted by the deep learning-based feature extractor and saved into the database. This completes the enrollment stage.

3) Authentication: During the authentication stage, when a test signal arrives, the receiver will first extract the corresponding feature and compare it against the database. The similarities between the feature of the test signal and the ones in the database can be quantified by their distances.

When kNN is used as the classifier, the labels of the K devices with the smallest distances will be returned. When the feature distances between the candidate device and the ones in the database are larger than a threshold, the candidate device is considered rogue (*identification task*). When the candidate device is legitimate, the *classification* is further performed, which is achieved by selecting the mode value of the returned K labels as the device index. We can also extract the claimed device identity and its RFF features for *verification*.

VI. Challenges and Visions for Future Work

A. Security

In general, the security analysis of RFFI is relatively limited and remains an open problem. Defining an impersonation attack to deep learning-based RFFI is difficult because raw signals are used instead of explicit features, but it is still possible. GAN has been leveraged to generate physical waveforms that are the same as the ones emitted by legitimate users [14], which will embed the intrinsic hardware impairments.

B. Impact of Wireless Channel

The dynamics of wireless channels will impact RFFI. As the training and test will be probably carried out at different times and locations, channel effects at these stages will be different. A engineered feature-based protocol is less affected by channel variations because channel effects are often mitigated during the estimation and extraction of hardware features. However, they cannot be removed completely, especially in rich multi-path environments. The channel variations will have a more pronounced impact on deep learning-based protocols because the channel impact is more severe to raw IQ samples [4]. Data augmentation can be leveraged by augmenting the collected data with varying channel conditions [11], but performing an accurate and comprehensive channel augmentation remains challenging.

C. Environment Effect

The stability of RFF features against environmental variation is currently overlooked. Oscillator is probably the most unstable component as it suffers from temperature and time drift. The work in [10] and [15] has revealed that frequency offsets of LoRa and WiFi devices change with temperature. However, the overall stability of RFFI systems is not clear because long-term evaluation lasting years is not available.

D. Dataset Construction

Different from other deep learning research where a large-scale dataset is available for benchmarking, e.g., the ImageNET for image classification, a comprehensive and open-source wireless waveform dataset is missing, which limits performance evaluation and comparison for RFFI [4].

- *Limited Number of DUTs*: As it is time-consuming to carry out experiments for capturing wireless transmissions, most prior art only involves a limited number of devices. The biggest dataset is the DARPA dataset [4], which includes 5,117 WiFi devices with 166 transmissions on average for each device as well as 5,000 ADS-B devices each with 76 transmissions per device on average. However, other works typically use less than 100 DUTs.

For example, the work in [8] employed 54 ZigBee nodes, which is the largest ZigBee population investigated by far.

- *Limited Scenarios*: It is important to evaluate RFFI protocols in as many scenarios as possible for demonstrating their practicality, such as different multipath levels (indoor, outdoor, urban, rural) and various SNR values.
- *Open Source*: There are currently very few open-source datasets suitable for RFFI. There are few examples though such as the WiFi dataset in [4] and the LoRa dataset in [11].

It is indeed challenging but necessary for the community to work together and contribute to an open-source RFFI dataset. It should ideally cover as many candidate wireless technologies as possible. A guideline needs to be agreed upon in advance, which can ensure that the same procedure is followed and compatible data sources can be created.

VII. Conclusions

In this paper, we have provided an overview of RFFI for device identification, verification and classification. We have outlined and compared three types of RFFI protocols, namely engineered feature-based protocols, deep learning-based protocols and a hybrid protocol. Our overview shows that RFFI is a promising device authentication technique but there are still several challenges that must be addressed before reaping its full potential.

References

[1] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 94–104, First Quarter 2016.

[2] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3974 – 3987, 2021.

[3] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. ACM MobiCom*, San Francisco California USA, Sep. 2008, pp. 116–127.

[4] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *Proc. IEEE INFOCOM*, 2020, pp. 646–655.

[5] A. Gritsenko, Z. Wang, T. Jian, J. Dy, K. Chowdhury, and S. Ioannidis, "Finding a 'new'needle in the haystack: Unseen radio detection in large populations using deep learning," in *Proc. IEEE DySPAN*, 2019, pp. 1–10.

[6] S. Hanna, S. Karunaratne, and D. Cabric, "Open set wireless transmitter authorization: Deep learning approaches and dataset considerations," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 59–72, Mar. 2021.

[7] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[8] 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. 2018.

[9] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel, "Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning," in *Proc. ACM WiSec*, 2017, p. 58–63.

[10] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2604 – 2616, Aug. 2021.

[11] G. Shen, J. Zhang, A. Marshall, and J. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 774–787, 2022.

[12] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 783–801, Jun. 2020.

[13] L. F. Abanto-Leon, A. Bäuml, G. H. Sim, M. Hollick, and A. Asadi, "Stay connected, leave no trace: Enhancing security and privacy in WiFi via obfuscating radiometric fingerprints," *Proc. ACM Measurement and Analysis of Computing Systems*, vol. 4, no. 3, pp. 1–31, 2020.

[14] S. Karunaratne, E. Krijestorac, and D. Cabric, "Penetrating RF fingerprinting-based authentication with a generative adversarial attack," in *Proc. IEEE ICC*, 2021, pp. 1–6.

[15] X. Gu, W. Wu, N. Guo, W. He, A. Song, M. Yang, Z. Ling, and J. Luo, "TeRFF: Temperature-aware radio frequency fingerprinting for smartphones," in *Proc. 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2022, pp. 127–135.

Junqing Zhang is a Senior Lecturer (an Associate Professor) at the University of Liverpool, U.K. His research interests include the Internet of Things, wireless security, physical layer security, key generation, radio frequency fingerprint identification, and wireless sensing. He was a recipient of the U.K. EPSRC New Investigator Award.

Guanxiong Shen received the B.Eng. degree from Xidian University, Xi'an, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, U.K. His current research interests include the Internet of Things, wireless security, and radio frequency fingerprint identification.

Walid Saad [S'07, M'10, SM'15, F'19] received his Ph.D. degree from the University of Oslo in 2010. Currently, he is a professor in the Department of Electrical and Computer Engineering at Virginia Tech. He is the author/co-author of ten conference best paper awards and of the 2015 IEEE ComSoc Fred W. Ellersick Prize. His research interests include wireless networks, machine learning, game theory, cybersecurity, unmanned aerial vehicles, and cyber-physical systems.

Kaushik Roy Chowdhury [M'09, SM'15] is a professor at Northeastern University. His current research interests involve systems aspects of networked robotics, machine learning for agile spectrum sensing/access, wireless energy transfer, and large-scale experimental deployment of emerging wireless technologies.