Understanding Radio Frequency Fingerprint Identification With RiFyFi Virtual Databases

Alice Chillet, Robin Gerzaguet, Karol Desnos, Matthieu Gautier, Elena Simona Lohan, Erwan Nogues, Mikko Valkama

IEEE ComSoc IEEE Open Journal of the Communications Society

Received 25 December, 2023; revised 4 April, 2024; accepted 7 Mai, 2024.

Understanding Radio Frequency Fingerprint Identification with RiFyFi Virtual Databases

Alice CHILLET[1], Robin GERZAGUET [1], Karol DESNOS [2], Matthieu GAUTIER [1], Elena Simona LOHAN [3], Erwan NOGUES [4], Mikko VALKAMA [3]

[1] Univ Rennes, CNRS, IRISA, firstname.name@irisa.fr
[2]Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, France, firstname.name@insa-rennes.fr
[3]Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
[4]DGA

ABSTRACT

This paper proposes to explore the Radio Frequency Fingerprint (RFF) identification with a virtual database generator. RFF is a unique signature created in the emitter transmission chain by the hardware impairments. These impairments may be used as a secure identifier as they cannot be easily replicated for spoofing purposes. In recent years, the RFF identification relies mainly on Deep Learning (DL), and large databases are consequently needed to improve identification in different environmental conditions. In this paper, we introduce the so-called RiFyFi_VDG, referring to **R**adi**o** **F**requenc**y** **Fi**ngerprint **V**irtual **D**atabase **G**enerator, and explore individually the impairment impact on the classification accuracy to highlight the most relevant impairment. Different transmission scenarios are then explored, such as the impact of the data type (being a preamble or a payload) and the data size. Design rules of real databases are finally drawn for the different scenarios. We found out that the power amplifier imperfections play the biggest role in RFF accuracy and that average accuracy levels of 94% can be reached when combining the various hardware impairments at the transmitter.

INDEX TERMS Radio Frequency Fingerprint, Deep Learning, Database, RF impairments models.

I. INTRODUCTION

IN recent years, the massive introduction of low-power Internet of Things (IoT) devices increases the sensitivity of wireless networks to attacks. Hence, secure authentication has been the topic of many researches [1], particularly the Radio Frequency Fingerprint (RFF) identification has been largely studied [2]–[5].

The purpose of RFF is to uniquely identify a device by recognizing defects in the signals it emits. This identification is based on the singularity of the hardware, the transmitter component impairments create unique electromagnetic distortions in the transmitted signal [6], and those distortions are used to differentiate devices. This identification solution addresses some security issues of IoT and can be used for non-spoofing authentication [7]. In most telecommunication standards, identification methods are based on the metadata communication protocol that gives an address to enable the authentication, for example, the Media Access Control (MAC) address, but those solutions can be counterfeited [8]. The RFF identification can be combined with a MAC address identification to improve and ensure identification without spoofing. For example, the RFF must correspond to the MAC address to validate authentication. The RFF identification can be useful in different application contexts such as authentication in low-power network devices [7], or cyber defense to recognize the attacker or intrusion [9], or monitoring the behavior of a suspicious device.

To identify a device thanks to its RFF, most related works use the signal in the time domain and classification methods to discriminate among the different potential emitter candidates. The State of the Art (SoA) proposes two families of classification methods, the first one is called model-based or parametric method, and the second is the Deep Learning (DL) based solution. The parametric methods use some features that describe the RFF [6], [10], [11]. This method requires RFF knowledge to choose and compute some feature estimators used for classification. The quality of the estimators can hinder the accuracy of parametric RFF methods.

Recently, with the explosion of the use of DL, many research works have been focused on this second family of classification solutions. The supervised DL techniques use labeled signals from different transmitters during the training phase and learn how to recognize the source of the

different signals. Many DL architectures exist, in particular Convolutional Neural Network (CNN) is used to extract and classify RFFs [12]–[16].

While DL techniques present promising results, there is a strong need for a large and robust database [2]. In the literature, databases are composed of raw signals from different devices captured with one or many Software Defined Radio (SDR). Each signal in the database is labeled to its emitter. During the learning phase, the network has to learn the RFF in the signals to estimate the labels and improve the classification. However, ensuring RFF learning requires an unbiased database [17]. A bias is created when a systematic error is introduced during the database creation. For example, a systematic error can be a different power at emission for each transmitter, a static propagation channel due to the static location of emitters during signal capture [13], [18], a MAC address in data, or the quality of emitters in the sense of the amount of distortion induced by the device hardware. These biases could impact positively or negatively the identification, by helping or distracting the network to classify.

Most existing RFF identification works use experimental data to explore RFF identification [13], [14], [18]. However, these works are limited by the lack of flexibility of their database and by the bias added involuntarily in data. As a consequence, they cannot explore RFF identification in many dimensions, they focus their work on one aspect and it is difficult to combine or compare the different contributions because the databases are not comparable. Few works are based on a virtual database and those that consider all impairments are rare. Zhang et al. [19] propose global modeling but they do not provide access and flexibility to create a new database. To address those issues we propose a new database generator based on a virtual model of both impairments and transmission scenarios. The **R**ad**i**o **F**requenc**y** **Fi**ngerprint **V**irtual **D**atabase **G**enerator (RiFyFi_VDG) allows exploring RFF identification and studying the impact of the quality of the device in a countermeasure context.

The core contributions of this paper are as follows:

- We present, RiFyFi_VDG [20], an open-source Virtual Database Generator based on both a RFF baseband model composed of four impairments and different scenarios of transmission. An open-source RFF Identification Framework called RiFyFi_IdF is also proposed for the evaluation.
- We model the transmitter quality, and study each impairment independently in order to define the limit dispersion to classify devices.
- We study all impairments together in different transmission contexts in function of transmitters quality and then highlight the most significant impairments for RFF identification.
- We study the impact on classification performance of the size of the database in different transmission contexts.

FIGURE 1. **Transmission frame in wireless communications.**

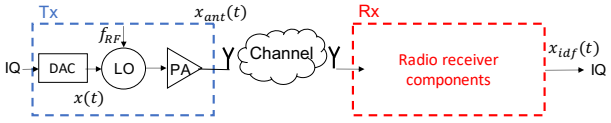

FIGURE 2. **Transmission and reception chain.**

The rest of this paper is organized as follows. In section II, transmission and RFF models are presented, with a focus on their intricacies and pitfalls in an RFF identification context, and some related works are introduced. Section III introduces the RiFyFi_IdF and the transmission models in particular RFF models used in RiFyFi_VDG to create the virtual databases. Then section IV describes precisely the database creation and scenario used in this paper. Then, the section V presents the individual effect of the impairments. Finally, section VI presents the interest of a virtual database in different transmission contexts. The conclusion of the paper is in section VII.

II. Database challenges for RFF identification

A. RF transmission and RFF models

1) RF transmission

Before the transmission, the source information data is converted into symbol sequences. Most wireless communication protocols generate symbol frames composed of three parts as detailed in Figure 1. The symbol sequences are composed of three different parts: the preamble (Pre) which contains synchronization information that is always the same for all the packets, the MAC addresses (MAC) of the emitter and the payload data (Pay). The symbol sequences are converted into IQ samples and transmitted via the RF chain (Tx, Figure 2). A Digital Analog Converter (DAC) transforms the signal into the analog domain to yield $x(t)$. The Local Oscillator (LO) modulates it at carrier frequency f_c, and the Power Amplifier (PA) amplifies the signal, creating $x_{ant}(t)$ for transmission via the antenna.

2) RFF definition and issues

The RFF of a transmitter is a unique signature created by the hardware components of the transmission chain. This is neither controlled nor done on purpose, but it is the result of small flaws in the manufacturing process. This signature called the RFF of the transmitter Tx, is denoted $\mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}$ and could be used in wireless communications to identify the device which has emitted the signals. The emitted signal

can be modeled by:

$$x_{\text{ant}}(t) = \mathcal{F}_{\text{RFF}_{\text{Tx}}}(x(t)), \quad (1)$$

$$x_{\text{ant}}(t) = \mathcal{F}_{\text{PA}} \circ \mathcal{F}_{\text{LO}}(x(t)), \quad (2)$$

where $\circ$ represents the function composition operator, with $h(x) = g \circ f(x)$ denoting $h(x) = g(f(x))$, which expresses the successive processing of the transmission chain and its underlying stages each creating a distortion function. Furthermore, $\mathcal{F}(\cdot)$ represents a component behavior, including its distortion. The nested functions of (2) show the impact of each component and the difficulty of extracting features and modeling the transmission with RFFs. The LO adds distortions related to frequency called Carrier Frequency Offset (CFO), gain and phase called IQ imbalance and Phase Noise (PN), and the PA impacts the gain.

In Figure 2, the channel block represents the wireless communication environment defined by the noise, interference signals, and the multi-path and fading channels that could impact the signal. The propagation channel is modeled by $\mathcal{F}_{\text{channel}}$. The red block Rx represents the receiver with its components (not detailed in the model) and its distortion function called the RFF of the receiver, but investigating its impact is beyond the scope of this paper. The received signal x_{idf} can thus be expressed as:

$$x_{\text{idf}}(t) = \mathcal{F}_{\text{RFF}_{\text{Rx}}} \circ \mathcal{F}_{\text{channel}} \circ \mathcal{F}_{\text{RFF}_{\text{Tx}}}(x(t)). \quad (3)$$

In an experimental study, $\mathcal{F}_{\text{channel}}$ changes in the function of the position and the relative position of transmitters and receiver(s) and the noise level; these also influence the received power. All of these disturbances make difficult the Tx RFF identification. This issue is largely studied in the SoA and will be called channel or environmental condition impact in the rest of the paper. The environmental conditions represent an important bias in database construction for RFF identification DL solutions.

In conclusion, the database challenges are multiple because of the large design space with many communication protocols and the type and number of transmitters. The next subsection presents the related works where the community explores the large design space of databases for RFF, such as the communication standard, the type of devices (emitters and receiver(s)), the part of the signal used to identify, the environmental condition of transmission and the data preprocessing such as transformation, slicing, and multi-packets.

B. Related Works

Recently, the number of contributions on RFF identification has increased [2], mainly due to the number of applications such as authentication for security in IoT systems. The different applications lead to different identification scenarios. For example, Guo et al. [7] use the term "1 to 1 authentication" to verify if the RFF of a device matches its MAC address. A second context defines a list of authorized devices and unauthorized devices, only based on the RFF or by combining RFF and address. Hanna et al. [34] formulate the problem of recognizing authorized transmitters and rejecting new transmitters.

In the literature, two families of RFF identification methods are investigated: parametric and DL techniques. The parametric method uses feature extractors combined with a classic machine-learning solution. This method exploits the intrinsic and unique nature of the impairments to identify the device. For example, PARADIS [6] uses IQ samples and defines different metrics: the IQ origin offset the frequency error and SYNC correlation, to characterize the "radiometric" signature of a device. The use of such parametric methods is strongly limited by the knowledge about the transmission chain, protocol, modulation, and the superposition of impairments.

Recently, the number of RFF identifications by classification methods has exploded with the advent of DL [7], [13]. In particular, supervised DL is massively used in RFF classification, as it automatically learns how to classify radio transmitters by recognizing complex patterns from labeled signals. DL techniques include CNN [14], [15], [35], [36], Long Short-Term Memory (LSTM) [16], [37], Recurrent Neural Network (RNN) [12] and transformers architectures [16], [25], [38]. CNNs are the most popular solution and different architectures have been explored such as AlexNet-inspired with 12M parameters, distributed on 8 convolutional layers and 3 fully connected layers, or CNNs with less layers, but more parameters [14], [15]. Numerous CNN are used for RFF without real comparison between them [2], [35], [36] except in [12].

DL solutions achieve good results, but the classification accuracy of such methods dramatically depends on the database used to train the network. Therefore, there is a strong need for large and robust databases composed of raw labeled signals [2] from different transmitters to ensure RFF recognition in many environmental conditions. Since 2019, the SoA on RFF identification with DL has increased and presents different databases to experiment RFF identification. A selection of recent papers on RFF identification is presented in Table 1. These databases are separated into two types: experimental-based and simulation-based as shown in Table 1. They are created for different wireless protocols which depend on the author's research affinities, presented in column 5 in Table 1 such as WiFi [14], [15], [28] and LoRa [25], [39]. Column 4 "Is data public?" gives information on public access and column 6 "Numbers of emitters" and 7 give respectively information about the number of emitters and additional information or contribution of the paper. The nature of the signals and the frame structure impact the results analysis deducted by the authors and it is therefore difficult to compare different works. For example, [15], [25] uses only the preamble data while in ORACLE [14] the frame consists of a MAC address field with always the same address and a random payload. Jian et al. propose to slice the signals containing the MAC address to be resilient to MAC address spoofing [36].

TABLE 1. Summary table of databases used for RFF identification

Reference	Year	Database	Is data public?	Protocol	Numbers of emitters	Additional informations or Contribution
Experimental Databases						
Hall et al. [21]	2003	Own	No	Bluetooth	10 transmitters	Exploit the phase to detect transient signal
Riyaz et al. [22]	2018	Own	No	WiFi	5 B210	CNN outperforms alternate Machine Learning
Sankhe et al. [14], [15]	2019	ORACLE	Yes	WiFi	16 X310	Add control impairments with feed-back driven to increase differentiability for bit-similar device
		DARPA	No	WiFi	140 DUTs (phones)	
Morin et al. [17]	2019	Own	Yes	WiFi	21 N2932	Generate data to minimise the role of the unwanted channel
Peng et al. [23]	2019	Own	No	ZigBee	54 DUTs	Adopt 4 novels modulation-based features effectives in ZigBee node classifying
Jian et al. [24]	2020	DARPA	No	WiFi ADS-B	5000 DUTs 5000 DUTs	Investigate 2 CNNs for RFF identification under different environmental scenarios
Soltani et al. [12]	2020	DARPA	No	WiFi	50, 250, 500, 5000 DUTs	Study the interest of data augmentation
Al-Shawacka et al. [13]	2020	Own	Yes	WiFi	13 N210 and 7 X310	Evaluate the impact of the wireless channel on CNN-based fingerprinting algorithms
		DARPA	No	WiFi, ADS-B	100 to 10000 DUTs	
Shen et al. [25]	2021	Own	No	LoRa	10 DUTs	Use a transformer as classifer, improve low SNR classification accuracy with data augmentation
Elmaghbub et al. [26]	2021	Own	Yes	LoRa	25 Pycom devices	Study the sensitivity to deployement variability
Reus-Muns et al. [27]	2020	Own	No	WiFi, 4G, 5G	4 USRP X310	Incorporate the triplet loss with the deep CNN
Hanna et al. [28]	2022	WiSig	Yes	WiFi	174 USRPs	Create 4 datasets for RFF identification varying, days, number of emitters, receivers and signals
Jagannath et al. [29]	2022	Own	Yes	Bluetooth	10 DUTs	2 days to do generalization
Chillet et al. [30]	2023	WiSig	Yes	WiFi	6 USRPs	Use Tangled Program Graph as a classifier Compare training duration to SoA CNN
Elmaghbub et al. [31]	2023	Own	Yes	WiFi	50 Pycom devices	2 datasets outdoor and indoor
Elmaghbub et al. [32]	2023	Own	Yes	Wifi	50 Pycom devices	4 datasets: wire, wireless, different locations
Simulation based Databases						
Soltani et al. [12]	2020	Own	Yes	WiFi	10 transmitters	Model only IQ imbalance impairment
Zhang et al. [33]	2021	Own	No	LoRa	50/200 transmitters	Uniformly and randomly distributed IQ imbalances and PA nonlinearities
Chillet et al.	2024	RiFyFi_VDG	Yes	WiFi	N_{Tx}	Virtual database generator Model IQ imbalance, PA, PN, CFO

Databases can be separated into two types: experimental-based and model-based. In the literature, experimental signals are mostly created with IoT Device Under Test (DUTs) or Software Defined Radio (SDR) platforms for both transmitters and receivers. The work of Zhang et al. [19] shows that the transmitter and receiver type is important in RFF identification because the capacity to discriminate two devices is linked to the RFF difference between two devices. For example, a Universal Software Radio Peripheral (USRP) X310, a high-hand quality device, is produced with low variability components, resulting in minimal RF front-end variations between two devices. Sankhe et al. [15] show that X310 is more difficult to separate, compared to B210. Moreover, [10] studies the receiver impact on the classification capacity, a receiver could be sensitive to one emitter. To avoid some classification problems as channel or receiver impacts, some authors propose to pre-process data before using the neural network, [15] proposes an under-complete demodulation that aims at removing only the channel effect from raw IQ samples, without compensating the device imperfections. The pre-processing used in RFF identification are synchronization, domain transform, or equalization [11], and some authors add a CFO compensation because the CFO is sensitive to temperature variation and reduces the system stability [25], [33]. The next subsections present different families of databases used in the SoA of RFF identification. First, the real databases are introduced private and public databases, and finally, we focus on virtual databases.

1) Real Private Database

The largest existing database for RFF identification was created by DARPA in 2020. This database, used by authors from Northeastern University in Boston, is a private one and is used in many papers [12], [13], [15], [24], [36]. This database is composed of two datasets, one with WiFi signals, with 5117 DUTs, and an average of 166 transmissions for each device. The second dataset is composed of ADS-B

signals from 5000 DUTs and an average of 76 transmissions for each device [12]. This database offers the possibility to train the network with a large number of devices. Unfortunately, this database is only available to researchers with US government sponsors.

Peng et al. [23], [40] designed a large RFF database for ZigBee standard. They use 54 DUTs as transmitters and one USRP as a receiver to create the database. They performed ten measurements for each ZigBee device at different locations with line-of-sight transmissions. Their database is only used for their work.

Exploring RFF identification needs a database adapted to the scenario with metadata. Thus for many papers in the literature, the authors create their own database, with few devices [21], [22], [25], [27] but they never give an open access to their database. Consequently, the reproducibility of experimental results is not possible.

2) Real and Public Database

The University of Boston creates its own databases for experiments in papers [14], [15], [22]. First, they created a database with 5 USRPs B210 transmitters with different distances varying between 2 and 50 ft [22]. Then in 2019, they created the ORACLE database with 20 USRPs X310 transmitters [14]. They suggested introducing software-controlled impairments at the transmitter side to enhance identification robustness. This recommendation arises from the fact that the X310 transmitters are produced with low variability components, resulting in minimal RF front-end variations between the two devices.

Hanna et al. proposed a new public database for RFF [28] called WiSig. WiSig is constructed with many signals and with information on how signals have been captured as transmitter location and the type of transmitter used (Atheros). They provide a large-scale WiFi dataset captured by 41 USRPs with 20 MHz bandwidth from different references. The signals come from 174 WiFi transmitters over four different days of captures performed over a month. The authors have created different databases with many transmitters (150), many receivers (32), and many signals (1000 for each transmitter). They present WiSig as a RFF database to explore the identification in a static environment with different types of transmitters and different numbers of transmitter/receiver/signal/days.

In the same way, Al-Shawabka et al. present in [13] a public database for RFF. This database is composed of 4 datasets, each of them is composed of 20 transmitters, 12 B210 and 8 X310, and one fixed receiver. They first explore the best pre-processing and then they explore the impact of antenna and channel with both wired and wireless communications in an anechoic chamber.

Morin et al. [17] work on unbiased database creation, they leverage FIT/CorteXlab anechoic chamber to capture signals and control the propagation environment as well as the interference profile, which enables the full control of the generated datasets. To increase channel variations and to reduce the possibility for the receiver to learn from the channel properties, the MultiRx setup is proposed where they merge the signals observed from several devices acting alternatively as identification receivers. However, this combination of signals creates confusion between the channel effect and receiver effect, which cannot be studied separately.

Jagannath et al. present in 2022 [29] a new public dataset that includes emissions from 10 COTS IoT emitters (2 laptops and 8 commercial chips) that are captured with a USRP X300 device. The dataset is split into two: Day1 and Day2 each of which is recorded in a different time frame, location, and testbed setup to enable critical generalization test of the trained DL model.

Elmaghbub proposes different WiFi datasets [31], [32] composed of 50 Pycom devices. They create outdoor and indoor scenarios and wired and wireless scenarios on different days, and static or dynamic propagation channels. They captured the first two minutes of transmissions using the USRP B210 at a sample rate of 45 MSps. The captured signals were then digitally down-converted to the baseband and stored as I/Q samples on a computer. To avoid any data dependency on the identity of the WiFi transmitter, all transmitters were configured to broadcast the same packets, which include the same spoofed MAC address and a payload of zero-bytes.

3) Virtual and Public Database

Soltani et al. [12] propose to simulate 10 virtual transmitters to create a custom dataset and study the impact of multiplying the number of channels seen by the network during the training phase. However, they decided to model only IQ mismatch because of the complexity of modeling many RF impairments.

Zhang et al. present [33], a model-based database with 4 impairments models. They work on a comprehensive study of RF impairments modeling to address the need for the design of a robust RFF identification protocol. Their model includes LO imperfections, IQ gain and phase imbalance, and PA non-linearities. They study the impact of individual and overall impairments in different configurations and define a robust RFF identification protocol when RF impairments cannot be reconfigurable or customized to help the identification. Their work focuses on the estimation and calibration of the CFO and the calibration of the IQ imbalance of the receiver.

C. Database Challenges

As a conclusion, the design space of RFF database is largely explored by the community. However, it is difficult to design a good training database related to an application context. In most cases, the data environment context of the training set will be different from the test set, making it difficult

to identify transmitters in the test. Real databases cannot provide the flexibility, reproducibility, and exhaustivity we need to understand and ensure that the network is currently learning the RFF, and creating a real database is a long process. Virtual databases are therefore really useful to study RFF identification scenarios and design a robust RFF identification protocols [33]. However, the main disadvantage of virtual databases is a common weakness of digital twins or modeling, which is the veracity of the choice of models. Zhang et al. [33] propose to combine relevant SoA impairment models to create a virtual transmission chain with impairments. However the authors only give access to the final database [12], [33], the community misses therefore a generic virtual database generator. This work addresses the need for the design of a robust database to perform RFF identification with DL. We propose a virtual database generator that creates a database based on the scenario description to study the DL RFF identification process in different scenarios and explore database design space such as the number of signals.

III. RiFyFi System overview

A. RiFyFi Identification Framework

In this section, we present our flexible framework for RFF identification coded in Julia language. Julia [41] is a high-level language, efficient in abstraction and execution, with many DL and telecom libraries [42]. The framework is composed of i) a virtual database generator, RiFyFi_VDG, which allows flexible exploration of many RFF scenarios and ii) a classification stage based on DL. The classification part is trained before being used as a classifier. The global framework presented in Figure 3 offers flexibility to use different classifiers in the classification stage or to use different virtual or real databases. With the virtual database generator, the framework takes as input the description of the database, called scenario parameters: number of signals per transmitter, the number of transmitters, size of the sequences, transmission context, and the information about network and training: the name of the architecture network, the number of training iteration called epoch.

The flexibility of databases created by the generator offers the possibility to explore transmission contexts such as frame symbols, RFF parameters, and environmental conditions such as channels. The SoA shows some difficulties with channel variation [13] and concludes with the need to have a robust system for channel variation. The application context will determine the properties of a system that can be defined as robust. For applications where the time window between training and identification is narrow, generalization is not expected to be a problem as the channel will remain static, especially when considering motionless devices. On the contrary, an application with motion devices requires more generalization to be able to classify devices in different locations. The framework can help to define the metaparameters for database design.

B. RFF Identification with Deep Learning

In recent years, DL has been massively used for classification for RFF as it could learn automatically how to classify transmitter [12]–[15]. Figure 4 describes the classification procedure in the framework. This stage takes as an input the database and the name of the architecture network. The database has been previously created, labeled, and split in Training and Test set. During the training part, the network takes signals from the training set grouped in batches as input. The labels of the signals in the batch are predicted and compared with the true labels using cross-entropy as the loss function to apply the back-propagation. This process is repeated for each batch and each epoch. At the end of the training, the framework saves the network status in `.bson` file and saves the performance evolution during the training phase in a `.csv` file.

The network is considered as an independent block from the database with fixed inputs and outputs. The input size depends on the size of the raw input signal. The output depends on the number of classifying devices, the outputs of the network are the probability of belonging to a class. From these probabilities, the classification prediction is done.

To evaluate the network classification performance, two metrics are used: the F1 score and the accuracy. The accuracy is calculated by counting the number of correct predictions out of the total number of classifications. The F1 score is calculated on the batch sequences as follows:

$$\mathrm{F1} = \mathop{\mathbb{E}}_{c\in\mathcal{C}}\left(\frac{2}{\frac{1}{P(c)}+\frac{1}{R(c)}}\right), \tag{4}$$

$$\text{with } \begin{cases} P(c) = \frac{t_p(c)}{t_p(c)+f_p(c)} \\ R(c) = \frac{t_p(c)}{t_p(c)+f_n(c)} \end{cases}$$

where $\mathbb{E}[\cdot]$ stands for the expectancy operator applied here on all the classes $c \in \mathcal{C}$. $P(c)$ is called the precision for the class c and is a function of the number of true positives $t_p(c)$ and false positives $f_p(c)$. $R(c)$ is the recall for the class c and is function of $t_p(c)$ and the false negatives $f_n(c)$. The F1 score is interesting when the dataset is not balanced. In

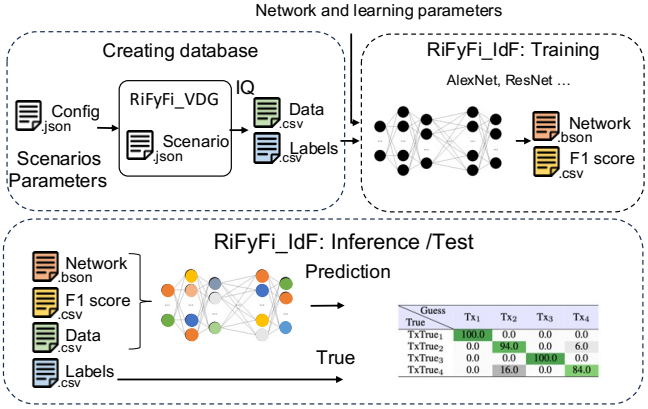

FIGURE 3. Framework flow.

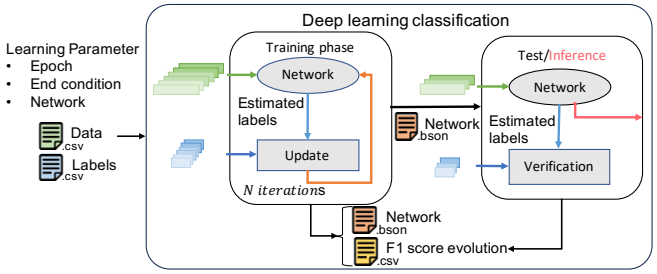

FIGURE 4. **DL classification in framework.**

our case, the F1 score is very close to the accuracy value because the dataset is completely balanced.

C. Proposed Virtual Database and Radio Models

The RiFyFi_VDG creates virtual databases to allow the exploration of impairments, database design, and learning models. Creating a virtual database requires wireless communication models, hardware impairments models, and wireless propagation channel models. This section describes models implemented in this generator.

1) Symbols

First of all, the wireless communication model between an emitter and a receiver requires creating a signal for transmission. We consider that the binary sequence is modulated by QAM symbols and then followed by an Orthogonal Frequency-Division Multiplexing (OFDM) modulation with subcarrier-based pilot insertion. OFDM modulation is massively used in standard communication, particularly in the RFF database as it is shown in Table 1 and the signal varies greatly in amplitude which makes it interesting for the analysis of non-linear imperfections. While this paper mainly focuses on an OFDM transmission similar to a WiFi communication system, we have implemented and studied a single carrier transmission case to present the flexibility of the RiFyFi_VDG in which the binary sequence is modulated by QAM symbols, followed by single carrier frequency modulation.

2) Transmitter impairment models

In this section, the objective is to detail the impairment models behind (2). The impairments modeling is described in Figure 5 and is based on SoA models. We consider here a classic Zero Intermediate Frequency or homodyne modulation stage with I/Q modulation. The signal is multiplied by a carrier frequency generated from a LO and different impairments occur in the transmission chain. As Zhang et al. [33] this study is focused on the main features: CFO impairments, gain and phase IQ imbalance, Phase Noise (PN), and Power Amplifier (PA) nonlinearity. The model of each impairment is described hereafter.

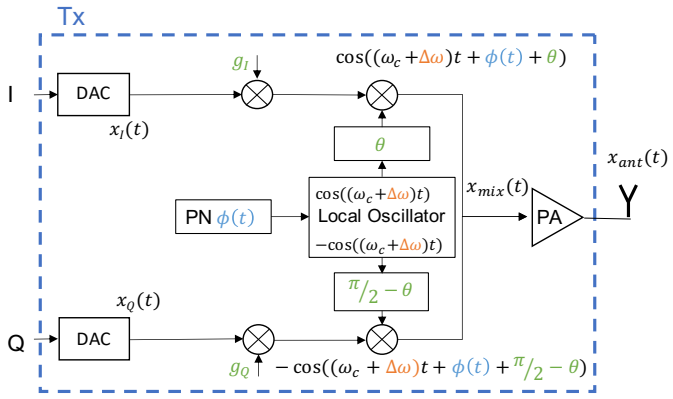

FIGURE 5. **Transmitter chain architecture and impairments.**

Before the LO, the analytical signal is modeled as:

$$\underline{x}(t) = x_I(t) + jx_Q(t), \tag{5}$$

where x_I and x_Q represent the real part and imaginary part of the complex signal $\underline{x}$. All complex variables will be underlined in the rest of the paper. The LO allows modulating the signal to a carrier frequency noted f_c, this modulation may create three different impairments. The first one is called CFO, then the LO is polluted by a phase noise and an imbalance between the two branches, called IQ imbalance.

CFO impairments: The LO modulates the signal at the ideal carrier frequency, f_c. However, CFO impairments introduce a frequency offset Δf, resulting in the effective carrier frequency, f_0, noted as:

$$f_0 = f_c + \Delta f. \tag{6}$$

For brevity, the models are expressed in terms of angular frequency with $\omega_* = 2\pi f_*$. and the modulated signal $x_{mix}(t)$ is expressed in terms of gain and phase error by:

$$x_{mix}(t) = x_I(t)\cos((\omega_c + \Delta\omega)t) - x_Q(t)\sin((\omega_c + \Delta\omega)t), \tag{7}$$

which can be equivalently written as:

$$\underline{x}_{mix}(t) = \underline{x}(t)e^{j(\omega_c+\Delta\omega)t}, \tag{8}$$

$$x_{mix}(t) = \Re(\underline{x}_{mix}(t)), \tag{9}$$

where $\Re$ stands for the real part of the complex number.

IQ imbalance impairments: In the presence of imbalance, the LO can be expressed according to Figure 5 in the form:

$$\underline{X}_{LO}(t) = g_I\cos(\omega_0 t + \theta) + jg_Q\cos(\omega_0 t + \frac{\pi}{2} - \theta), \tag{10}$$

$$\underline{X}_{LO}(t) = g_I\cos(\omega_0 t + \theta) + jg_Q\sin(\omega_0 t - \theta),$$

where θ is the phase impairment considering balanced, and g_I and g_Q the gain impairments. The expression can be simplified as done by Valkama et al. [43]:

$$\underline{X}_{LO}(t) = K_1 e^{-j\omega_0 t} + K_2 e^{j\omega_0 t}, \tag{11}$$

where $K_1 = \dfrac{g_I e^{-j\theta} + g_Q e^{j\theta}}{2}$, $K_2 = \dfrac{g_I e^{j\theta} - g_Q e^{-j\theta}}{2}$.

The signal $\underline{x}_{mix}(t)$ at the output of the LO in the presence of IQ imbalance could be expressed:

$$\begin{aligned}\underline{x}_{mix}(t) &= \underline{x}(t) \times \underline{X}_{LO}(t), \\ &= \underline{x}(t)K_1 e^{-j\omega_0 t} + \underline{x}(t)K_2 e^{j\omega_0 t}.\end{aligned} \tag{12}$$

In our model, as it is often done in the SoA, a balanced IQ mismatch is considered with $g_I = g_Q = \frac{g}{2}$.

Phase noise impairments: The PN has been modeled in the literature with different models, like Gaussian, Wiener or Lorentz and we focus on the Wiener model as it is a commonly used case in the literature to model free oscillator [19]. The LO PN $\phi(t)$ may be modeled [44]:

$$\phi(t) = \sqrt{c}B(t), \tag{13}$$

where $B(t)$ denotes a standard Wiener process and parameter c describes the LO quality called diffusion rate [44]. $B(t)$ is defined as $B(t_2) - B(t_1)$ with t_1 and t_2 correspond to the duration that forms the noise of variance $\sqrt{t_2 - t_1}\mathcal{N}(0,1)$. In the rest of the paper, we consider the digital Wiener PN model parameterized by its state noise variance σ_ξ^2 [45].

Considering all impairments described from now, the output of the LO that is $\underline{x}_{mix}(t) = \mathcal{F}_{\mathrm{LO}}(x(t))$, could be expressed by:

$$\underline{x}_{mix}(t) = \underline{x}(t)K_1 e^{-j(\omega_0 t + \phi(t))} + \underline{x}(t)K_2 e^{j(\omega_0 t + \phi(t))}. \tag{14}$$

Power amplifier impairments without memory: At the end of the transmission chain, the PA amplifies a low-power signal to a higher-power one. To model the memoryless nonlinear effect of the power amplifier in our system, the Saleh model used in SoA [33] is chosen. The non-linearity is modeled as amplitude/amplitude (AM/AM) denoted $A(t)$ and amplitude/phase (AM/PM) distortions denoted $\xi(t)$.

$$\begin{aligned}A(t) &= \frac{\alpha_{AM}|\underline{x}_{mix}(t)|}{1 + \beta_{AM}|\underline{x}_{mix}(t)|^2}, \\ \xi(t) &= \frac{\alpha_{PM}|\underline{x}_{mix}(t)|^2}{1 + \beta_{PM}|\underline{x}_{mix}(t)|^2},\end{aligned} \tag{15}$$

where $|.|$ denoted L1 norm. $\alpha_{AM}, \alpha_{PM}, \beta_{AM}, \beta_{PM}$ are the parameters of Saleh model [33].
Finally, the signal $\underline{x}_{PA}(t)$ after the PA is modeled as:

$$\underline{x}_{PA}(t) = A(t)e^{j(\angle \underline{x}_{mix}(t) + \xi(t))}, \tag{16}$$

where $\angle$ represent the angle of $\underline{x}_{mix}(t)$.
Power amplifier impairments with memory: The power amplifier can be modeled by memory model, the signal $\underline{x}_{PAM}(t)$ after the PA is modeled as [46]:

$$x_{PAM}(t) = \sum_{\substack{p=1 \\ p\ odd}}^{P} f_p(t) * (|\underline{x}_{mix}(t)|^{p-1}|\underline{x}_{mix}(t)|), \tag{17}$$

where P is the nonlinearity order of the model and $f_p(t)$ denotes the pth-order response of the polynomial model.

3) Notes on impact of carrier frequency:
The models proposed and used in our database generator are carrier frequency independent, but the parameterization of the model will depend on the carrier frequency. For example, the CFO depends on the carrier frequency following:

$$\Delta F_{max} = \frac{ppm}{10^6} f_c, \tag{19}$$

where ppm corresponds to the oscillator precision in part per million. For instance, a precise oscillator (Temperature Compensated X Oscillator, or an oscillator whose frequency is controlled by digital/analog compensation) at 0.13 *ppm* as chosen in the paper corresponds to a CFO of 300 Hz at 2.4 GHz. All these models are implemented in RiFyFi, the value of each parameter as well as the similarities between devices is discussed in the next sections.

4) Channel models
The channel models implemented in our database generator are multipath fading channel models. Two different models are specified with different delay profiles. The models are the Extended Vehicular A (EVA) model and the Extended Typical Urban model (ETU) [45]. EVA model represents a medium delay spread environment while the ETU model is a low delay spread environment. The signal obtained after the channel is modeled as:

$$\underline{x}_{channel}(t) = \underline{x}_{PA} * h(t) + n(t), \tag{18}$$

where $*$ is the convolution operator, $h(t)$ represents the propagation channel and, $n(t)$ is a Gaussian additive white noise of 30 dB.

IV. Practical use: from model to scenarios

A. Database generation parameters

In this section, the implementation and use of the RiFyFi_VDG are characterized, to create interesting databases to analyze. The generator offers the possibility to explore easily RFF identification scenarios, thanks to parametrization.

For all database creations, it is possible to choose:

- the number of transmitters N_{Tx},
- the similarities or dissimilarities between RFF emitters,
- the number of signals per device N_{signals}, in the database and used to train our network,
- the size of a signal $ChunkSize$
- the transmission symbols scenario
- the modulation
- the fingerprint scenario
- the channel scenario

As the SoA shows, the RFF identification conditions are multiples such as the frame of data used to identify the transmitter [36], the level of noise [24], the number of signals, the number of propagation channel views, the

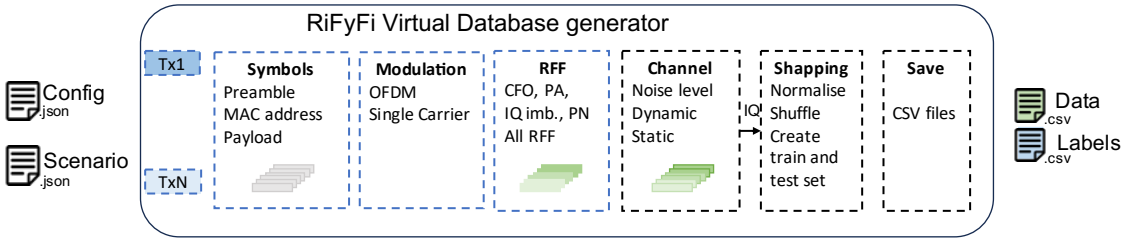

FIGURE 6. **Parametric database generator chain.**

number of transmitters and the quality of them. Therefore, exploring these different scenarios with a single framework seems interesting and could help in designing a real database.

In RiFyFi, a signal *scenario* consists of key parameters that can be separated into 3 aspects: symbol generation, RFF, and propagation channel. The first one, the symbol, represents the part of the signal used for identification: Preamble, MAC address, or Payload (Pre, MAC, Pay). This is represented by the *Symbols* block in Figure 6. Then, it is possible to choose the type of modulation between at least three possibilities, OFDM, single carrier, or LoRa. Then, the *RFF* block defines the transmitter impairments considered: CFO, PN, PA, IQ imbalance, or all impairments. After the transmitter model description, the *channel* block defines environmental conditions, such as noise or channel model. Finally, it is possible to add a receiver model with its own RFF. Note that in this paper the impact of the receiver will not be explored and is left for future studies. In the end, a large matrix of IQ samples of size ($ChunkSize$, 2, $N_{\text{signals}} \times N_{\text{Tx}}$) is created and passes through a shaping block where data is shuffled and split to create, training and test sets. Then the database and the labels are saved in 4 CSV files to be used by the network. The labels matrix is composed of 2 dimensions ($N_{\text{Tx}}, N_{\text{signals}}$). The matrix contains only 0 and 1. The 1 allows us to attribute the signal to the corresponding transmitter.

In the context of this work, is followed the conventional approach found in the literature, which involves feeding the network with complex IQ signals, as it is done in [14] and [18].

B. Symbols Scenario and Modulation

A *signal* is a sequence of $ChunkSize$ complex IQ samples. In this paper 256 IQ samples are considered corresponding to the input size of the network [28], [30]. The baseline of the signal is created with OFDM symbols composed of 548 IQ samples with an FFT size of 512 and a cyclic prefix size of 36. This database generator can simulate different scenarios: Preamble, MAC address, and Payload.

Creating a Preambule-based database requires generating the same sequence of symbols for all emitters. It can be a specific data sequence or a special sequence such as Zadoff-Chu sequences. The second possibility is to generate a unique sequence for each transmitter. This scenario is close to a MAC address scenario where the signal contains the MAC address. The last possibility is to generate different sequences for each transmission, this scenario is called Payload where the identification can only be done through payload.

To simulate these scenarios, the generator creates a sequence called a burst composed of 64 *signals* of 256 IQ samples, this burst is repeated to complete the N_{signals}. The preamble is the same for each burst and each transmitter. In the MAC address scenario, a different burst is defined for each transmitter. Finally, in the Payload scenario, all bursts are different.

C. Fingerprint Scenario

Contrary to the symbols scenario, the fingerprint scenarios are not all realistic, but they offer exploration possibilities. First of all, it is possible to activate one or multiple impairments, to combine their effects. Seven scenarios are created and explored:

- CFO: only CFO impairment
- Imbalance: only gain and phase IQ imbalance impairment
- PN: only PN impairment
- PA: PA impairment with Saleh model
- PA_memory: PA with measured memory model
- All_impairments: CFO, imbalance, PA with Saleh model and PN.

All scenarios are studied in this paper. First, in section V impairments are independently studied, and then the most realistic scenario is addressed: All_impairments.

The impairment models described in Section 2 are parametric models. Each impairment is defined by one or multiple parameters. The gap between the values of an impairment parameter for two transmitters determines the RFF similarity between two transmitters. Other papers propose to create a grid to make sure that the space between two impairment values is sufficient. In Zhang paper [33], the impairments follow a uniform random distribution within an interval. This paper presents the study of the impact of the space between two impairment values with different similarity percentages around a mean value.

D. DL Network

The flexible nature of the framework offers the possibility to explore different network architectures or other classifi-

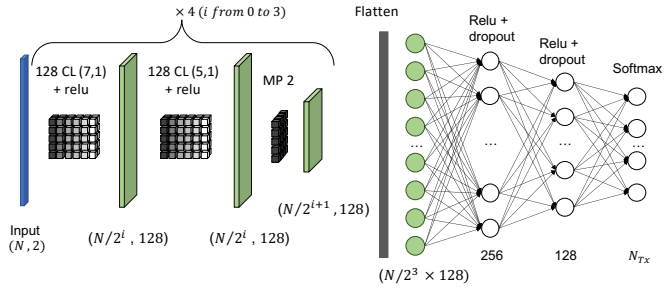

FIGURE 7. **Deep Learning architecture.**

cation solutions. In this paper, the exploration of network architecture to determine the best for various impairments is not undertaken. The network chosen in this paper has been proposed and studied for RFF identification in [15], which is one of the first contributions on RFF with DL solution and public database. It is a CNN inspired by AlexNet, with 4 convolutional layers, each layer is composed of two blocks of 128 filters size 7×1 and 5×1 and a max-pooling stage. The chosen activation function is ReLu and the optimizer is Adam with γ learning rate. The batch size has been empirically explore and set as 64. After the 4 convolutional layers, the CNN has 3 fully connected layers with 256 nodes, 128 and the number of classes (in this case 6 classes). After the two first fully connected layers a dropout layer is added with dr $\in [0; 1]$. In input, the network takes complex baseband signals without pre-processing. These signals are split into two raws I and Q, and the input size N corresponds to $ChunkSize$. For N = 256 as done in [28], [30], this architecture is presented in Figure 7 and has 1,232,774 parameters. In Figure 7, the notation corresponds to $N_{\text{Filters}}CL(a, b)$ for Convolutional Layers with (a,b) the size of the filter, and *MP 2* for the Maxpooling 2.

The next section presents an exhaustive investigation of the individual impact of impairments with the network introduced in [15] aimed to reveal the most discriminant impairments for this network. The selection of an [15]-like network is based on several studies indicating that networks composed of convolutional and fully connected layers have demonstrated strong performance in RFF classification tasks [12], [15], [22], [24]. Conversely, it is important to note that other architectures such as transformers [25] or Tangled Program Graph [30] have also shown reliable performance.

V. Investigation of the individual impact of impairments

In this section, impairments are separately studied with different confidence intervals around a fixed mean value, inspired by the SoA [33] and defined in Table 2. In [12], Soltani et al. propose to create 10 virtual transmitters, and they vary the amplitude imbalance from 1 to 5.5 dB with steps of 0.5 dB and phase imbalance from 1° to 82° with steps of 9°. This simulation seems not realistic. Zhang et al. [33] set the range of gain and phase imbalances to [-1 1] dB and [-5 5] degree, which seem to be more realistic values. For the PA they used the values presented in Table 2 which vary within $\pm 5\%$. In this paper, different intervals are explored. The confidence interval is a metric to model the quality of transmitters. The RFF identification complexity depends on the similarities between the RFF transmitters. For a given number of transmitters, a large interval reduces the similarity between two transmitters. However a small interval increases the RFF similarity, and, therefore, it makes the identification difficult. For this study, some learning parameters are empirically adjusted upstream for each impairment in order to compare them in favorable situations. The parameters are specifically the dropout and the learning rate. For this study, we chose the number of transmitters as a function of the number of impairment parameters we have to explore, 2 transmitters are not enough to explore 2 or 4 parameters. The results presented in this section are obtained by means of 5 different seeds, the different colors in the tables evaluated the performance.

TABLE 2. **Mean value chosen for impairment parameters.**

Impairment	Parameters	Mean value
CFO	Δf	300 Hz
Imbalance	$\bar{g_Q}$	1.5 dB
	$\bar{\theta}$	2.5°
PN	σ^2_ϕ	10^{-7}
PA	$\bar{\alpha_{AM}}$	2.1587
	$\bar{\beta_{AM}}$	1.1517
	$\bar{\alpha_{PM}}$	4.0033
	$\bar{\beta_{PM}}$	9.104

A. CFO

To study the CFO impairment, we set the mean value at 300 Hz and create different similarities with p% for two transmitters with the following CFO values:

$$\Delta f_{Tx1} = \bar{\Delta f}(1 - p\%), \tag{19}$$

$$\Delta f_{Tx2} = \bar{\Delta f}(1 + p\%), \tag{20}$$

with $p = 5\%$, 2%, 1% and 0.5%. The CFO values of both transmitters for each similarity scenario p are given in Appendix (Table 17).

Figure 8 presents the F1 score evolution during the training phase and Table 3 summarizes the results with the mean F1 score obtained on the training set and test set at different epochs. For the next impairments, and for the sake of conciseness we only use tables to present results.

TABLE 3. **Mean F1 score evolution during training phase for different CFO impairments, $\gamma = 10^{-4}$, 2 transmitters and 900 signals per transmitters for train.**

F1 score at	20 epochs		45 epochs		90 epochs		125 epochs	
p	Train	Test	Train	Test	Train	Test	Train	Test
5%	98%	98%						
2%	94%	93%	98%	98%				
1%	85%	81%	94%	90%	99%	95%		
0.5%	57%	52%	86%	80%	91%	86%	99%	92%

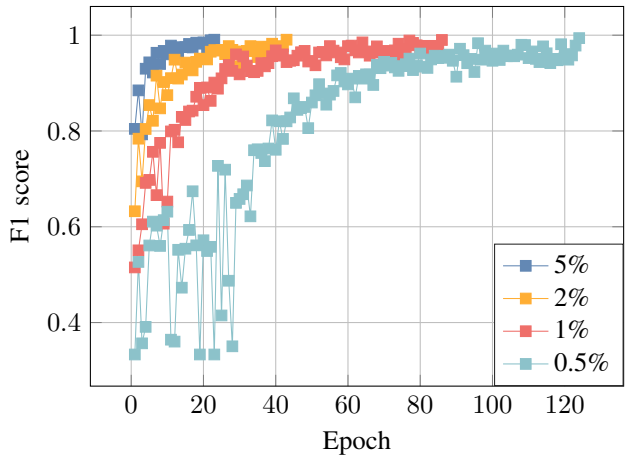

FIGURE 8. **Evolution of the F1 score of the different CFO impairments values, 2 transmitters and 900 signals per transmitters for train.**

Results are obtained with a learning rate $\gamma = 10^{-4}$ and no dropout and they show that narrowing the impairment interval between two transmitters increases the network's difficulty in learning how to distinguish between these transmitters. Nevertheless, this is compounded by the fact that numerous studies have demonstrated the instability of the CFO, which further exacerbates the situation we will not focus on this. The study of CFO highlights the link between RFF transmitters similarity scenario and the capacity of the network to separate transmitters.

B. IQ imbalance

As done with the CFO, we explore different similarity configurations for IQ imbalance, defined by g_Q, g_I and θ with $g_Q = -g_I$ and $\theta \in \{\theta_{min}, \theta_{max}\}$ with 4 transmitters.

$$g_{QTx1} = g_{QTx3} = g_{\bar{Q}}(1 - p\%) \quad (21)$$

$$g_{QTx2} = g_{QTx4} = g_{\bar{Q}}(1 + p\%) \quad (22)$$

$$\theta_{Tx1} = \theta_{Tx2} = \theta_{max} \quad (23)$$

$$\theta_{Tx3} = \theta_{Tx4} = \theta_{min} \quad (24)$$

The two impairments, gain and phase have been explored together by testing all combinations, with p equal 10%, 5%, 3% and 1% and the ensemble $\{\theta_{min}, \theta_{max}\}$ takes $\{0°, 5°\}$, $\{1°, 4°\}$ and $\{2°, 3°\}$. Table 18 in the Appendix presents gain and phase values for each transmitter. The results are obtained without dropout and a learning rate at 10^{-4}.

Table 4 presents F1 score values at different times for the different impairment combinations. Comparing the first rows of results with a 10% similarity scenario shows that increasing the phase similarity from $\{0°, 5°\}$ to $\{1°, 4°\}$ increases the number of epochs required for the network to converge. Moreover for $\{2°, 3°\}$, the performance in test falls down even a after long training. Then, comparing the first row of the 10% similarity scenario and the first row of the 3% one shows a slight difference in F1 score value at the same time. Moreover in combination with $\{2°, 3°\}$ and gain over 3%, results show an over-learning on training

TABLE 4. **Mean F1 score evolution during training phase for different IQ imbalance impairments, $\gamma = 10^{-4}$.**

F1 score at	85 Epochs		130 Epochs		210 Epochs	
	Train	Test	Train	Test	Train	Test
g: 10% [0°;5°]	92%	88%				
g: 10% [1°;4°]	66%	61%	95%	80%		
g: 10% [2°;3°]	44%	42%	48%	44%	84%	48%
g: 5% [0°;5°]	94%	91%				
g: 5% [1°;4°]	69%	66%	89%	79%		
g: 5% [2°;3°]	58%	49%	67%	48%	96%	50%
g: 3% [0°;5°]	90%	87%				
g: 3% [1°;4°]	70%	64%	86%	76%		
g: 3% [2°;3°]	58%	49%	58%	48%	86%	53%
g: 1% [0°;5°]	63%	55%	84%	55%		
g: 1% [1°;4°]	30%	28%	69%	55%	91%	51%
g: 1% [2°;3°]	12%	12%	13%	13%	70%	18%

data as it stops around 50% on Test data. The analysis of the confusion matrix in Table 5, under 10% and $\{2°, 3°\}$ similarity conditions, reveals an effective classification of Tx1 with 89% of correct classification. However, it exhibits confusion between Tx1 and Tx3, as well as Tx2 and Tx4. In summary, a $1°$ gap between two transmitters is insufficient for a clear differentiation.

When the IQ imbalance gain is set below 1%, during training, the network tends to over-learn and stops at around 50%. This is confirmed by the confusion matrix in Table 6, which highlights a classification issue, as the network only seems to recognize two classes.

The study of IQ imbalance shows a decrease in convergence speed when the similarity between impairments decreases for the gain and phase with a limit for recognizing devices at 1% for gain and at $1°$ difference for phase.

TABLE 5. **Confusion Matrix for test data for IQ imbalance impairment, g: 10% and $[2; 3]$ combination.**

Guess / True	Tx_1	Tx_2	Tx_3	Tx_4
$TxTrue_1$	89.0	0.0	11.0	0.0
$TxTrue_2$	0.0	52.0	0.0	48.0
$TxTrue_3$	82.0	0.0	18.0	0.0
$TxTrue_4$	0.0	48.0	0.0	52.0

TABLE 6. **Confusion Matrix for test data for IQ imbalance impairment, g: 1% and $[2; 3]$ combination.**

Guess / True	Tx_1	Tx_2	Tx_3	Tx_4
$TxTrue_1$	67.0	0.0	18.0	15.0
$TxTrue_2$	66.0	0.0	17.0	17.0
$TxTrue_3$	67.0	0.0	19.0	14.0
$TxTrue_4$	65.0	0.0	18.0	17.0

C. Phase Noise

The PN is a particular impairment because, as it is a noise, it is difficult to find the specific difference between transmitters only based on PN. To study the PN, different PN values (between 10^{-7} and 10^{-4}) are set for 4 different transmitters and experiments have been done with different learning rates and dropout. However, the results are always bad: a F1 score around 25% is achieved on the test set, even after a large

number of epochs. This result shows that the network is not able to separate the transmitters. To conclude, the PN is not a relevant impairment to separate transmitters.

D. PA model

To study the PA effect, two types of model introduced in Section 2 are used. The first one is the Saleh model, and the second is the memory model based on coefficients from real measurements. This memory model (MM) is presented by [47]. The memory model contains around 100 emitters models but we extract 2 groups of 4 PA models to better stress the impact of closed PA configurations "MM close" and distinct PA configurations "MM far". For the Saleh model, the different parameters are presented in Table 19 of the Appendix for different impairment similarities expressed as:

For α_{AM} and α_{PM},

$$\alpha_{Txi} = \alpha_{Txi+2} = \alpha(1-p\%) \text{ with i = 1,} \quad (25)$$

$$\alpha_{Txi} = \alpha_{Txi+2} = \alpha(1+p\%) \text{ with i = 2.} \quad (26)$$

For β_{AM} and β_{PM},

$$\beta_{Txi} = \beta_{Txi+1} = \beta(1+p\%) \text{ with i = 1,} \quad (27)$$

$$\beta_{Txi} = \beta_{Txi+1} = \beta(1-p\%) \text{ with i = 3.} \quad (28)$$

Different experiments are done and present instability of the network during the training phase. To reduce this problem the dropout is put at 0.25 and the learning rate is decreased at 10^{-5}. Table 7 presents the F1 score value during training for the train dataset and test dataset. This shows a decrease in convergence speed when the similarity between impairments increases and for $p \leq 0.5\%$ the network overlearns on training data. The use of the memory model allows us to show the flexibility of our framework in particular the interest of the generator is to use any RFF parametric models. Finally, it shows that the results obtained with the Saleh model are realistic in terms of convergence speed.

E. Conclusion of individual impairment effects

The investigation of the individual impact of impairment reveals the link between the RFF impairments similarity and the capacity of the network to classify several devices. The impairments are not all relevant, in particular, the PA and IQ imbalance seems to be interesting. This study shows the importance of tuning learning parameters to adapt the network to the data. Moreover the network seems to converge faster for the CFO and IQ imbalance.

TABLE 7. Mean F1 score evolution during training phase for different Power Amplifier impairments, $\gamma = 10^{-5}$ dr = 0.25.

F1 score at	160 epochs		500 epochs		970 epochs	
	Train	Test	Train	Test	Train	Test
5%	98%	99%				
2%	74%	70%	96%	95%		
1%	28%	26%	80%	59%	95%	76%
0.5%	24%	23%	50%	38%	81%	49%
0.3%	25%	23%	44%	24%	87%	23%
MM far	86%	87%	91%	92%	93%	93%
MM close	33%	31%	57%	49%	74%	54%

VI. Conglomerate scenarios study

In this section, different transmission scenarios with all impairments are studied with $N_{Tx} = 6$ transmitters and p% interval, for IQ imbalance, CFO and PA (Saleh model). For the PN two variances of state noise center around $10^{-7}$and 10^{-4} are explored. The values chosen for each parameter of the 6 transmitters are calculated following (31), Table 20 in Appendix presents the parameter values calculated for $p = 5\%$ similarity. Except for θ, the parameter values $P^p_{Tx_k}$ for device $k \in [1, N_{Tx}]$ are computed as:

$$P^p_{TX_k} = P^p_{min} + k\frac{(P^p_{max} - P^p_{min})}{N_{Tx}}, \quad (29)$$

$$\text{with } P^p_{min} = \text{Mean Value}(1-p\%), \quad (30)$$

$$P^p_{max} = \text{Mean Value}(1+p\%), \quad (31)$$

with P^p_{min} the minimum of the impairment parameter in the p% similarities scenario and P^p_{max} the maximum. Four different similarity scenarios are studied in this paper, 5%, 3%, 2% and 1%. For θ parameters, P^p_{TXk} follows (31) but P^p_{min} and P^p_{max} depend of the similarity scenario. For $p = 5\%$ and 3%, we set:

$$P^p_{min} = 0^\circ \qquad P^p_{max} = 5^\circ, \quad (32)$$

while for $p = 2\%$ and 1%, we set:

$$P^p_{min} = 1^\circ \qquad P^p_{max} = 4^\circ. \quad (33)$$

A. Preamble scenario

1) How close can the RFF of 6 devices be?

This section addresses the convergence speed of the CNN in preamble scenarios with all impairments and different contexts. The databases are composed of 6 emitters with 1000 WiFi-like signals per emitters, with OFDM modulation. Each database is split into 90% and 10% to create training and test sets, respectively. Table 8 presents the F1 score values during training for both training and test sets, for the different similarity scenarios. The training is ended when the network obtains an F1 score of 98% on the training set. First, at 5% similarity, two state noise variances of the PN scenarios are studied, 10^{-7} and 10^{-4}. The results indicate that increasing the state noise variance enhance both classification and generalization challenges due to the additional noise introduced to the signal. At 30 epochs the network as reached 98% for test in 10^{-7} scenario, but for 10^{-4} the network obtain only 93% on test. The results are interesting and present good performance for both PN scenarios with the worst result in the test for 10^{-4} as the first study shows the PN was not relevant but could disturb the network by adding noise and making the identification difficult. For the rest of the paper, phase noise is set around 10^{-7}.

Then different similarity scenarios: 5%, 3%, 2% and 1% are compared. The network has no difficulty in classifying

TABLE 8. Mean F1 score evolution during training phase for Preamble and different similarity scenarios, $\gamma = 10^{-4}$ dr = 0.

F1 score at		30 epochs		60 epochs		280 epochs	
PN	p	Train	Test	Train	Test	Train	Test
10^{-4}	5%	96%	93%				
10^{-7}	5%	98%	98%				
	3%	48%	45%	96%	85%		
	2%	32%	25%	43%	19%	85%	30%
	1%	31%	17%	45%	17%	84%	18%

TABLE 9. Mean F1 score evolution during training phase for Preamble and different similarity scenarios, $\gamma = 10^{-5}$ dr = 0.25, PN=10^{-7}.

F1 score at	280 epochs		500 epochs		1100 epochs	
p	Train	Test	Train	Test	Train	Test
2%	42%	17%	73%	30%	97%	50%
1%	41%	16%	76%	19%	99%	17%

the 6 transmitters in the 5% scenario. As the similarity increases, the network needs more time to learn and classify the devices. The complexity of the classification problem increases as device impairments become closely situated, making it more challenging for the network to distinguish between them. To solve this issue it is possible to change some learning parameters such as the learning rate and add dropout to avoid overfiting, as it is presented in Table 9. In the 2% similarity scenario, these changes improve the F1 score in the test but not enough. Furthermore, in the case of the 1% similarity scenario, the test F1 score remains at approximately 18%, close to random value $1/N_{Tx}$. This suggests that the network struggles to learn RFF due to the proximity of impairments and just overfits on the training set. We propose to increase and explore the number of signals per transmitter required to improve the performance of the network and avoid overfitting. Figure 9 presents the F1 score obtained in test as a function of the number of signals in the training dataset. The network obtains an F1 score of 80% in the test when 9000 signals per transmitter are used in the training dataset for a 1% similarity. The number of required signals to train the network increases with the similarity between devices. It is thus more difficult for the network to separate and classify them. This reveals a countermeasure to RFF identification by using emitters with very similar impairments.

The Preamble scenario over-fits on the data: for another preamble used in the test, the network obtained around F1 score of 25% and is not able to identify the RFF in other data contexts. However, if the identification application uses only the preamble to identify the device, over-learning in those conditions gives the guarantee that the neural network will perform in this situation.

2) How about the impact of signal modulations?

This subsection addresses the diversity of signal modulations by considering single carrier frequency modulation. For this, a QAM sequence is upsampled and filtered by a square root raised cosine filter with a roll-off of 0.33. The QAM

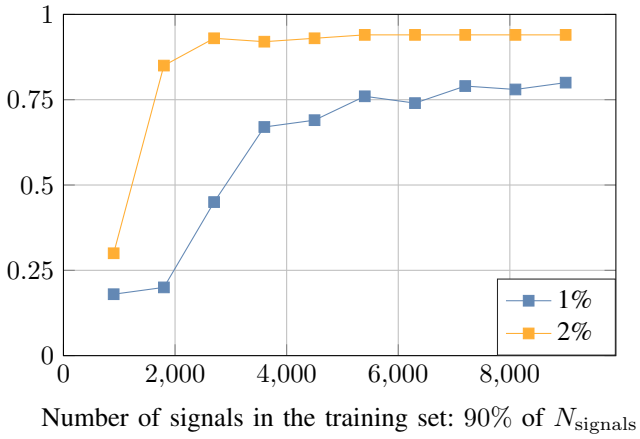

FIGURE 9. F1 score obtained in test in function of the number of signals used to train the network when training has reached 98% of F1 score, in Preambule scenario with $\gamma = 10^{-4}$ dr = 0.

sequence is the same for all transceivers (in preamble mode) and set to have the same length as the OFDM sequences.

The results presented in Table 10 are very close to the results presented in the previous Table 8. The convergence speed is comparable to the convergence speed obtain with OFDM and decrease when the similarity between devices increase. It is important to notice that our simulator readily accommodates additional modulation schemes or even standard-compatible signals. It paves the way for specialized analysis focused on standards or applications beyond the scope of this paper.

TABLE 10. Mean F1 score evolution during training phase for Preamble and different similarity scenarios for single carrier modulation, $\gamma = 10^{-4}$ dr = 0.

Modulation	F1 score at	35 epochs		60 epochs		400 epochs	
		Train	Test	Train	Test	Train	Test
Single Carrier	5%	98%	95%				
	3%	64%	59%	98%	81%		
	2%	15%	10%	33%	10%	98%	24%

3) What is the most relevant feature ?

To study the most relevant impairment, we choose to use the 1% similarity scenario and increase to 10% one after one the interval for one impairment. At 10% in the previous section, all individual impairments allow separating transmitters. Here we study the co-existence of all impairments and explore how they interfere together and impact the classification accuracy. Table 11 presents the results obtained in test when the network has reached 98% F1-score on train set for differnet situation. The best performances are obtained when the PA is set to 10% and reveals the importance of PA in RFF identification.

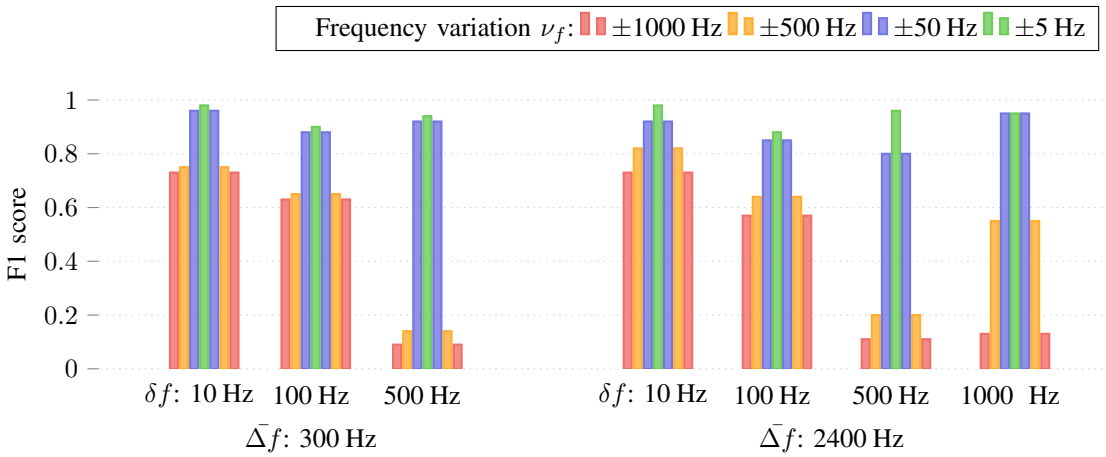

FIGURE 10. **F1 score obtained in test in function of the CFO dispersion values δ_f for two different $\bar{\Delta f}$, and 6 transmitters.**

TABLE 11. **F1 score obtained in test when training has reached 98% for different RFF at 10%, $\gamma = 10^{-5}$ dr = 0.25.**

CFO	IQ imbalance	PA	F1 score Test
10%	1%	1%	20%
1%	10%	1%	34%
1%	1%	10%	94%

4) Does the dynamic CFO impact the classification ?

The previous result highlights the most relevant features and Table 11 shows that the CFO is not really impacting in our context. In Section 3, we presented and chose a precise oscillator: an oscillator with a compensation system with 0.13 *ppm*, and fixed value. In this condition, the CFO does not impact the classification. However, the state of the art extensively covers this topic and leads to the conclusion that the CFO has a significant impact on RFF identification [19], [32], [48], [49]. Considering this point, we propose a simulation to study the impact of the dynamic CFO on the classification. First, two different mean values $\bar{\Delta f}$ of the CFO are chosen: 300 Hz and 2400 Hz, corresponding to around 0.1 *ppm* and 1 *ppm* respectively, both at 2.4 GHz. In these two scenarios, we consider different dispersion scenarios called δ_f, which corresponds to the CFO difference between 2 devices and that is expressed as:

$$\Delta f_{Txi+1} = \Delta f_{Txi} + \delta_f, \tag{34}$$

where Δf_{Txi} and Δf_{Txi+1} correspond to the CFO impairment for transmitters i and $i+1$ during the training phase.

Figure 10 presents the F1 score obtained in the test when the CFO has shifted between the training phase and test phase for different dispersion scenarios. This shift, called frequency variation, and noted ν_f, can model the impact of a temperature variation and is expressed as:

$$\Delta f_{Test_{Txi}} = \Delta f_{Train_{Txi}} \pm \nu_f, \tag{35}$$

where $\Delta f_{Test_{Txi}}$ and $\Delta f_{Train_{Txi}}$ correspond to the CFO value of transmitter i during the training or test phase.

The first part of Figure 10 concerns a mean CFO value at 300 Hz with three dispersion values: 10 Hz, 100 Hz and 500 Hz between each transmitter. For a dispersion δ_f of 10 Hz, the result shows that an important CFO variation such as $\nu_f = \pm 1000$ Hz, in red, between the training and the test set, affects the classification accuracy but the network is still able to classify many signals (around 75%). In other words, the CFO is too weak to be a relevant impairment for the network. For $\delta_f = 500$ Hz, the results are different. For a $\nu_f = 500$ Hz or 1000 Hz, accuracy falls down, which means that the network associates the transmitter to a particular CFO value. This reveals the importance of the CFO dispersion in this scenario to classify the transmitters. In other words for 500 Hz dispersion, the CFO is a relevant impairment for the network. In this case, a CFO variation due to temperature can affect dramatically the identification.

Finally, for $\bar{\Delta f} = 2400$ Hz, the conclusions are the same that the ones done with $\bar{\Delta f} = 300$ Hz. For $\delta_f = 1000$ Hz, the orange bar, which corresponds to $\nu_f = 500$ Hz, has reached 50%. This occurs because the network's decision boundary is positioned midway between two CFO values. As a result, 50% of the sequences are correctly classified, while the other 50% are classified into the nearest class. In the rest of the paper, we keep the parameterization of a precise oscillator: 0.13*ppm* (temperature compensated X oscillator, or an oscillator whose frequency is controlled by digital/analog compensation).

5) How the channel can impact the classification accuracy?

We propose to compare 3 training situations:

1) Static environment without a multipath channel, a SNR of 30dB,
2) Static environment with a multipath channel for each transmitter and a SNR of 30dB,
3) Dynamic environnement with several and different multipath channels for training and test with a SNR of 30dB.

TABLE 12. Channel study $\gamma = 10^{-5}$ dr = 0.25, for 6 transmitters, 900 signals pers transmitter and per channel for training.

	Multipath channel scenario		Similarity scenario		
Exp	Train	Test	5%	3%	2%
1	no channel	no channel	96%	91%	50%
2	static	static	94%	76%	46%
		dynamic	74%	50%	21%
3	dynamic	dynamic	98%	98%	95%

For the first scenario featuring a static environment without a multipath channel, where only noise is taken into account. The performance of the network is evaluated on the same scenario with 30 dB of noise. The results presented in Table 12 are obtained with the test dataset when the training has reached 98% of accuracy for 3 different similarities. In the second situation, the database corresponds to a wireless transmission with a fixed static multipath channel for each transmitter. The performance of the network trained with this database is evaluated in two situations, with the same static channel and with 100 different static channels for each transmitter. Finally, in the third situation, the training is done on multiple (10) multipath propagation channels and tested with 100 different static channels for each transmitter.

The results obtained show the channel sensitivity. In particular, the third row with static training and dynamic testing scenario shows that the network learns the channel, this result has been already shown in the literature. The last row demonstrates that exposing the network to various propagation channel conditions during training enhances its robustness to channel effects. In conclusion, a network trained with a static channel database cannot guarantee a generic capacity to detect the transmitters in other channel situations. In other words, a training database has to be according to the testing scenario space to guarantee good genericity.

6) How does the number of transmitters impact the classification ?

The number of transmitters is multiplied by two and the RFF impairment values are computed in 5% intervals around the mean values. In this situation, the network required on average 250 epochs to achieve 98% of accuracy in training. Compared with the 6 transmitters situation, the network requires more epochs to converge because the complexity of the problem has increased.

TABLE 13. Mean F1 score during training phase for 5% similarity and 12 devices and 6 devices with $\gamma = 10^{-4}$ dr = 0.

F1 score at	30 epochs		60 epochs		250 epochs	
N_{Tx}	Train	Test	Train	Test	Train	Test
6 Tx	96%	93%				
12 Tx	88%	86%	89%	87 %	98%	94%

B. MAC address scenario

In this section, we study the classification of 6 transmitters where the sequence emitted by the transmitter (training and test sets) contains a different MAC address per transmitter. After 6 epochs the network has reached 99% F1 score on the training set and 98% on the test set for the 5% similarities and 1% similarities scenario with a learning rate at 10^{-4} and no dropout. The confusion matrix given in Table 14 presents the result of classification in the test without MAC spoofing. Table 15 is obtained when the Tx1 spoofed the MAC address of Tx3. The spoofing represents a real risk in cybersecurity, it's possible to use the MAC address of another device to be identified as this device by an authentication system.

In the MAC address scenario, the address in the signal is the strongest signature and prevents the network from focusing on RFF, the learning system and the network only learn the MAC address to identify the device, in this situation the identification system will not be robust to spoofing. To tackle such issues, the virtual database allows exploring the scenario to determine a way to secure the transmission by slicing the signal [36].

TABLE 14. Confusion Matrix for test data in MAC scenario without spoofing.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
$TxTrue_1$	100.0	0.0	0.0	0.0	0.0	0.0
$TxTrue_2$	8.0	87.0	2.0	1.0	1.0	1.0
$TxTrue_3$	0.0	0.0	100.0	0.0	0.0	0.0
$TxTrue_4$	0.0	0.0	0.0	100.0	0.0	0.0
$TxTrue_5$	0.0	0.0	0.0	0.0	100.0	0.0
$TxTrue_6$	0.0	0.0	0.0	0.0	0.0	100.0

TABLE 15. Confusion Matrix for test data in MAC scenario with Tx1 spoofed MAC address from Tx3.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
$TxTrue_1$	0.0	0.0	100.0	0.0	0.0	0.0
$TxTrue_2$	8.0	87.0	2.0	1.0	1.0	1.0
$TxTrue_3$	0.0	0.0	100.0	0.0	0.0	0.0
$TxTrue_4$	0.0	0.0	0.0	100.0	0.0	0.0
$TxTrue_5$	0.0	0.0	0.0	0.0	100.0	0.0
$TxTrue_6$	0.0	0.0	0.0	0.0	0.0	100.0

C. Payload scenario

The Payload scenario is the most difficult one because all data are different. In this section, the number of signals required to obtain robust RFF learning in the Payload scenario is studied.

Such a scenario represents non-correlated data and is complex for the network, for example with 900 signals at 5% similarity the F1 score in the test stays around 30% compared to the Preamble situation where the network achieves 98% in the test. In the Payload scenario, the network overfits on training data, to avoid this issue the number of signals used in the train is explored. The results are obtained with a learning rate at 10^{-4} and no dropout. Figure 11 presents the F1 score obtained in the Test set when the network has reached 98% of the F1 score on the training set, obtained for 2 different similarity scenarios. In blue, we represent the 5% similarity scenario database, and yellow represents the 10% similarity

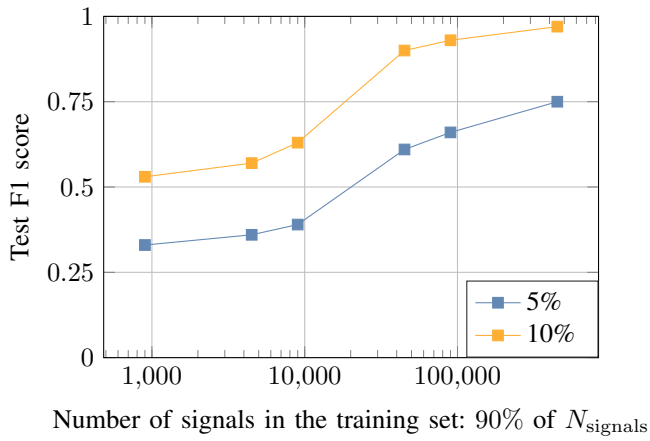

FIGURE 11. **F1 score obtained in test in function of the number of signals used to train the network when training has reached 98% of F1 score.**

TABLE 16. **Time required for the network to reach 98% of F1 score on training data, in Payload context, on GPU architecture**

Number of signals	5%	10%
900	27s	25s
4500	160s	110s
9000	8min	220s
45000	45min	19min
90000	2h	38min
450000	+10h	2h12

scenario. Table 16 completes the results by adding the time of training to reach 98% of the F1 score.

Figure 11 shows a great improvement of the F1 score in the test when the number of signals is increased. However, Table 16 presents the time required to achieve the different training and the time convergence speed increase because of the number of data seen in an epoch. The number of signals and the time of training can represent some limit depending on the application context. For example, in cyber defense, the quantity of data can be limited by the difficulty of capturing data and the time to train the network can be limited by the need of short reaction time.

VII. Conclusion

This paper introduces a virtual database generator based on wireless transmission and RFF models included in a flexible framework for RFF identification. This work proposes an exploration of database design for RFF identification with DL considering the similarity between the RFF of transmitters, the transmission scenario, and the number of signals.

Our analysis showed the impact of similarity between RFF transmitters on the network convergence speed and the F1 score performance in a preamble context. For a 2% similarity, decreasing the learning rate and adding dropout helped the network to improve the classification in the test from 30% to 45%. Moreover increasing the number of signals permitted to achieve 65% of accuracy. The CFO analysis is correlated with the SoA for a 500 Hz dispersion between transmitters the network focuses the training on this impairment. However, for 100 Hz dispersion between transmitters, the network does not focus the training on the CFO, so for a CFO variation of 1000 Hz the classification accuracy is affected but still around 75%.

A very large number of signals per transmitter is needed when the RFF similarities are strong between transmitters or in a Payload context consequently having similar RFF devices can be a countermeasure to avoid RFF identification. The virtual database generator can help to pre-assess the needed database design with many flexibility as it was shown by changing the OFDM modulation by a single carrier modulation. The authors commit to release this generator as an open-source tool immediately after acceptance of the paper [20].

APPENDIX

Below are presented the tables of all the parameters used in the various scenarios.

TABLE 17. **CFO values for different similarity scenarios.**

300Hz	Tx1	Tx2
CFO 5%	285	315
CFO 2%	294	306
CFO 1%	297	303
CFO 0.5%	298.5	301.5

TABLE 18. **Gain and phase impairments values for different IQ imbalance.**

	p%	Tx1	Tx2	Tx3	Tx4
Gain	g: 10%	1.350	1.650	1.350	1.650
	g: 5%	1.425	1.575	1.425	1.575
	g: 3%	1.455	1.545	1.455	1.545
	g: 1%	1.485	1.515	1.485	1.515
Phase	[0°;5°]	5°	5°	0°	0°
	[1°;4°]	4°	4°	1°	1°
	[2°;3°]	3°	3°	2°	2°

ACKNOWLEDGMENT

This work is funded by DGA and Brittany region under the Creach Lab founding and by the French National Research Agency (ANR) under the grant number ANR-22-CE25-0007-01 (RedInBlack project).

REFERENCES

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[2] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A comprehensive survey on radio frequency (RF) fingerprinting: Traditional approaches, deep learning, and open challenges," *Computer Networks*, vol. 219, p. 109455, 2022.

[3] N. Miguélez-Gómez and E. A. Rojas-Nastrucci, "RF Fingerprinting: Hardware-Trustworthiness Enhancement in the Hardware Trojan Era: RF Fingerprinting-Based Countermeasures," *IEEE Microwave Magazine*, vol. 24, no. 11, pp. 35–52, 2023.

[4] B. P. Muller, L. J. Wong, W. H. Clark, and A. J. Michaels, "A real-world dataset generator for specific emitter identification," *IEEE Access*, vol. 11, pp. 110 023–110 038, 2023.

[5] H. Gu, L. Su, W. Zhang, and C. Ran, "Attention is Needed for RF Fingerprinting," *IEEE Access*, vol. 11, pp. 87 316–87 329, 2023.

TABLE 19. Values of impairments for different PA impairments.

p%	parameter	Tx1	Tx2	Tx3	Tx4
5%	α_{AM}	2.051	2.267	2.051	2.267
	β_{AM}	1.209	1.209	1.094	1.094
	α_{PM}	3.803	4.203	3.803	4.203
	β_{PM}	9.559	9.559	8.649	8.649
2%	α_{AM}	2.116	2.202	2.116	2.202
	β_{AM}	1.175	1.175	1.129	1.129
	α_{PM}	3.923	4.083	3.923	4.083
	β_{PM}	9.286	9.286	8.922	8.922
1%	α_{AM}	2.137	2.180	2.137	2.180
	β_{AM}	1.163	1.163	1.140	1.140
	α_{PM}	3.963	4.043	3.963	4.043
	β_{PM}	9.195	9.195	9.013	9.013
0.5%	α_{AM}	2.148	2.169	2.148	2.169
	β_{AM}	1.157	1.157	1.146	1.146
	α_{PM}	3.983	4.023	3.983	4.023
	β_{PM}	9.150	9.150	9.058	9.058
0.3%	α_{AM}	2.152	2.165	2.152	2.165
	β_{AM}	1.155	1.155	1.148	1.148
	α_{PM}	3.991	4.015	3.991	4.015
	β_{PM}	9.131	9.131	9.077	9.077

TABLE 20. Values of impairments for different all impairments.

Parameter 5%	Tx1	Tx2	Tx3	Tx4	Tx5	Tx6
Imbalance g	1.425	1.455	1.485	1.515	1.545	1.575
Imbalance θ	0.000	0.017	0.035	0.052	0.070	0.087
CFO Δf	285	291	297	303	309	315
PN σ^2 10^{-7}	0.950	0.970	0.990	1.01	1.03	1.05
PN σ^2 10^{-4}	0.950	0.970	0.990	1.01	1.03	1.05
PA α_{AM}	2.051	2.094	2.137	2.180	2.223	2.267
PA β_{AM}	1.094	1.117	1.140	1.163	1.186	1.209
PA α_{PM}	3.803	3.883	3.963	4.043	4.123	4.203
PA β_{PM}	8.649	8.831	9.013	9.195	9.377	9.559

[6] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *ACM international conference on Mobile computing and networking (MobiCom)*, San Francisco, USA, 2008, p. 116.

[7] X. Guo, Z. Zhang, and J. Chang, "Survey of Mobile Device Authentication Methods Based on RF Fingerprint," in *IEEE Conference on Computer Communications (INFOCOM) Workshops*, Paris, France, 2019, pp. 1–6.

[8] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 MAC layer spoofing using received signal strength," in *IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, USA, 2008.

[9] S. Ur Rehman, K. Sowerby, and C. Coghill, "RF fingerprint extraction from the energy envelope of an instantaneous transient signal," in *2012 Australian Communications Theory Workshop (AusCTW)*. Wellington, New Zealand: IEEE, Jan. 2012, pp. 90–95. [Online]. Available: http://ieeexplore.ieee.org/document/6164912/

[10] S. U. Rehman, K. Sowerby, and C. Coghill, "Analysis of receiver front end on the performance of RF fingerprinting," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sydney, Australia, 2012, pp. 2494–2499.

[11] Y. Lin, J. Jia, S. Wang, B. Ge, and S. Mao, "Wireless Device Identification Based on Radio Frequency Fingerprint Features," in *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1–6.

[12] N. Soltani, K. Sankhe, J. Dy, S. Ioannidis, and K. Chowdhury, "More Is Better: Data Augmentation for Channel-Resilient RF Fingerprinting," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 66–72, 2020.

[13] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. Costa 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 *IEEE Conference on Computer Communications (INFOCOM)*, Toronto, Canada, 2020, pp. 646–655.

[14] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," in *IEEE Conference on Computer Communications (INFOCOM)*, Paris, France, 2019, pp. 370–378.

[15] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "*No Radio Left Behind:* Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 165–178, 2020.

[16] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Towards length-versatile and noise-robust radio frequency fingerprint identification," *IEEE Transactions on Information Forensics and Security*, 2023.

[17] C. Morin, L. Cardoso, J. Hoydis, J.-M. Gorce, and T. Vial, "Transmitter Classification With Supervised Deep Learning," *arXiv:1905.07923 [cs, eess]*, 2019.

[18] S. Hanna, S. Karunaratne, and D. Cabric, "WiSig: A Large-Scale WiFi Signal Dataset for Receiver and Channel Agnostic RF Fingerprinting," arXiv, article arXiv:2112.15363, 2022.

[19] Z. Zhang, A. Hu, W. Xu, J. Yu, and Y. Yang, "An artificial radio frequency fingerprint embedding scheme for device identification," *IEEE Communications Letters*, vol. 26, no. 5, pp. 974–978, 2022.

[20] A. Chillet, *RiFyFi*. [Online]. Available: https://github.com/JuliaTelecom/Rifyfi.jl

[21] J. Hall, M. Barbeau, and E. Kranakis, "Detection of Transient in Radio Frequency Fingerprint using Signal Phase," *Wireless and optical communications*, vol. 9, no. 13, pp. 1–6, 2003.

[22] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[23] 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 of Things Journal*, vol. 6, no. 1, pp. 349–360, Feb. 2019.

[24] 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.

[25] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Radio Frequency Fingerprint Identification for Security in Low-Cost IoT Devices," *arXiv:2111.14275 [eess]*, 2021.

[26] A. Elmaghbub and B. Hamdaoui, "LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability," *IEEE Access*, vol. 9, pp. 142 893–142 909, 2021.

[27] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. R. Chowdhury, "Trust in 5g open rans through machine learning: Rf fingerprinting on the powder pawr platform," in *IEEE Global Communications Conference (GLOBECOM)*, 2020, pp. 1–6.

[28] S. Hanna, S. Karunaratne, and D. Cabric, "Wisig: A large-scale wifi signal dataset for receiver and channel agnostic rf fingerprinting," *IEEE Access*, vol. 10, pp. 22 808–22 818, 2022.

[29] A. Jagannath and J. Jagannath, "Embedding-assisted attentional deep learning for real-world rf fingerprinting of bluetooth," *IEEE Transactions on Cognitive Communications and Networking*, 2023.

[30] A. Chillet, B. Boyer, R. Gerzaguet, K. Desnos, and M. Gautier, "Tangled program graph for radio-frequency fingerprint identification," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023.

[31] A. Elmaghbub, B. Hamdaoui, and W.-K. Wong, "Adl-id: Adversarial disentanglement learning for wireless device fingerprinting temporal domain adap," *arXiv preprint arXiv:2301.12360*, 2023.

[32] A. Elmaghbub and B. Hamdaoui, "Eps: Distinguishable iq data representation for domain-adaptation learning of device fingerprints," *arXiv preprint arXiv:2308.04467*, 2023.

[33] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio Frequency Fingerprint Identification for Narrowband Systems, Modelling and Classification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3974–3987, 2021.

[34] S. Hanna, S. Karunaratne, and D. Cabric, "Open set wireless transmitter authorization: Deep learning approaches and dataset considerations," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 59–72, 2020.

[35] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for rf device fingerprinting in cognitive communication networks," *IEEE journal of selected topics in signal processing*, vol. 12, no. 1, pp. 160–167, 2018.

[36] T. Jian, B. C. Rendon, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "MAC ID spoofing-resistant radio fingerprinting," in

IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2019, pp. 1–5.
[37] D. Roy, T. Mukherjee, M. Chatterjee, E. Blasch, and E. Pasiliao, "RFAL: Adversarial learning for RF transmitter identification and classification," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 783–801, 2019.
[38] G. Shen, J. Zhang, A. Marshall, and J. Cavallaro, "Towards Scalable and Channel-Robust Radio Frequency Fingerprint Identification for LoRa," *arXiv:2107.02867 [eess]*, Jul. 2021.
[39] J. Toonstra and W. Kinsner, "Transient analysis and genetic algorithms for classification," in *IEEE Communications, Power, and Computing Conference (WESCANEX)*, Winnipeg, Canada, 1995, pp. 432–437.
[40] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep Learning Based RF Fingerprint Identification Using Differential Constellation Trace Figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, Jan. 2020.
[41] K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, and Z. Huo, "Julia language in machine learning: Algorithms, applications, and open issues," *Computer Science Review*, vol. 37, p. 100254, 2020.
[42] C. Lavaud, R. Gerzaguet, M. Gautier, and O. Berder., "AbstractSDRs: Bring down the two-language barrier with Julia Language for efficient SDR prototyping," *IEEE Embedded Systems Letters*, pp. 1–1, 2021.
[43] M. Valkama, M. Renfors, and V. Koivunen, "Blind I/Q signal separation-based solutions for receiver signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 16, pp. 1–11, 2005.
[44] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and wiener phase noise," *IEEE Transactions on communications*, vol. 43, no. 2/3/4, pp. 191–193, 1995.
[45] R. Gerzaguet, L. Ros, F. Belvèze, and J.-M. Brossier, "On the performance of digital adaptive spur cancellation for multi-standard radio frequency transceivers," *Digital Signal Processing*, vol. 33, pp. 83–97, 2014.
[46] J. Pihlajasalo, D. Korpi, M. Honkala, J. M. Huttunen, T. Riihonen, J. Talvitie, A. Brihuega, M. A. Uusitalo, and M. Valkama, "Deep learning ofdm receivers for improved power efficiency and coverage," *IEEE Transactions on Wireless Communications*, 2023.
[47] M. Abdelaziz, L. Anttila, A. Brihuega, F. Tufvesson, and M. Valkama, "Digital predistortion for hybrid mimo transmitters," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 3, pp. 445–454, 2018.
[48] H. Fu, L. Peng, M. Liu, and A. Hu, "Deep learning based rf fingerprint identification with channel effects mitigation," *IEEE Open Journal of the Communications Society*, 2023.
[49] J. Huang, H. Chen, J. Ren, and F. Ye, "A novel joint estimation and compensation algorithm for non-idealities of analog front-end in DC-OFDM system," in *ASIC (ASICON), 2013 IEEE 10th international conference on*, Oct. 2013, pp. 1–4, iSSN: 2162-7541.

BIOGRAPHIES

Alice Chillet received the master's degree in Electronics, Electrical Energy, Automatics engineering from The University of Rennes, where she is currently pursuing a Ph.D. in telecommunications engineering and cybersecurity. In 2022, she worked with Tampere University on the Radio Frequency fingerprint model. Her research interest is Wireless device identification thanks to the Radio frequency fingerprint by using deep learning solutions.

Robin Gerzaguet received the degree in electrical engineering from the Grenoble Institute of Technology, France, in 2011, and the Ph.D. degree from the Gipsa-Laboratory, University of Grenoble and ST-Microelectronics, in 2015. From 2015 to 2017, he joined CEA-Leti, where he was involved in physical layer in the context of millimeter wave transmissions. He is currently associate professor at the French University of Rennes and in the IRISA laboratory. His research interests are embedded signal processing, including algorithms for physical layer, software-defined radio architectures, and the implementation of algorithms on embedded systems with strong energy constraints.

Karol Desnos is an associate professor at the National Institute of Applied Science (INSA) of Rennes. He holds a joint appointment at the Institute of Electronics and Digital Technologies (IETR) in the VAADER team. He obtained his PhD in Signal and Image Processing from the INSA Rennes in 2014. His research interests focus on dataflow models of computation and associated implementation techniques for the rapid prototyping of applications running on heterogeneous MPSoCs. Since 2019, he's leading the development of the open-source GEGELATI library, a lightweight framework for training and executing artificial intelligences based on tangled program graphs.

Matthieu Gautier holds the Engineering and M.Sc. degrees in electronics and signal processing engineering from the ENSEA graduate school and a Ph.D. degree from Grenoble-INP in 2006. From 2006 to 2011, he was a research engineer at Orange labs, INRIA and CEA-LETI laboratory. He is currently an Associate Professor at the University of Rennes and the IRISA Laboratory since 2011. His research activities are in the two complementary fields of embedded systems and digital communications for energy efficient communication systems.

Elena Simona Lohan (Senior Member, IEEE) the M.Sc. degree in electrical engineering from the Politehnica University of Bucharest, Bucharest, Romania, in 1997, the DEA degree (French equivalent of master) in economet- rics from École Polytechnique, Paris, France, in 1998, and the Ph.D. degree in telecommunica- tions from the Tampere University of Technology, Tampere, Finland, in 2003. She is a Professor with the Electrical Engineering Unit, Tampere University, Tampere. Her current research interests include global navigation satellite system (GNSS), low earth orbit positioning, navigation and timing (LEO-PNT), indoor location techniques, and privacy-aware positioning solutions.

Erwan Nogues received his engineering and Ph.D. degrees respectively from University of Rennes and INSA Rennes in 2000 and 2016. He held several positions as a signal processing expert in the semiconductors industry. He is now a technical expert in the field of system security and is an associate researcher at the Electronics and Computer Engineering department of INSA Rennes. His main research interests include signal processing, embedded systems and cyber-security.

Mikko Valkama[S'00, M'01, SM'15, F'22] received his M.Sc. (Tech.) and D.Sc. (Tech.) degrees (both with honors) from Tampere University of Technology, Finland, in 2000 and 2001, respectively. In 2003, he was with the Communications Systems and Signal Processing Institute at SDSU, San Diego, CA, as a visiting research fellow. Currently, he is a Full Professor and the Head of the Unit of Electrical Engineering at the newly formed Tampere University, Finland. His general research interests include radio communications, radio localization, and radio-based sensing, with particular emphasis on 5G and 6G mobile radio networks.