Sensitive devices Identification through learning of radio-frequency fingerprint

Alice Chillet

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Télécommunication*

Par
Alice CHILLET

Sensitive Devices Identification through Learning of Radio-Frequency Fingerprint

Thèse présentée et soutenue à Lannion , le « 21 Novembre 2024 »
Unité de recherche : IRISA UMR 6074

Rapporteurs avant soutenance :

Jean-Marie GORCE	Professeur des Universités, INSA Lyon
Laurent ROS	Professeur des Universités, Université Grenoble Alpes

Composition du Jury :

Présidente :	Karine AMIS	Professeur, IMT Atlantique
Examinateurs :	Fayçal AIT AOUDIA	Chercheur NVIDIA
	Jean-Marie GORCE	Professeur des Universités, INSA Lyon
	Erwan NOGUES	Chercheur, DGA
	Laurent ROS	Professeur des Universités, Université Grenoble Alpes
Dir. de thèse :	Matthieu GAUTIER	Professeur des Universités, Université de Rennes
Encadrants de thèse :	Karol DESNOS	Maitre de Conférence (HDR), INSA Rennes
	Robin GERZAGUET	Maitre de Conférence, Université de Rennes

Remerciements

Faire une thèse était un beau projet et comme dans tous les projets, il y a parfois des obstacles, des milliers de portes à ouvrir (ou pas), mais bien entouré on finit toujours par avancer, et j'ai eu la chance d'être très bien entouré. Alors avant de laisser place aux 200 pages de ce manuscrit, je tiens à remercier toutes les personnes qui ont permis de mettre de la couleur dans ces trois années.

Je tiens tout d'abord à remercier les membres de mon jury d'avoir accepté d'étudier mes travaux et pour nos échanges qui permettront de nourrir de futures discussions au sein de l'équipe. Parmi eux, je remercie particulièrement mes rapporteurs Jean-Marie Gorce et Laurent Ros pour leur relecture minutieuse de mon manuscrit. Membre du jury mais également référent tout au long de ces trois années de thèse, je remercie Erwan Nogues pour la grande richesse de nos discussions lors de nos rencontres à la DGA, discussions qui ont été vecteurs de motivations au cours de ces trois années.

Pour cette thèse j'ai eu la chance d'être accompagnée par une dream team, bienveillante, à l'écoute, ouverte et pédagogue. Un grand merci à mon directeur de thèse Matthieu Gautier pour la relation de confiance et de collègue que nous avons pu avoir dès le début qui m'a permis de tracer mon chemin dans ce sujet bien vaste avec l'assurance d'être dans une bonne direction. Je remercie également Karol Desnos de m'avoir fait profité de son expertise IA tout au long de cette thèse, pour sa patience lors des discussions modélisation/signal, et pour sa disponibilité malgré la distance et la rédaction de son HDR qui est une tâche particulièrement chronophage. Enfin un immense merci à Robin Gerzaguet pour sa gestion au quotidien de mes états d'âme et de mes problèmes techniques divers et variés, pour son écoute lors des moments de doutes, pour nos nombreuses discussions scientifiques, et nos grands questionnement sur le module de TNS. Et également un grand Merci de m'avoir fait confiance et accompagnée dans l'encadrement des TP et TD de ce module clairement pas si évident. À trois, vous avez su m'apporter toutes les clefs dont un doctorant a besoin pour comprendre et appréhender le monde de la recherche. Vous avez également créer un climat particulièrement propice à mon évolution dans ce monde: conseils, mise en relation, organisation de séminaires, conseils pour la vulgarisation scientifique et également la proposition d'une mobilité en Finlande. Cette expérience en

Finlande a été particulièrement enrichissante, je remercie Simona Elena Lohan et Mikko Valkama de m'avoir fait l'honneur de travailler ensemble sur des aspects de modélisation qui me semblaient bien obscurs, et pour leur accueil particulièrement enjoué. Je remercie également les doctorants avec qui j'ai pu échanger pendant mon séjour : Anna, Daria, Hans et Nikita.

Un grand merci à Emma pour nos discussions et nos grands questionnements du vendredi. Être deux face aux problèmes a été une grande source de réconfort. Merci à Paul pour la création des bases de données expérimentales à quelques semaines de la soumission du manuscrit. Et merci à Lucia pour son soutien et sa lumière. Merci à tous mes collègues doctorants et permanents des équipes GRANIT et TARAN pour les pauses améliorées (le pipeline galette), pour nos discussions, pour les repas du midi, et plus généralement la bonne ambiance.

Je pense aussi aux copains de la Fam qui, dispersés dans toute la France, ont su être source de soutien au cours de ces trois années, Kevin, Camille, Corentyn, Laetitia, Rémy et Alice (Coco et Lanka) pour leur accueil chaleureux chez eux, Mathieu et Hugo pour leur bonne humeur et leur engagement à souffrir sur une séance de sport. Léo pour son aide immense lors de mon emménagement et pour nos soirées avant son départ. Merci à Michel pour nos nombreuses discussions. Et enfin un immense merci à Alix pour son soutien indéfectible au quotidien malgré la distance qui nous éloigne, et immense n'est pas encore assez grand pour tout ce qu'elle a dû supporter de moi pendant ces trois ans, états d'âme et petit canard... Aux plus anciens, merci à Chloé, Marion et Oriane, et merci à Mathilde et Thomas pour nos soirées, nos petits goûters récap de vie et votre accueil à plusieurs reprises à Chaville. J'ai aussi une pensée pour toutes les belles personnes que j'ai rencontrées sur les patins pendant cette thèse.

Un très grand merci à mes parents, et à Karl pour leur présence, leur soutien, mais surtout pour les moments créatifs et sportifs qui m'ont permis tout au long de mes études de me déconnecter quand j'en avais besoin. Un grand merci à mes Mamies, pour qui mon sujet est très nébuleux mais ne fait, je crois, que renforcer leur fierté.

Enfin un immense merci à Fabien, même si encore une fois immense n'est pas suffisant, merci pour ton soutien, ta présence, et pour la bulle d'oxygène que tu as su créer pendant les périodes les plus difficiles...

Resumé

L'identification de dispositifs dits sensibles est soumise à différentes contraintes de sécurité ou de consommation d'énergie, ce qui rend les méthodes d'identification classiques peu adaptées. Pour répondre à ces contraintes, il est possible d'utiliser les défauts intrinsèques de la chaîne de transmission des dispositifs pour les identifier. Ces défauts altèrent le signal transmis et créent alors une signature par nature unique et non reproductible appelée empreinte Radio Fréquence (RF). Pour identifier un dispositif grâce à son empreinte RF, il est possible d'utiliser des méthodes d'estimation d'imperfections pour extraire une signature qui peut être utilisée par un classifieur, ou bien d'utiliser des méthodes d'apprentissage telles que les réseaux de neurones. Toutefois, la capacité d'un réseau de neurones à reconnaître les dispositifs dans un contexte particulier dépend fortement de la base de données d'entraînement. Dans cette thèse, nous proposons un générateur de bases de données virtuelles basé sur des modèles de transmission et d'imperfections RF. Ces bases de données virtuelles permettent de mieux comprendre les tenants et aboutissants de l'identification RF et de proposer des solutions pour rendre l'identification plus robuste. Dans un second temps, nous nous intéressons aux problématiques de complexité de la solution d'identification via deux axes. Le premier consiste à utiliser des graphes programmables intriqués, qui sont des modèles d'apprentissage par renforcement, basés sur des techniques d'évolution génétique moins complexes que les réseaux de neurones. Le second axe propose l'utilisation de l'élagage sur des réseaux de neurones de la littérature pour réduire la complexité de ces derniers.

Abstract

Identifying so-called sensitive devices is subject to various security or energy consumption constraints, making conventional identification methods unsuitable. To meet these constraints, it is possible to use intrinsic faults in the device's transmission chain to identify them. These faults alter the transmitted signal, creating an inherently unique and non-reproducible signature known as the Radio Frequency (RF) fingerprint. To identify a device using its RF fingerprint, it is possible to use imperfection estimation methods to extract a signature that can be used by a classifier, or to use learning methods such as neural networks. However, the ability of a neural network to recognize devices in a particular context is highly dependent on the training database. This thesis proposes a virtual database generator based on RF transmission and imperfection models. These virtual databases allow us to better understand the ins and outs of RF identification and to propose solutions to make identification more robust. Secondly, we are looking at the complexity of the identification solution in two ways. The first involves the use of intricate programmable graphs, which are reinforcement learning models based on genetic evolution techniques that are less complex than neural networks. The second is to use pruning on neural networks found in the literature to reduce their complexity.

Table of Contents

Acronyms

AWGN Additive White Gaussian Noise.

BER Bit Error Rate.

CFO Carrier Frequency Offset.

CIS Channel Independant Spectrogramm.

CNET National Center for Telecommunications Studies.

CNN Convolutional Neural Network.

COTS Commercial Orbital Transportation Services.

CP Cyclic Prefix.

CSI Channel State Information.

DAC Digital to Analog Converter.

DARPA Defense Advanced Research Projects Agency.

DCC Dilated Causal Convolution.

DCTF Differential Constellation Trace Figure.

DL Deep Learning.

DSP Digital Signal Processing.

DUT Device Under Test.

DWT Discrete Wavelet Transforms.

EMI ElectroMagnetic Interference.

ETU Extended Typical Urban.

EVA Extended Vehicular A.

FEC Forward Error Correction.

FFT Fast Fourier Transform.

FrFT Fractional Fourier Transform.

FT Fourier Transform.

GAN Generative Adversarial Network.

GPU Graphics Processing Unit.

GRU Gated Recurrent Unit.

IoT Internet of Things.

IQ In Phase - Quadrature.

KNN K Nearest Neighbor.

LDA Linear Discriminant Analysis.

LO Local Oscillator.

LRT Likelihood Ratio Test.

LSTM Long Short-Term Memory.

MAC Media Access Control.

MDA/MLE Multi discriminant analysis and Maximum Likelihood Estimation..

ML Machine Learning.

MLE Maximum Likelihood Estimation.

OFDM Orthogonal Frequency Division Multiplexing.

PA Power Amplifier.

PAM Pulse Amplitude Modulation.

PCA Principal Component Analysis.

PN Phase Noise.

PSD Power Spectral Density.

QAM Quadrature Amplitude Modulation.

RF Radio Fréquence.

RFF Radio Frequency Fingerprint.

RL Reinforcement Learning.

RNN Recurrent Neural Networks.

RSS Received Signal Strength.

SDR Software Defined Radio.

SEI Specific Emitter Identification.

SoA State of the Art.

SVM Support Vector Machine.

TPG Tangled Program Graph.

USRP Universal Software Radio Peripheral.

WT Wavelet Transformation.

List of Figures

List of Tables

List of Symbols

Hardware impairment model, Chapter 4

$x_{ant}(t)$	Signal emitted by the transmitter
$\mathcal{F}_{\text{RadioFrequencyFingerprint(RFF)}_{\text{Tx}}}$	Distorsion fonction of the transmitter
$\mathcal{F}_{\text{PA}}$	Distorsion fonction of PA
$\mathcal{F}_{\text{LO}}$	Distorsion fonction of LO
$\mathcal{F}_{\text{DAC}}$	Distorsion fonction of DAC
$\mathcal{F}_{\text{RFF}_{\text{Rx}}}$	Distorsion fonction of the receiver
Δ_ω, Δ_f	CFO impairment
g_I, g_Q	Gain IQ imbalance
θ	Phase IQ imbalance
Φ	Phase Noise impairment
$\underline{x}(t)$	Analytic signal
f_c	Carrier frequency
$x_{mix}(t)$	Output of Local oscillator
$x_{LO}(t)$	Local oscillator function with IQ imbalance
B	Standard Wiener process
c	Diffusion rate (LO quality)
σ_ξ^2	State Noise variance of PN
A	Amplitude non linearity of PA
ξ	Phase non linearity of PA
α_{AM}, β_{AM}, α_{PM}, β_{PM}	Parameters of PA Saleh model
$\angle x_{mix}$	Angle of x_{mix}
$x_{PA}(t)$	Signal after the PA Saleh model
$x_{PAM}(t)$	Signal after the PA memory model
$y(t)$	Signal after the propagagtion channel
h	Propagation channel
n	AWGN

Practical use, Chapter 4 Experimental parameter model, Chapter 5

N_{Tx}	Number of transmitters
N_{signals}	Number of signals per device
$ChunkSize$	Number of IQ samples in a signal

$p\%$	Similarity scenario
γ	Learning rate
dr	Dropout
N_{Tx}	Number of transmitters
$P^p_{Tx_k}$	Parameter value of transmitter k for $p\%$ similarity
P^p_{min}	Minimun value of parameter for $p\%$ similarity
P^p_{max}	Maximum value of parameter for $p\%$ similarity
δ_f	CFO Dispersion
ν_f	Frequency variation

Résumé étendu

La communication et l'identification sont deux composantes essentielles de la survie d'une espèce, qu'il s'agisse des êtres humains, des animaux ou encore des plantes. Toutes ont en effet développé des moyens de communication et d'identification pour partager des informations. L'être humain, par exemple, utilise naturellement la parole pour communiquer, et au fil des siècles il a développé de nombreuses techniques de communication, telles que le télégraphe, la communication par câble, puis la télécommunication, jusqu'à l'ère d'Internet et des objets connectés, aussi appelés Internet des Objets (IoT). Depuis une dizaine d'années, le déploiement de l'IoT ne cesse de croitre dans de très nombreux domaines tels que la santé, le sport, domotique, villes et batiments intelligentes, etc. L'introduction massive de cette technologie pose des défis en matière de sécurité pour assurer la transmission des données entre les dispositifs légitimes. Parmi ces défis, on peut notamment noter la question de l'identification de l'émetteur.

Dans le monde biologique, chaque individu a son identité biologique qui permet d'éviter que les indivius soient confondus. Pour simplifier l'identification, l'être humain utilise un numéro d'identification tels que numéro de sécurité sociale, ou d'identité numérique. Toutefois, dans certains contextes, une identification biologique est utilisée pour limiter les risques d'usurpation d'identité. L'empreinte digitale, par exemple, est une signature unique et biologique intrinsèque à l'être humain qui permet de nous reconnaître. De la même manière, les dispositifs communicants sans fil sont la plupart du temps différentiés grace à un identifiant, aussi appelé adresse MAC. Cependant, les traitements de codage et de chiffrement nécessaires à l'utilisation de cette clé d'identification peuvent s'avérer lourds en termes de calcul pour l'émetteur et ne permettent pas toujours d'éviter l'usurpation d'identité. C'est la raison pour laquelle des travaux proposent, à l'instar de l'empreinte digitale pour l'être humain, d'utiliser l'empreinte Radio Fréquence (RF) ou RFF des dispositifs.

L'empreinte RF d'un émetteur est une signature unique, créée par les composants matériels de la chaîne de transmission, qui apparaît dans les signaux transmis. La chaîne de transmission est composée d'un convertisseur numérique-analogique (CNA), d'un oscillateur local (LO) et d'un amplificateur de puissance (AP), comme détaillé sur la Figure 1.

Le CNA transforme le signal complexe dans le domaine analogique pour obtenir $x(t)$. L'oscillateur module le signal à la fréquence porteuse f_c, et l'AP amplifie le signal, créant $x_{ant}(t)$ pour la transmission via l'antenne.

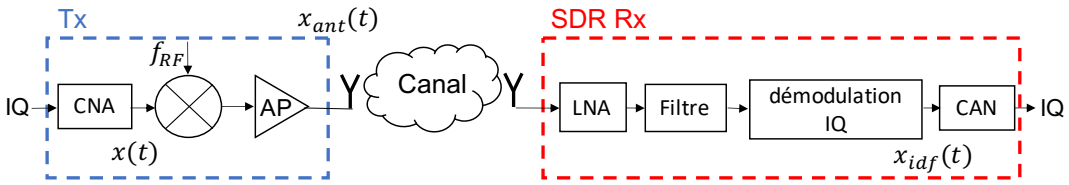


Figure 1 – Chaîne de transmission et de réception.

Tous ces composants déforment le signal et créent la signature appelée l'empreinte RF de l'émetteur notée $\mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}$. Le signal émis peut être modélisé par :

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}\left(x(t)\right), \tag{1}$$

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{AP}} \circ \mathcal{F}_{\mathrm{LO}} \circ \mathcal{F}_{\mathrm{CNA}}\left(x(t)\right), \tag{2}$$

où $\circ$ représente l'opérateur de composition de fonction qui exprime les traitements successifs de la chaîne de transmission, chaque étape créant une fonction de distorsion. $\mathcal{F}_*$ représente le comportement d'un composant, y compris sa distorsion. Les fonctions imbriquées de (2) montrent l'impact de chaque composant et la difficulté d'extraire des caractéristiques et de modéliser la transmission. Le LO ajoute des distorsions liées à la fréquence appelées Carrier Frequency Offset (CFO), au gain et à la phase appelés déséquilibre IQ, et du bruit de phase ou Phase Noise (PN). L'AP a un impact sur le gain et la phase, car il introduit une non-linéarité dans l'amplitude complexe du signal. La Table 1 présente les composants matériels et les dégradations correspondantes, les articles référencés dans ce tableau correspondent aux travaux qui utilisent et présentent les dégradations.

Dans la Figure 1, le bloc canal représente l'environnement de propagation sans fil défini par le bruit, les signaux d'interférence et les canaux à trajets multiples ainsi qu'à évanouissement qui pourraient avoir un impact sur le signal. Ce canal de propagation est modélisé par $\mathcal{F}_{\mathrm{canal}}$. Le bloc rouge Rx représente le récepteur avec ses composants (non détaillés dans le modèle mais similaires au modèle inversé Tx) et sa fonction de distorsion appelée RF du récepteur, $\mathcal{F}_{\mathrm{RFF}_{\mathrm{Rx}}}$. L'étude de l'impact du récepteur dépasse le cadre de cette thèse et nous ne considérons qu'un récepteur unique pour l'identification. De plus, il est important de noter que le récepteur peut être non légitime et ne pas disposer de

Composants	Imperfections	Références
Horloge	Clock jitter	[119]
CNA	Erreur d'échantillonnage	[112, 75]
Oscillateur local	Bruit de phase	[126]
	Offset de fréquence porteuse	[106]
	I/Q imbalance	[110, 107]
Amplificateur de puissance	Non linéarité	[75]

Table 1 – Relations entre les composants matériels et les dégradations utilisées pour l'identification des empreintes RF.

beaucoup d'informations sur l'émetteur et le canal de transmission, et donc manquer d'informations pour extraire correctement les dégradations RFF. Le signal reçu x_{idf} peut donc être exprimé comme suit :

$$x_{\text{idf}}(t) = \mathcal{F}_{\text{RFF}_{\text{Rx}}} \circ \mathcal{F}_{\text{canal}} \circ \mathcal{F}_{\text{RFF}_{\text{Tx}}} \left(x(t)\right). \tag{3}$$

L'environnement de propagation est défini par l'emplacement des appareils, la position relative des émetteurs et des récepteurs, le niveau de bruit, les signaux brouillés, etc. Toutes ces perturbations rendent difficile l'identification des empreintes RF. Ces problématiques sont largement étudiées dans le cadre de l'état de l'art et seront appelées impact des conditions du canal ou de l'environnement dans cette thèse.

Contexte d'application

Récemment, le nombre de contributions lié à l'identification par empreinte RF a connue une croissante importante [46, 115], et divers contextes d'application sont présentés. Le contexte d'application induit des hypothèses et des connaissances a priori différentes sur le(s) émetteur(s) et les signaux émis. Par conséquent, il est important de le prendre en compte.

Authentication pour renforcer la sécurité : il est possible d'utiliser les imperfections RF pour réduire le risque d'usurpation d'identité et améliorer les niveaux de sécurité dans les systèmes sans fil.

Authentication avec réduction de la consommation d'energie : il est possible de réduir la taille du paquet de transmission et donc de limiter le coût energétique d'une transmission. Elle est particulièrement intéressante pour les dispositifs IoT avec des

paquets courts pour lesquels la surcharge induite par l'authentification est importante. En outre, il s'agit d'une solution inviolable pour authentifier des dispositifs IoT à faible consommation d'énergie et/ou à faible capacité de calcul [47].

Identification pour la défense ou l'attaque : les cybercriminels peuvent utiliser des failles de sécurité et des systèmes logiciels pour masquer leurs activités, mais l'identification par empreinte RF peut permettre de les traquer en exploitant l'hypothèse qu'ils utilisent à un moment donné leur véritable identité.

Cette thèse ayant été financée par la DGA, le context applicatif qui nous intéresse est celui de la défense. Il pourrait être intérressant de pouvoir différencier des émetteurs alliers des émetteurs inconnus et donc potentiellement ennemis.

L'identification

Dans l'état de l'art, il existe deux familles de méthodes d'identification par empreinte RF : les méthodes paramétriques et les méthodes par apprentissage. Les méthodes paramétriques suivent deux étapes : premièrement les caractéristiques de l'empreinte sont extraites du signal par estimation ou changement de domaine. Puis un algorithme de classification est utilisé pour classer les signaux et estimer le dispositif émetteur, comme présenté sur la Figure 2.

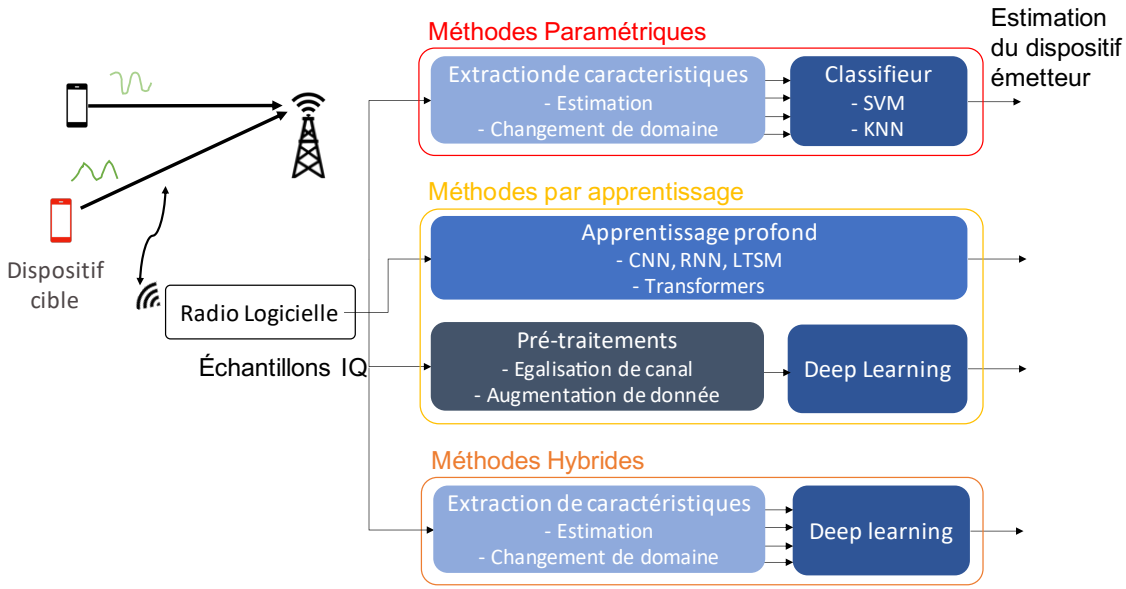


Figure 2 – Etat de l'art des solutions d'identification.

Les méthodes par apprentissage profond supervisé sont de plus en plus répandues et souvent réalisées par des Réseaux de Neurones qui prennent en entrée le signal brut ou pré-traité avec une égalisation de canal ou un changement de domaine. Les réseaux les plus couramment utilisés sont des Réseaux de Neurones Convolutifs ou Convolutional Neural Network (CNN). Par exemple, dans [92, 91], Sankhe et al. explorent différentes architectures de CNN avec plus ou moins de paramètres. D'autres types de réseaux ont également été explorés comme les réseaux récurrents [102], ou encore les *transformers* [96]. Dans nos travaux nous utilisons principalement un réseau de la littérature [91]. Pour évaluer la bonne classification d'un réseau, deux métriques sont utilisées, le Score F1 et la précision. Dans notre contexte le Score F1 est très proche de la précision car les bases de données sont toujours équilibrées, il est exprimé en pourcentage ou encore par une valeur comprise entre 0 et 1 (dans l'intervalle [0 ; 1], avec 1 la meilleure classification. Le réseau procède à la classificiation à partir de données qui lui sont fournies. Dans le cadre de l'identification par empreinte RF, ce sont directement les deux signaux temporels correspondant aux voies I et Q.

Pour faciliter l'apprentissage, les données complexes sont transmises au réseau par 2 voies indépendantes la voie I et la voie Q. Certains auteurs proposent de pré-traiter les données pour faciliter la reconnaissance de l'empreinte RF comme l'égalisation de canal qui permet d'estimer le canal de propagation afin de le compenser et de réduire son impact sur les données. En effet, la présence de ce canal déforme le signal et peut rendre l'empreinte invisible pour le réseau, ce qui rend son identification difficile, voire impossible. Une autre possibilité est de faire de l'augmentation de données, cette technique consiste à ajouter de la diversité dans la base de données tout en augmentant la quantité de données pour permettre au réseau de mieux se focaliser sur l'empreinte RF et non sur une caractéristique de la base de données. En classification d'image par exemple, l'augmentation de données consiste à flouter l'image, la retourner, ajouter du bruit etc. Dans notre contexte, ajouter du bruit ou changer l'environnement de propagation à plusieurs reprises lors de l'enregistrement des signaux peut-être une piste.

Bases de données et challenges

Si les Réseaux de Neurones obtiennent des résultats de classification prometteurs, leur entrainement nécessite une base de données qui soit importante et robuste pour permettre la résilience et la généralisation. En particulier l'état de l'art, ainsi qu'une étude menée

au cours de cette thèse, démontrent une chute des performances lorsque les conditions d'enregistrement des signaux changent entre l'entrainement et le test, ce qui montre que le réseau de neurone ne parvient pas à généraliser la classification. Les bases de données de l'état de l'art peuvent être divisées en plusieurs catégories. Dans un premier temps on trouve des bases de données privées et experimentales telles que celles utilisées pour les projets DARPA [91, 94]. Ensuite il existe des bases de données experimentales publiques comme Oracle [92] ou bien WiSig [40] qui sont deux bases de données intéressantes pour l'identification d'empreinte RF car chacune est composée de plusieurs enregistrements dans différents contextes. Enfin il existe des bases de données synthétiques crées grâce à des modèles de communications sans fils et des modèles d'empreinte RF. Ce type de bases de données proposé par [122], offre de la flexibilité pour explorer et comprendre l'identification par empreintes RF, toutefois cette base de données n'est pas publique. Les études menées sur les bases de données réelles montrent la difficulté d'interprétabilité des résultats. Pour améliorer cette interprétabilité, il serait intéressant de pouvoir créer facilement des bases de données pour explorer différents contextes et mises en situations. Toutefois créer une base de données n'est pas trivial et peut être chronophage, c'est pourquoi dans ces travaux nous proposons de surmonter les limites des bases de données réelles grâce à un générateur de bases de données virtuelles, ou synthétiques.

Le générateur de bases de données

Le générateur de bases de données virtuelle implémenté au cours de cette thèse est composé de différents blocs pour offrir de la flexibilité. Il est possible de définir le nombre de transmetteurs à simuler, la taille de la base de données, les données qui sont transmise et la modulation. En effet le type de trames peut avoir un impact sur la capacité à retrouver l'empreinte radio fréquence tout comme la modulation peut avoir un impact sur la signature RF. Les modèles d'empreinte RF sont paramétrables et permettent donc de définir une multitude d'empreintes en changeant les paramètres des modèles. Enfin il est possible de simuler la propagation de l'onde RF dans un canal de propagation. La Figure 3 présente les différents blocs du générateur de bases données. Le générateur est utilisé pour tester la capacité d'un réseau de neurones à classifier les données lorsque le scenario d'acquisition des données est différent de celui qui a été utilisé pour l'acquisition des données d'entrainement. La résilience du réseau peut donc être facilement testée à différent niveau suivant les scénarios. Le réseau utilisé pour ces travaux est un CNN

de l'état de l'art composé de deux couches convolutives et de trois couches entièrement connecté.

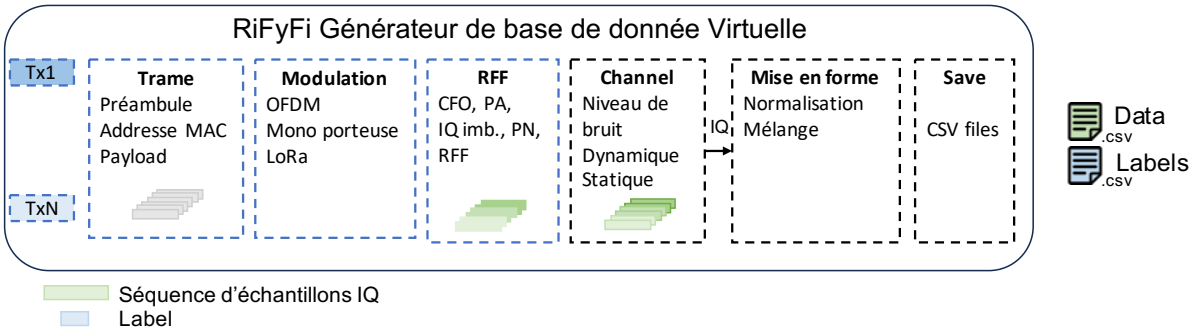


Figure 3 – Générateur de base de données virtuelle.

Nous étudions différents scénarios et particulièrement l'impact du type de trames transmises par les émetteurs sur notre capacité à identifier les emetteurs, en fonction de la similarité des empreintes RF. La similarité entre les émetteurs est définie par un intervalle exprimé en pourcentage, plus la valeur est faible plus la similarité entre les empreintes des émetteurs augmente. Une première étude est menée sur l'impact individuel des imperfections en fonction de leur similarité d'un transmetteur à l'autre sur la capacité du réseau à classifier les signaux dans un contexte préambule idéal. Dans ce mode, tous les transmetteurs émettent et répètent la même séquence. Les résultats révèlent l'importance de l'AP pour séparer les transmetteurs. Ensuite une étude de la combinaison des imperfections est proposée, avec un interet particulier pour l'impact des similarités des empreintes sur la capactité de classification. Cette étude présente et évalue les leviers à notre disposition pour améliorer l'apprentissage. Nos résultats présentent une spécialisation du réseau de neurones aux données d'entrainement lorsque les transmetteurs présentent une forte similarité d'empreinte RF, ce phénomène est peut-être enrayé en fournissant plus de données au réseau lors de l'entrainement, comme présenté Figure 4.

Une étude poussée sur l'impact du CFO a été menée car l'offset de fréquence est impacté par la température ambiante et donc de fortes variations de CFO peuvent apparaitre sur un même transmetteur à différents instants.

Enfin, nous proposons de faire l'apprentissage avec des données de type payload. Dans ce mode, les transmetteurs émettent tous des séquences de symboles aléatoires, différentes d'un transmetteur à l'autre et d'une émission à l'autre, avec une très forte diversité due à la génération de séquences aléatoires pour chaque transmetteur. Ainsi nous montrons

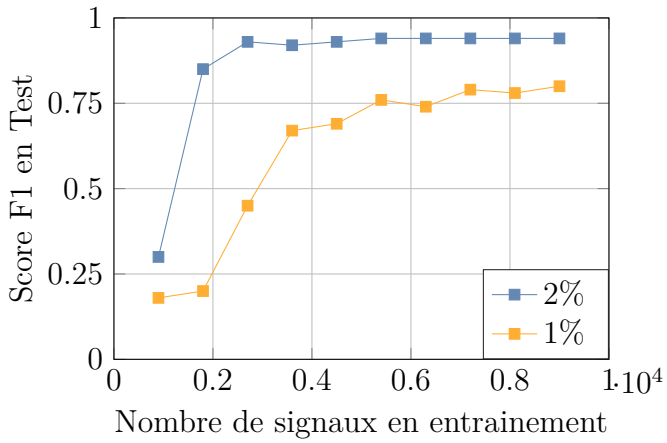


Figure 4 – Score F1 obtenu en test en fonction du nombre de signaux utilisés pour entraîner le réseau lorsque l'entraînement a atteint 98% du score F1.

qu'il faut utiliser un plus grand nombre de données lors de l'entrainement pour permettre d'obtenir de bon résultats. Cette méthode d'apprentissage permet d'obtenir de bons résultats en test, même dans des conditions bruitées par un canal de propagation. Ce qui permet de proposer une solution d'augmentation de données moins complexe à mettre en place que les solutions de l'état de l'art basé sur la diverstié de canaux de propagation.

Des données virtuelles aux données réelles

L'utilisation de données basée sur des modèles offre une grande flexibilité d'exploration, mais il est intérressant de confronter les résultats obtenus à ceux que l'on peut obtenir grâce à une base de données constituée de signaux expérimentaux. Nous avons donc proposé différents scénarios allant d'un cas de communication idéal réalisé par un cable jusqu'à des conditions réelles avec un éloignement des transmetteurs de quelques mètres. Pour chaque scénario, nous proposons deux modes, le mode Préambule et le mode Payload tous deux présentés précédemment. Les résultats permettent de confirmer les conclusions établies avec le générateur de bases de données. Les bases de données expérimentales créées permettent d'évaluer le degré de résilience d'un réseau de neurones en fonction des différents scénarios.

Apprentissage automatique léger pour l'identification d'empreinte RF

Les réseaux de neurones présentent une bonne capacité de classification et des résultats prometteurs, toutefois la complexité de ces systèmes en apprentissage et en inférence dépend fortement de leur architecture mais restent fortement complexes. Dans le contexte de l'loT, l'identification par RFF peut être contrainte en termes de complexité, d'architecture matérielle et de consommation d'énergie. Nous proposons d'adresser cette problématique suivant deux axes, le premier étant l'utilisation des graphes programmables intriqués pour l'identification et le second axe concerne l'élagage des réseaux de neurones.

Introduit en 2017 par Kelly et al. [51], les Tangled Program Graph (TPG) sont des modèles d'apprentissage par renforcement construits grâce à des techniques de programmation génétique. Contrairement aux réseaux de neurones dont la topologie est choisie par un expert en science de la donnée, les TPG sont construits au fil des évolutions génétiques. Par conséquent leur topologie et leur complexité s'adaptent automatiquement à la complexité de la tâche à apprendre. Les TPG ont prouvé leur compétitivité face aux réseaux de neurones de l'état de l'art, permettant une réduction de la complexité et de la quantité de mémoire nécessaire, en entraînement et en inférence [50].

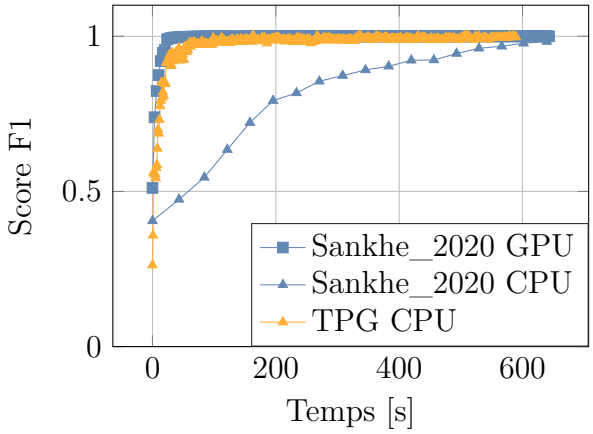


Figure 5 – Evolution temporelle du score F1 des différents réseaux sur différents matériels, pour une taille de lot de 200.

Pour évaluer les performances et valider l'intérêt d'utiliser les TPG, nous comparons la vitesse de convergence de la phase d'entraînement d'un TPG et d'un réseau de la littérature. La Figure 5 présente le score F1 obtenu pour chaque réseau en fonction du temps. La performance du TPG sur CPU est représentée avec les triangles jaunes et celles

du CNN sur CPU et GPU avec respectivement les triangles bleus et les carrés bleus. En ce qui concerne les résultats obtenus sur CPU, le TPG présente une accélération importante par rapport au CNN. Sa vitesse est en effet très proche d'un entraînement du CNN sur un GPU avec un avantage pour les systèmes embarqués : l'apprentissage peut se faire sur une plateforme sans accélération GPU spécifique avec une vitesse similaire.

L'élagage (ou *pruning*) part du constat qu'un réseau contient naturellement trop de paramètres et possède de nombreuses redondances, provoquant un gaspillage d'espace et de ressources de calculs. L'élagage d'un réseau tire son nom de la botanique, et consiste retirer certaines parties du réseau d'apprentissage, qu'il s'agisse de neurones ou voire même de couche complète (filtre), de façon à rendre le réseau plus léger et rapide. Il existe différentes méthodes d'élagage basées sur l'élagage des filtres ou des neurones entiers, ou plus fine, en retirant certaines parties des filtres ou certains poids des neurones. Une étude est menée sur différent types d'élagage et on montre qu'il est possible de diviser la taille du réseau par deux ou plus en fonction de sa taille initiale sans perdre de performances. La Figure 6 présente le score F1 en fonction du niveau d'élagage obtenu pour un réseau de la littérature sur deux bases de données. Le niveau d'élagage est définit suivant la *sparsity* de 0 à 1, avec 0 le réseau non élagué.

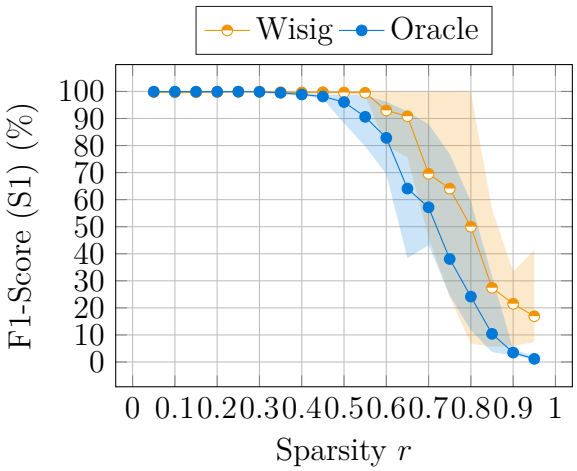


Figure 6 – Score F1 en fonction du niveau de l'élagage du réseaux de neurones.

Conclusions et perspectives

Ces travaux mettent en évidence la difficulté d'assurer l'identification par reconnaissance des empreintes RF et non une reconnaissance d'un biais de la base de donnée tel

que la position relative du dispositif. Les résultats de cette thèse offrent donc de nombreuses perspectives de travaux. L'outil développé au cours de la thèse peut permettre d'envisager le transfert learning ou encore de modéliser un véritable jumeaux numériques de radio logicielle connue telles que celles utilisées pour nos experimentations. A long terme, étudier et proposer des méthodes de prétraitement des données pourrait permettre l'identitifcation par empreinte RF dans un contexte non controlé. Enfin, l'identification par empreinte RF dans un contexte basse consommation semble intéressant à explorer avec une étude de consomation d'énergie des solutions d'identification.

Le reste du manuscrit se décline sous six chapitres.

— Le chapitre 1 introduit et définit l'identification par empreinte RF, les contextes d'application et les challenges qui en découlent.

— Le chapitre 2 introduit les principes de communication numériques et définit les empreintes RF puis présente l'état de l'art des solutions d'identification par empreinte RF. Les méthodes paramétriques et les méthodes par apprentissage profond sont présentées dans ce chapitre.

— Le chapitre 3 présente un état de l'art des bases de données utilisées pour l'identification par empreinte RF dans le contexte académique. Cet état de l'art permet de définir les challenge liés aux bases de données en particulier lorsqu'on utilise des réseaux de neurones, et une première étude est menée sur deux bases de données et trois réseaux de neurones de la littérature.

— Le chapitre 4 décrit le générateur de bases de données développé pendant la thèse. Cet outil permet de créer des bases de données virtuelles incluant des modèles d'imperfections RF. Ce générateur est un outil en libre accès [11] et disponible pour la communauté.

— Le chapitre 5 présente les résultats obtenus grâce au générateur de bases de données virtuelles qui ont été présentés en conférence [12] avec une étude de l'impact du canal de propagation sur l'identification. Ce chapitre présente également d'autres travaux basés sur le générateur publiés dans un journal [14] abordant l'impact du mode de transmission, du CFO et de la similarité des imperfections.

— Le chapitre 6 présente une étude menée au laboratoire avec des radios logicielles pour confronter les résultats et conclusions tirées de l'utilisation du générateur de bases de donnée virtuelles et des signaux réels non simulés.

— Le chapitre 7 introduit l'intérêt des réseaux légers pour les applications d'identification d'empreinte RF et présente deux solutions d'apprentissage permettant de réduire la complexité de l'identification.

— Enfin le chapitre 8 presente une conclusion et les perspectives de cette thèse.

Chapter 1

Introduction

1.1 History of communication

The world we know today would only exist with communication. From time immemorial, living species - trees, animals, fungi, and humans - have developed communication techniques to interact with other species members. These communications are based on the generation of an acoustic (voice), mechanical, electrical, or electromagnetic wave and the transmission of this wave via a propagation channel. For example, animals, such as humans, communicate using sound waves with a more or less complex communication language.

Most natural communications are short-distance, for example, the sound allows us to communicate with someone in the same area. In 1794, with the first optical telegraph, researchers wanted to make communication between humans at long distances possible. Since then, they always tried to improve these communications as it is shown in Figure 1.1. In 1832, the Morse language was created to improve communication and after several years the phone became a particular object of houses. The communication distance increases gradually with the communication between countries and then continents.

During the Second World War, the laboratories of the belligerents perfected new applications to offer new technical opportunities for the war such as radar and walkie-talkie. After the war, the first American satellite was launched in 1958, in 1962 the Telstar was launched and involved sending images and sounds between countries via space satellites. In France, Pierre Marzin, the director of the National Center for Telecommunications Studies (CNET), convinced the administration to install the French reception station in Pleumeur-Bodou. In 11 July 1962 at 0.47 am, in front of an audience of 190 technicians and 150 journalists, the first televised images of the United States via the Telstar 1 satellite were captured by the antenna located in Pleumeur-Bodou.

In 1973, Martin Cooper placed the first cellular mobile call to his rival at Bell Labs. The first mobile phone had a maximum talk time of 30 minutes, and it took a year for

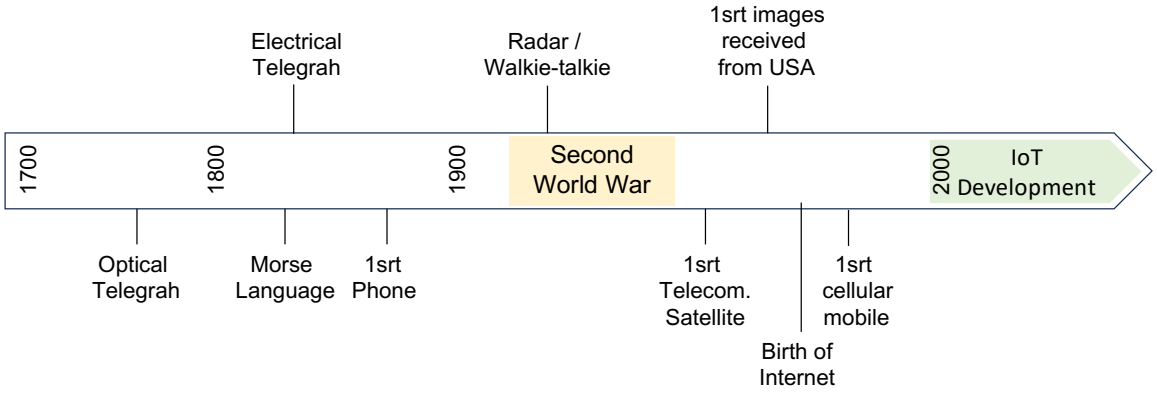
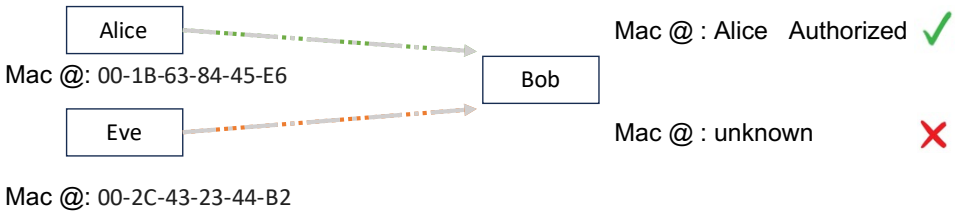


Figure 1.1 – Telecommunication timeline.

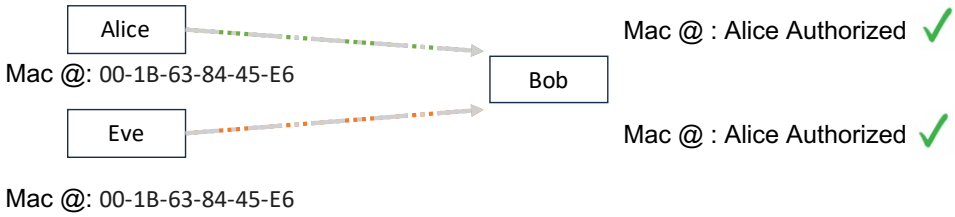
the battery to recharge. The phone would eventually be a prototype for Motorola's first mobile phone. In 1966, the concept of the Internet was created with a real development across several years and was born officially in 1983, gradually internet replaced other communication methods until it became the most widely used means of communication. The advent of sensor miniaturization changed the paradigm and developed a need to share and exchange data. At the same time, the technical progress allowed network creation with several devices that exchange information, and the concept of the Internet of Things (IoT) was created. Since a decade ago, the IoT has been introduced in health, for pain control applications, in sports for performance evaluation, in quotidian life, and in the industry for domotic flat to control lighting and security. This massive introduction of the IoT in many applications, challenges the researcher and engineer to propose low-consumption systems to improve the embedded characteristics. Moreover, security is also a challenge to ensure a correct transmission between the legitimate transmitter and the legitimate receiver. Improving the security challenge can be separated into two axes: encryption of data and authentication of the transceiver. In this PhD, we focus on the emitter identification.

1.2 Need of secure identification

Each biological species has its own identity and cannot be identified as another one. This identity is biological but to simplify communication we decided in our society to allocate an identification (ID) number to people to identify them. In most cases, the ID number is sufficient for identification but in other critical situations, it is not sufficient. To ensure security and avoid spoofing identification by using the identity of another person,



(a) Identification thanks to MAC address.



(b) Spoofing identification with MAC address.

Figure 1.2 – Identification and spoofing presentation.

the biological identity is used. For example, the fingerprints, the DNA (deoxyribonucleic acid), or the background of the eye are a biological signatures which can be used to identify people without any doubt. In the same manner, the development of electronic devices that can communicate with each other requires identification protocols. This identification is mainly based on the meta-data of the communication protocol that gives an address or a registration number to enable the authentication as shown in Figure 1.2a, with the Media Access Control (MAC) address identification protocol. However, because this device can be a victim of spoofing as shown in Figure 1.2b, the identification may require more secure identification.

In Figure 1.2b, Bob recognizes Alice thanks to its MAC address but Eve can pretend to be Alice if she knows Alice's MAC address, and sends the wrong information to Bob, this is called spoofing. Much research has been done to improve the encryption of MAC addresses to make it more robust to attack but the algorithms are heavy and not always adapted for IoT. The IoT constraint limits the development of the security process, encryption for example, for cost reasons or just to reduce energy consumption. Therefore, IoT devices become an easy target for attackers to get access to the IoT network, which can be included in a bigger private network without a real security system.

That is why, to ensure identification, the State of the Art (SoA) proposes to use the electronic identity of the device as the biological identity for humans. The analogy between humans and electronic devices is illustrated in Figure 1.3. This electronic identity

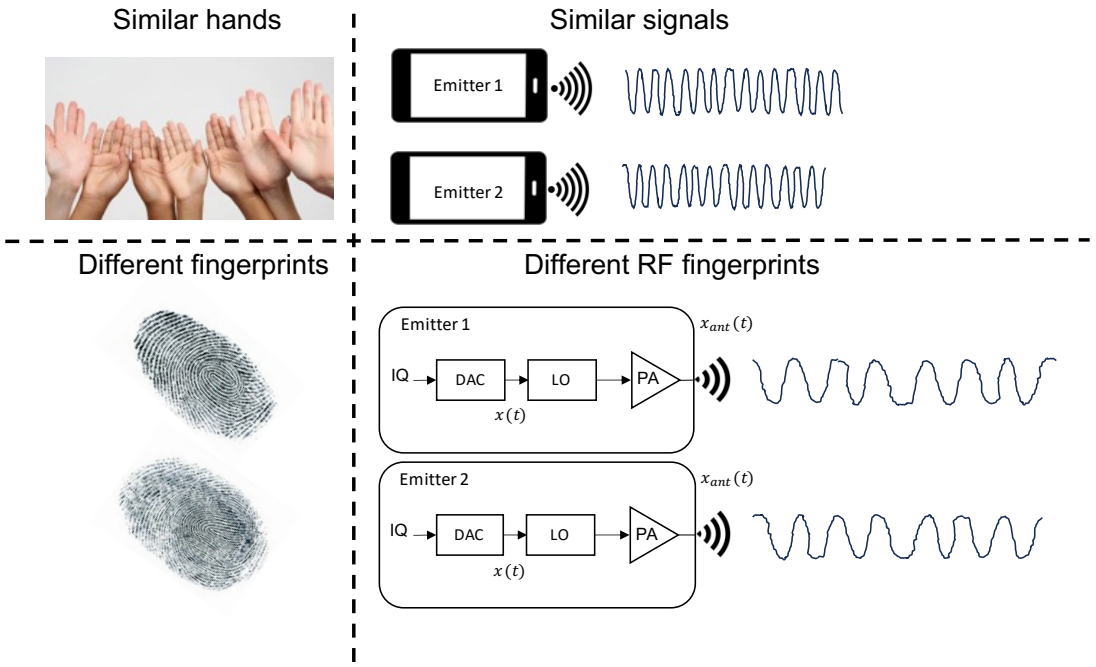


Figure 1.3 – Analogy between transmitter impairments and human biological characteristics.

is the aggregation of device imperfections which create some distortions in the transmitted signals. These distortions are called: Radio Frequency Fingerprint (RFF), specific emitter identity, or, physical layer identity.

The RFF identification is a sub-family of Physical-Layer authentication in wireless communications [115], called sometimes Specific Emitter Identification (SEI). This identification is and must be independent of the location of wireless users, as opposed to the methods based on channel properties [71, 99, 114], that require a strong assumption on users' stationarity [77]. The RFF identification can be used lonely or combined with the MAC address or key to improve the authentication security, depending on the application context.

The intrinsic definition of RFF makes it hard to replicate and allows a secure authentication. Moreover, it is possible to reduce the communication protocol without sending an address. Hence, a trade-off has to be found between the security of data and the energy consumption to ensure robust and lightweight authentication systems [3]. The signal is transmitted over a wireless propagation channel, which filters the signal and adds noise that can affect the RFF and make identification difficult.

1.3 Deep Learning RFF identification and challenges

The RFF identification can be considered as a signal classification. For a decade, the Deep Learning (DL) techniques have been massively introduced for image classification and obtained very good performance for recognition and classification issues. Since 2018, because of the complexity of RFF identification caused by the different distortions of the different components, the SoA has switched to parametric-based classification in favor of DL classification. In particular, the classification is done thanks to a neural network, and most of the time a Convolutional Neural Network (CNN).

However, the DL identification challenges the community on different aspects of the training. Firstly, while DL techniques present promising results, there is a strong need for a large and robust database [46]. Secondly, the SoA results show a performance penalty when the signal acquisition conditions change. The signal acquisition conditions are defined by the day of capture, the position of the both transmitter and the receiver, and the environmental parameters such as temperature and electromagnetic environment. Finally, the complexity of the DL-based SoA techniques makes it difficult to have an embedded identification system.

In this PhD, we first focus on the database issues, and then we propose to interest us to the complexity of the classification solutions. The behavior of the network training with SoA database shows a disturbance caused by environmental changes and makes robust identification impossible. Moreover, Jagannath et al. propose to identify static devices by their location [46], which signifies that the distortion of the propagation channel can be a solution to recognizing the device by the propagation channel characteristics between transmitter and receiver. However, location methods have strong limitations because they are intrinsically sensitive to environmental variations and identification accuracy falls in a dynamic context. A secure identification solution should be robust to time and environment changes, especially in a wireless context.

Most existing RFF identification works use experimental data to explore RFF identification [40, 94, 92]. A study is proposed in Chapter 3 on two different databases to understand the behavior of the network. This study shows that the number of data/signals in the current databases is not enough. In particular, the diversity of data and the metadata about database creation is not pertinent to understand correctly the network behavior. We discuss the limits induced by the lack of information of experimental protocol, and the difficulty to create a dataset without bias to ensure the RFF identification.

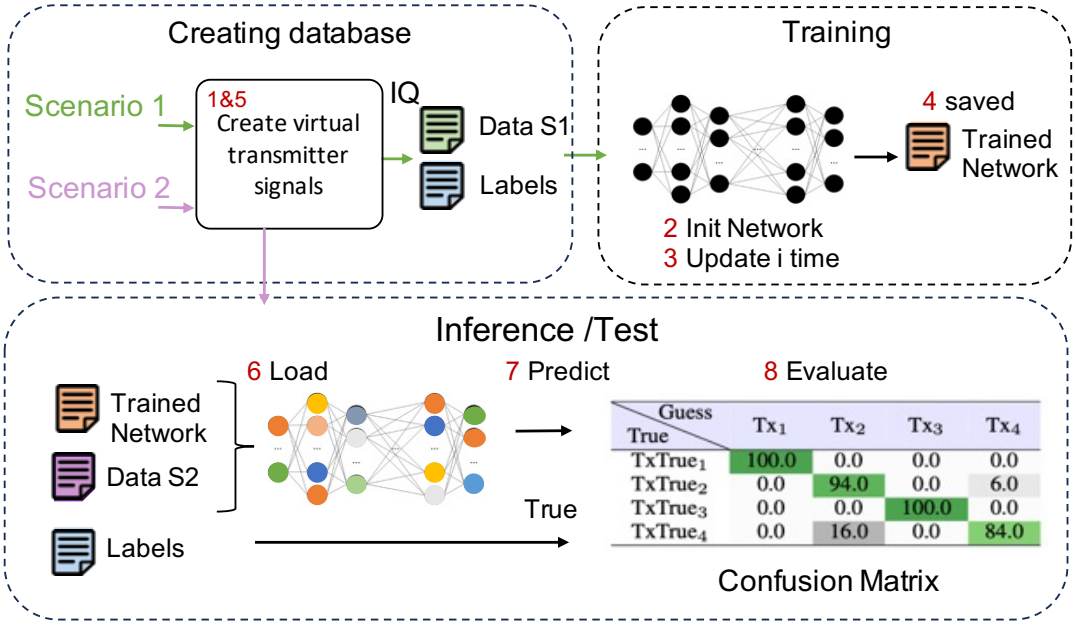


Figure 1.4 – A description of what the virtual database generator is designed to do.

1.4 RFF Virtual Database Generator

To overcome the limits of databases, we propose a virtual database generator as a first contribution. This generator allows to create a virtual database based on wireless transmission models, RFF models and wireless propagation channel models. A large panel of parameters can be chosen to create different scenarios such as the type of data to be emitted, the wireless protocol, the number of signals, and the presence of noise for example. This flexible generator, presented in Chapter 4, allows us to explore several database issues and propose some solutions validated thanks to the virtual databases. The principle objective is presented in Figure 1.4 and is called RiFyFi. First, a database is created following a scenario (S1), then the network is initialized and trained. Then, a second database is created with the same virtual transmitters according to another scenario (S2), the trained network is loaded and predicts the class of the signals of the second database. Finally, the capacity of the network to correctly classify the transmitters is evaluated. A scenario is defined by multiple meta-parameters such as the number of signals and the type of the transmitted sequence which can be a preamble sequence that corresponds to the part of a communication header or it can be payload sequences that are always different.

In Chapter 5, the results of the different studies are presented and some design rules of experimental databases are drawn for different scenarios. We found out that the power amplifier imperfections play the biggest role in RFF accuracy. We evaluate the number of signals required to train the network depending on different scenarios in particular in preamble and payload scenarios with and without propagation channels. The flexibility of the generator allows us to analyze the impact of the unstable known behavior of the frequency, by changing the parameter values of the model.

After this study, and the conclusion obtained we propose to create different experimental datasets to test and determine if the same conclusions can be made with the experimental data and confront the synthetic world and real world. The digital twins are a great tool to explore and understand the classification mechanism. However, the objective is to improve our understanding of real data so creating a dataset thanks to the conclusions done with the digital twins and observing the network training behavior allows us to conclude on the interest of this generator.

1.5 Lightweight Machine Learning for RFF identification

Throughout this PhD, a particular interest in lightweight classification has been developed with different research. Firstly, the recent interest in the Tangled Program Graph (TPG), mainly applied to image classification, makes it interesting in our context. The TPG is a reinforcement learning model based on genetic programming techniques. The main advantage of TPG is the adaptability of the complexity. We show that the convergence speed of the TPG in CPU is close to the SoA neural network on a Graphics Processing Unit (GPU) and obtained close results in the test in a favorable scenario compared to the neural network of the SoA. Finally, we propose to use pruning techniques to reduce the size of the neural network. Different architectures of networks are pruned and compared in terms of complexity and classification performance. The performance is evaluated thanks to real databases studied in Chapter 3 and the generalization capability is evaluated thanks to a second dataset scenario.

1.6 Contributions

International Journal

— A. Chillet, R. Gerzaguet, K. Desnos, M. Gautier, E. Lohan, E. Nogues, and M. Valkama "Understanding Radio Frequency Fingerprint Identification with RiFyFi Virtual Databases," in IEEE Open Journal of Communication Society vol. 5, pp. 3735-3752, 2024.

International Conferences

— (submitted) A. Chillet, R. Gerzaguet, K. Desnos, P. Bazerque, E. Nogues, and M. Gautier "Data Diversity for a Channel-Resilient Training Database for Radio Frequency Fingerprint Identification", in IEEE International Conference on Communications (ICC), 2025.

— E. Bothereau, A. Chillet, R. Gerzaguet, M. Gautier, and O. Berder "Investigating Sparse Neural Networks for Radio Frequency Fingerprint Identification", in IEEE Vehicular Technology Conference (VTC), 2024.

— A. Chillet, R. Gerzaguet, K. Desnos, M. Gautier, E. Lohan, E. Nogues, and M. Valkama "How to Design Channel-Resilient Database for Radio Frequency Fingerprint Identification?", in IEEE International Conference on Communications (ICC), 2024.

— 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.

National Conferences

— A. Chillet, B. Boyer, R. Gerzaguet, K. Desnos et M. Gautier, "Graphes programmables intriqués pour l'identification d'empreintes Radio-Fréquence", in GRETSI 2023–29ème colloque du Groupement de Recherche en Traitement du Signal et des Images.

Mobility

I stayed in Finland for 3 months in 2023 between February and June. The work presented in the Open Journal of Communication Society and the ICC conference has been done in collaboration with Professors from Tampere University Pr. Mikko Valkama and Pr. Elena Simona Lohan.

1.7 Thesis Organization

This thesis is structured as follows

— Chapter 2 presents some fundamentals of digital communications and RF impairments definitions, and the RFF identification solutions with parametric-based methods and DL-based methods.

— Chapter 3 presents the SoA of databases for RFF identification and the challenges linked to the databases. A preliminary study is then presented to evaluate the limits of two public databases which seem interesting.

— Chapter 4 introduces the proposed Virtual Database Generator, called RiFyFi_VDG, with RFF impairments models which offers scenario flexibility description.

— Chapter 5 presents the results of the exploration of both the parameters and the scenarios obtained thanks to RiFyFi_VDG. First, an independent study is proposed to analyze the individual impact of each impairment. Then we explore different situations in a preamble transmission mode and secondly in a payload mode.

— Chapter 6 proposes experimental datasets creation to validate the lessons learned from the virtual database generator, in different modes and scenarios.

— Chapter 7 introduces two lightweight identification solutions, the TPG and the DL pruning techniques. The two solutions are independently studied and confront to generalization issues to evaluate the performance of the identification solutions.

— Finally Chapter Conclusions and Perspectives presents the main points of this PhD and the future works that can be considered. The perspectives are presented following the two main contributions: the first one concerns the use of the RiFyFi generator for transfer learning and the second point focuses on lightweight opportunities and solutions.

Chapter 2

RFF Identification State of the Art

In recent years, the RFF identification has been largely studied [46] to improve authentication security. The purpose of RFF is to uniquely identify a device by recognizing flaws in the emitted signal. These flaws are created by hardware impairments of the transmitter. The impairments create unique electromagnetic distortions in the transmitted signal [8], and these distortions are used to differentiate devices. While, in most telecommunication standards, identification methods are based on the meta-data of the communication protocols such as a MAC address, the RFF identification can be combined with such classic identification to improve and ensure identification without spoofing [34]. While RFF identification principle was born in the 2000s, using parametric models, the number of RFF identification by classification methods has recently exploded with the advent of DL [94, 34]. 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.

This chapter presents the SoA of RFF identification methods. Section 2.1 presents the ideal (i.e. without any impairments) communication chain between an emitter and a receiver. Section 2.2 describes and defines RF transmission and the different impairments, Section 2.3 presents different applications contexts. Section 2.4 describes the identification system and Section 2.5 presents the parametric methods used for identification based on particular feature extraction which is characteristic of RFF. Section 2.6 presents the DL methods that take in input the signal in time domain. Finally, Section 2.7 presents the recent methods that combined feature extraction and DL techniques, and Section 2.8 concludes this chapter.

2.1 Introduction to digital communication principles

This section presents the communication model between an emitter and a receiver. To ensure the communication, the useful data are concatenated in a packet, as detailed in

Figure 2.1 – Transmission packet.

Figure 2.1 where the preamble (Pre) part corresponds to the synchronization information, then the MAC address (MAC) of the transmitter and the receiver, and the useful data, payload (Pay). This packet is then processed as a binary stream following the description in this section. First, the ideal baseband model is presented, then this model is updated to introduce the carrier frequency model, with single and multi-carrier waveforms.

2.1.1 Baseband transmission

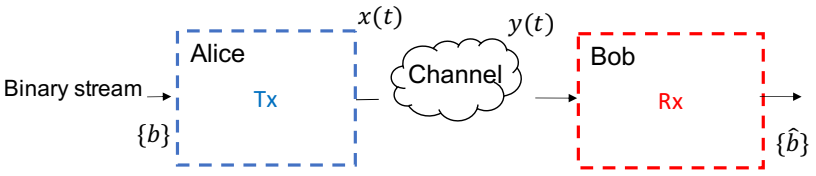


Figure 2.2 – Transmission and reception chains.

To illustrate the transmission of information, we suppose that Alice and Bob are respectively the sender and the receiver and that Alice wishes to send a message, a binary stream, to Bob via a propagation channel, this situation is presented in Figure 2.2. The transmission of this information involves many disciplines: source coding, channel coding, and, more generally, what is known as information theory, which makes it possible to guarantee the reliability of the transmission as a function of the propagation channel. Information theory and binary information protection are beyond the scope of this work, however, the rest of the transmission chain is the main point of this PhD. To carry out the transmission, the transmitter must convert the binary stream into a signal, which is transmitted through the propagation channel which can modify the signal. Then the receiver decodes the signal to obtain a binary stream. The performance of this communication chain is evaluated thanks to the Bit Error Rate (BER). The BER is a fundamental metric used to quantify the accuracy of digital communication systems. It represents the ratio of bits received in error to the total number of bits transmitted. In other words, BER measures the probability that a bit transmitted over a communication channel is received

incorrectly due to various factors such as noise, interference, distortion, or channel properties. Considering Additive White Gaussian Noise (AWGN) $b(t)$ propagation channel, the received signal $y(t)$ can be expressed as $y(t) = x(t) + b(t)$. Assuming a digital sampling with sampling time T, the equation can also be represented as a discrete sequence $y[n] = x[n] + b[n]$ with $x[n] = x(nT)$.

A. Transmitter

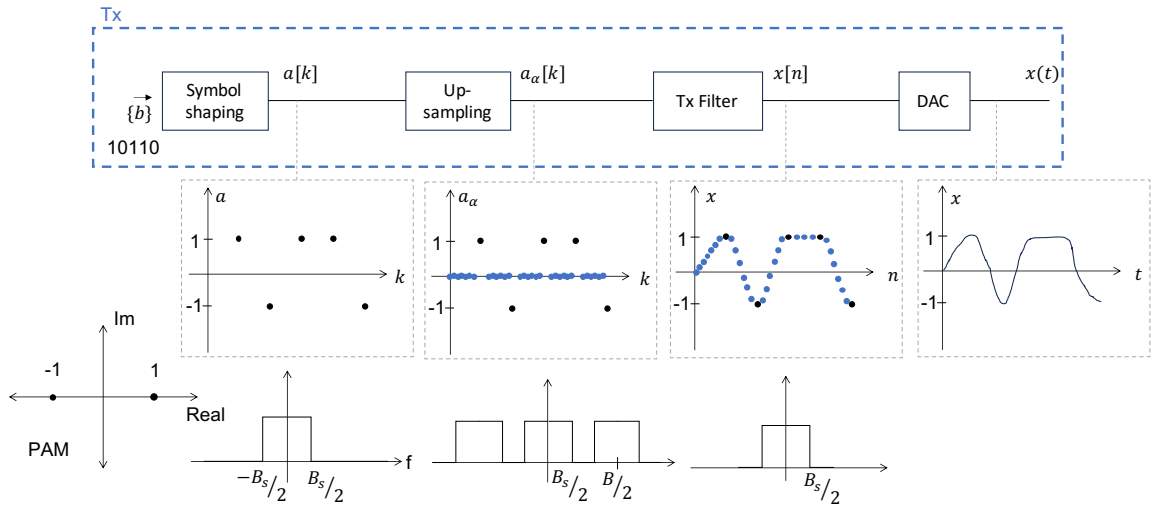


Figure 2.3 – Transmission chain for baseband signals.

The transmitter chain is composed of different steps that are described in Figure 2.3 in a baseband context. First, the binary stream is converted into symbols a_k, two symbols -1 and 1 to represent respectively the bit 0 and 1, this symbol modulation is called Pulse Amplitude Modulation (PAM). Other modulations exist and allow to decrease the BER with specific channel or propagation conditions. Then this signal $a[k]$ is up-sampled with α factor, and filtered to obtain the digital signal $x[n]$. The filter is designed to prevent symbol interference. Finally, this digital signal is converted to an analog one thanks to the Digital to Analog Converter (DAC).

The transmitted signal $x(t)$ can be express as

$$x(t) = \sum_{k=-\infty}^{+\infty} a[k] h_e(t - kT_s), \tag{2.1}$$

where h_e is the formatting filter, and T_s corresponds to the sampling time which corresponds to Nyquist theorem, and $B_s = \frac{1}{T_s}$.

B. Receiver

At the receiver side, the processing chain is the opposite of the transmission chain as represented in Figure 2.4. The received signal $y(t)$ is firstly converted to digital signal $y[n]$ and then synchronized to choose the corrected sample during the down-sampling. After the synchronization, the signal is filtered and down-sampled. Finally, the symbols are demapped to obtain the estimated binary stream. The objective is to obtain the lowest BER and to reduce it the receiver implements a Forward Error Correction (FEC) after the symbols demapping, not present in Figure 2.4 because it is out of the scope of this PhD.

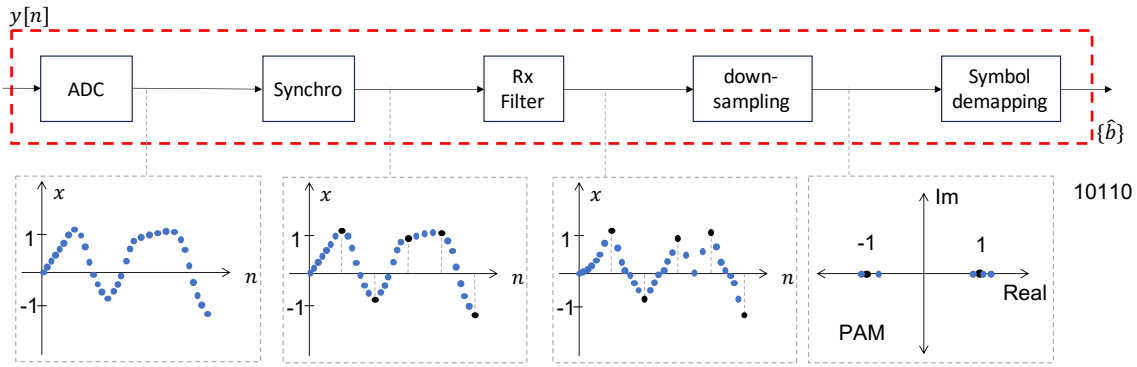


Figure 2.4 – Reception chain for baseband signals.

C. Limit of baseband transmission

The baseband signal is typically used for wired transmission communication. The frequency of signals does not allow communication over long distances. In addition, baseband transmission uses the entire bandwidth of the transmission medium, which limits the number of independent channels that can operate simultaneously without interferences. This reduces the spectral efficiency compared to broadband transmission techniques. Because baseband signals operate at lower frequencies, they are more susceptible to noise and interference, including ElectroMagnetic Interference (EMI) and crosstalk from adjacent channels or transmission lines. This can degrade signal quality and limit achievable data rates.

2.1.2 RF Transmission

To overcome the problems of baseband signals, such as long-distance transmission, it is possible to use carrier frequency transmission. Carrier frequency modulation allows transmitting the signals at high frequency. In this section, two technologies are presented single-carrier and multi-carrier.

A. Single-carrier

Figure 2.5 shows the transmitter chain in carrier frequency transmission. The first difference with the baseband chain is the symbol shaping, here the modulation can be done in the complex domain to get I and Q paths, represented by x_I and x_Q. The symbol modulation example here is 4-Quadrature Amplitude Modulation (QAM) with 2 bits per symbol, but there are many other QAM modulations to represent more bits per symbol. The rest of the chain is then very similar to the baseband chain but operates on each part of the complex signal. After the DAC, a Local Oscillator (LO) is introduced to modulate the signals at a chosen carrier frequency. Both parts of the complex signal are processed by the Power Amplifier (PA) and emitted by the antenna.

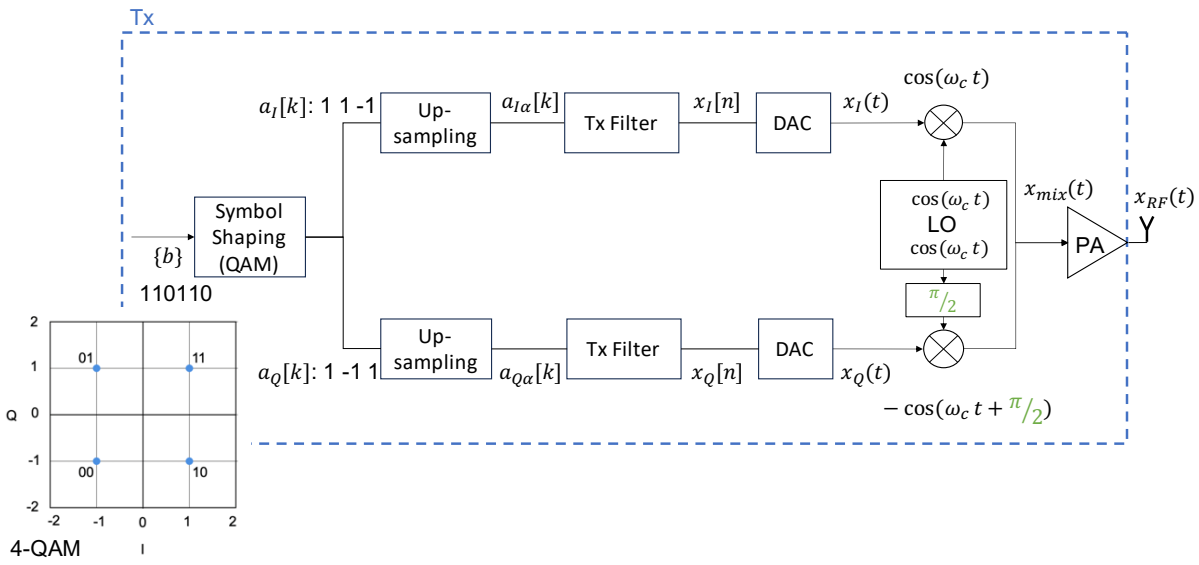


Figure 2.5 – Transmission chain with carrier frequency.

The complex analog signal can be written as $\underline{x}(t) = x_I(t) + jx_Q(t)$ after the DAC. Then the signal is modulated by the LO at a chosen carrier frequency here f_c with $\omega_c = 2\pi f_c$, and the complex signal [1] at the output of the LO is expressed as:

$$\underline{x}_{mix}(t) = \underline{x}(t)e^{j\omega_c t}, \tag{2.2}$$

$$\underline{x}_{mix}(t) = x_I(t)\cos(\omega_c t) - x_Q(t)\sin(\omega_c t) + jx_Q(t)\cos(\omega_c t) + jx_I(t)\sin(\omega_c t). \tag{2.3}$$

This signal can also be expressed as:

$$x_{mix}(t) = \Re(\underline{x}_{mix}(t)), \tag{2.4}$$

$$x_{mix}(t) = x_I(t)\cos(\omega_c t) - x_Q(t)\sin(\omega_c t), \tag{2.5}$$

where $\Re$ stands for the real part of the complex number. Finally the signal $x_{mix}(t)$ is amplified by the PA and the transmitted signal is expressed as:

$$x_{RF}(t) = G_{PA} \times x_{mix}(t), \tag{2.6}$$

which can be expressed with the baseband model:

$$\underline{x}_{RF}(t) = G_{PA}\underline{x}_{mix}(t), \tag{2.7}$$

$$\underline{x}_{RF}(t) = G_{PA}|\underline{x}_{mix}(t)|e^{j(\angle \underline{x}_{mix}(t))}, \tag{2.8}$$

where $|.|$ denoted L1 norm, and $\angle$ represent the angle of $\underline{x}_{mix}(t)$. (2.8) expression will be useful in Chapter 4.

This signal is then transmitted via the wireless propagation channel. This propagation channel is a multipath channel defined by its impulse response h. Therefore the received signal $y(t)$ can be expressed as:

$$\underline{y}(t) = h * \underline{x}_{RF}(t). \tag{2.9}$$

The path of the channel depends on the configuration of the environment propagation, such as the signal bouncing off different surfaces with different paths before arriving at the

1. All complex variables will be underlined in the rest of the PhD.

receiver. For example, considering a 3-path channel the received signal can be expressed as:

$$\underline{y}(t) = h_1\underline{x}_{RF}(t) + h_2\underline{x}_{RF}(t-t_2) + h_3\underline{x}_{RF}(t-t_3). \tag{2.10}$$

At the receiver side, the interesting signal has been convoluted by the channel, and the BER after demapping symbols can be affected by the channel. To remove this effect, the receiver implements a channel equalization. The objective is to estimate h^{-1} and convolute the received signal by h^{-1} to obtain an estimation of the transmitted signal. Several techniques exist to equalize the propagation channel [10, 33] such as using a known part of the signal to estimate the difference between the expected signal and the received one.

B. Multi-carrier

Multi-carrier communication is a technique where data is simultaneously transmitted over multiple carrier frequencies. It is possible to transmit multiple data streams in parallel, leading to efficient spectrum utilization and high data rates. The transmitted signal at the output of a multi-carrier transmitter scheme can be expressed as:

$$x_u(t) = \sum_{n=-\infty}^{\infty}\sum_{k=1}^{K} a[n,k]G_{T_u}(t-nT_u)e^{2j\pi f_k t}, \tag{2.11}$$

with K the number of subcarriers, G_{Tu} the formatting filter, f_k the subcarrier frequency and T_u the symbol time. The first sum corresponds to the elements transmitting in the time domain, and the second sum corresponds to the subcarriers.

The most known example of multi-carrier communication is Orthogonal Frequency Division Multiplexing (OFDM), but other multi-carrier schemes exist as well, such as Discrete Multitone Modulation (DMT) [59] or Filter Bank Multi-carrier (FBMC) [30].

The transmitted signal at the output of an OFDM transmitter scheme can be expressed as:

$$x_u(t) = \sum_{n=-\infty}^{\infty}\sum_{k\in\Omega} a[n,k]\Pi_{T_u}(t-nT_u)e^{2j\pi f_k t}, \tag{2.12}$$

with Ω the subcarrier ensemble, Π_{Tu} the formatting filter which is a gate, f_k the multi-carrier frequency which is linked to the subcarrier index k and the subcarrier spacing Δ_F with the relationship

$$f_k = k\Delta_F = k\frac{F_e}{N} = \frac{k}{NT_e}. \tag{2.13}$$

To ensure orthogonality, in the presence of a multipath channel model, a Cyclic Prefix (CP) is added to extend the symbol size from T_u to T_s and ensure having a complete period of the subcarrier frequency, with the presence of several paths with different delays if there are lower than the CP duration. The signal can be expressed as:

$$x(t) = \sum_{n=-\infty}^{\infty} \sum_{k\in\Omega} a[n,k]\Pi_{T_s}(t-nT_s)e^{2j\pi\frac{k}{N}\left(\frac{t}{T_e}-N_{\mathrm{CP}}\right)} \tag{2.14}$$

with N_{CP} the number of samples added to extended the symbol size. In particular, if the signal is sampled at the frequency F_e, the term in the second sum corresponds to an inverse discrete Fourier transform of the input sequence of symbols at time index n, corresponding to an inverse discrete Fourier transform.

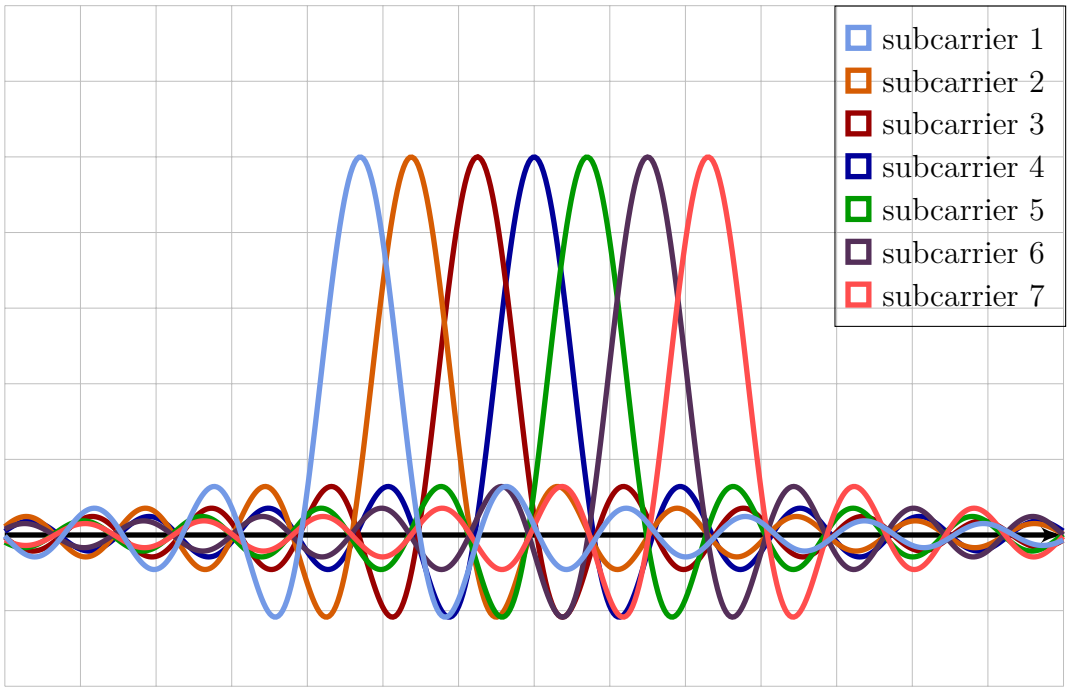


Figure 2.6 – Spectrum of OFDM subcarriers.

Figure 2.6 presents how OFDM divides the available spectrum into multiple orthogonal subcarriers, allowing for efficient use of the spectrum. Each subcarrier can be modulated independently, following the process described in Figure 2.5. OFDM presents several advantages such as the resistance to frequency selective fading which is a common issue in wireless communication where different frequency components of a signal experience different levels of attenuation and delay. OFDM is highly flexible and can be adapted to various communication standards and requirements. It is used in many wireless standards such as Wi-Fi, LTE, WiMAX, and digital television broadcasting.

OFDM simplifies frequency equalization compared to single-carrier modulation. OFDM can support multiple access schemes like Orthogonal Frequency Division Multiple Access (OFDMA), enabling efficient sharing of the spectrum among multiple users or devices. Overall, OFDM combination of spectral efficiency, robustness against various channel impairments, flexibility, and compatibility with various communication standards makes it a widely used and important technology in modern communication systems.

C. Conclusion

The transmission chain considered in this PhD is the multi-carrier transmission chain, in particular the OFDM one. The chain is composed of different hardware components to transform the binary sequence to a symbols sequence and then modulate the signal to transmit it with the carrier frequency. The hardware components of the transmission chain have some manufacturing defects, which create impairments in the transmitted signals, the RFF. The next section presents the different impairments and the RFF identification.

2.2 RF impairments and RFF definition

The RFF of a transmitter is a unique signature created by the hardware components of the transmission chain, which appears in the transmitted signals. The transmission chain is composed of different components presented in the previous section, and is detailed in Figure 2.7. First of all, the binary source information data is converted into symbol sequences thanks to symbol modulation, presented in the previous section, for example, the 4-QAM. Then a DAC transforms the complex signal into the analog domain to yield $x(t)$. The LO modulates it at the carrier frequency f_c, and the PA amplifies the signal, creating $x_{ant}(t)$ for transmission via the antenna.

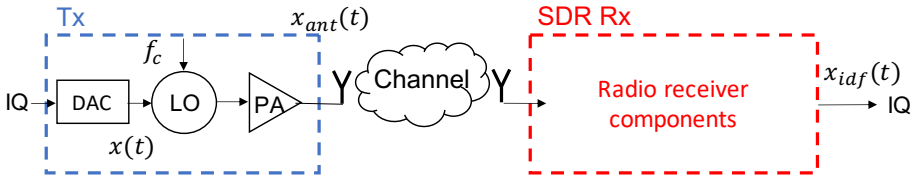


Figure 2.7 – Transmission and reception chains with RFF.

Devices	Impairments	References
Clock	Clock jitter	[119]
Digital to Analog Converter	Sampling error	[112, 75]
Local Oscillators	Phase Noise	[126]
	Carrier Frequency Offset	[106]
	I/Q imbalance	[110, 107]
Power Amplifier	Non linearity	[75]

Table 2.1 – Relationship between the hardware components and the impairments used for RFF identification.

All those components distort the signal and create the signature called the RFF of the transmitter denoted $\mathcal{F}_{\mathrm{RFF_{Tx}}}$. The emitted signal could be modeled by:

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{RFF_{Tx}}}\left(x(t)\right), \tag{2.15}$$

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{PA}} \circ \mathcal{F}_{\mathrm{LO}} \circ \mathcal{F}_{\mathrm{DAC}}\left(x(t)\right), \tag{2.16}$$

where $\circ$ represents the function composition operator which expresses the successive processing of the transmission chain, each stage creating a distortion function. $\mathcal{F}_{\cdot}$ represents a component behavior, including its distortion. The nested functions of (2.16) show the impact of each component and the difficulty of extracting features and modeling the transmission with RFFs. The LO adds distortions related to the frequency called Carrier Frequency Offset (CFO), gain and phase called In Phase - Quadrature (IQ) imbalance, and Phase Noise (PN). The PA impacts the gain and the phase in particular the PA introduces non-linearity in the complex amplitude of the signal. Table 2.1 presents the hardware components and the corresponding impairments, the papers referenced in this table correspond to works that used and present the impairments.

In Figure 2.7, 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 $\mathcal{F}_{\mathrm{channel}}$. The red block

Rx represents the receiver with its components (not detailed in the model but similar to the Tx reversed model) and its distortion function called the RFF of the receiver, $\mathcal{F}_{\mathrm{RFF}_{\mathrm{Rx}}}$. Investigating the impact of the receiver is beyond the scope of this PhD and we only consider a unique receiver for identification. However, it is important to note that the receiver may be illegitimate, and so may not have much information about the transmitter and the transmission channel, and so may be missing information to correctly extract the RFF impairments. The received signal x_{idf} can therefore be expressed as

$$x_{\mathrm{idf}}(t) = \mathcal{F}_{\mathrm{RFF}_{\mathrm{Rx}}} \circ \mathcal{F}_{\mathrm{channel}} \circ \mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}\left(x(t)\right). \tag{2.17}$$

$\mathcal{F}_{\mathrm{channel}}$ models the propagation environment of the signal between the transmitter and the receiver used to capture the signal for RFF identification. This propagation environment is defined by the location of devices, the relative position of transmitters and receiver(s), the noise level, the interfered signals, etc.; these also influence the received power. All these disturbances make difficult the RFF identification. This issue is largely studied in the SoA and will be called channel or environmental condition impact in the rest of the PhD. In the next chapter, a preliminary study of databases will highlight the impact of the environmental conditions on RFF identification and the database design biases.

2.3 Application Contexts

Recently, the number of contributions on RFF identification has increased [46, 115], and presents diverse application contexts such as IoT and cybersecurity, authentication, or defense. The application context of RFF identification is important to consider because it induces different knowledge of the transmitter(s) and emitted signals. Therefore, the different applications lead to different identification scenarios.

2.3.1 Authentication to enhance security

First of all, the RFF identification can be used to enforce the security of device authentication. For example, Guo et al. [34] use the term "1 to 1 authentication" to verify if the RFF of a device matches its MAC address [37]. In this context it is possible to use the RF impairments to enhance security levels in wireless systems, this supposes to have access to the devices to create an identification system that can recognize the RFF of each device in the wireless network.

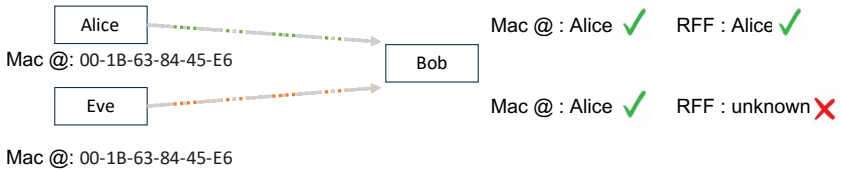


Figure 2.8 – Authentication to enhance security.

The legitimate receiver, called Bob in Figure 2.8, received a signal from the authorized device Alice, Bob has to check the MAC address to recognize Alice and ensure the authentication by checking the RFF. If a malicious device, Eve, impersonates the MAC address of Alice, the RFF checking will not correspond and allow to reject the authentication. In their paper, Guo et al. [34] also present the "1 to N" authentication as recognizing an authorized device but not which one it is. The identification system has two classes authorized and unauthorized devices.

2.3.2 Authentication with reduced overhead

The RFF identification can be used as an energy-efficient technique for the transmitter because it reduces the size of the transmitting packet. It is particularly attractive for short-packet IoT devices where the overhead of authentication is important. Moreover, it is a tamper-proof solution for authenticating low-power/computationally capable IoT devices [47]. The low-power devices are subject to spoofing attacks, because of poorly designed software security systems. Moreover, due to its openness, wireless networks are more vulnerable to malicious attacks than traditional wired networks. Cybercriminals are taking advantage of these vulnerabilities to impersonate. However, hardware impairment-based authentication hardened the security and required fewer power resources for the transmitter. Contrary to the first application, here no key is required to identify devices so the transmitter can adapt and skip the MAC address transmission to reduce power consumption. Contrary to the previous application, in this context, the authentication only relies on the RFF identification.

2.3.3 Defense or Attack

Finally, the RFF identification can be useful in a defense context or attack. For instance, cybercriminals are taking advantage of security vulnerabilities, and a malicious user can implement software systems to cover the tracks of its behavior. In this context,

Polak et al. [75] present a work that concentrates on breaking criminals' anonymity in wireless systems. This approach can be used by digital forensics to exploit the standard assumption that a criminal at some point in time employs their true identity. In another paper, Polak et al. [77] suggest testing devices from a pool of suspects to decide which one was most likely used during the crime. In this context, it is important to note that this type of approach can be complex to implement in practice, as it requires specific signals from suspects to improve identification.

In this PhD, the term "sensitive devices" is introduced in the title to address the defense context in particular the cyber electronic, and allow the detection of non-legitimate electronic devices by wireless transmission. Once a non-legitimate device is detected, a defense strategy could be to interfer its transmission using a jammer to alter the eavesdropper or a beamforming approach to isolate the transmissions between the legitimate transceivers.

2.3.4 Conclusion

The RFF identification can be useful in different application contexts such as IoT authentication or military defense. As this thesis was funded by the DGA, the considering application context is the defense. It could be interesting to be able to differentiate allied transmitters from unknown and therefore potentially enemy transmitters.

2.4 Identification System

To identify the transmitter with RFF, the signal should be captured and classified among the different potential candidates. In the SoA, Software Defined Radio (SDR) is massively used in database creation to record the signals or as transmitters due to their flexibility and accessibility. A SDR is an RF wireless communication system where the traditional hardware components of a radio, such as mixers, filters, amplifiers, modulators/demodulators, and detectors, are implemented using software on a computer or embedded system. In SDR, most of the signal processing functions are performed using software running on a general-purpose computer or specialized Digital Signal Processing (DSP) hardware. This allows for greater flexibility, reconfigurability, and adaptability in the radio system [64, 20]. One of the key advantages of SDR is its ability to support multiple communication standards and protocols through software updates or reconfiguration,

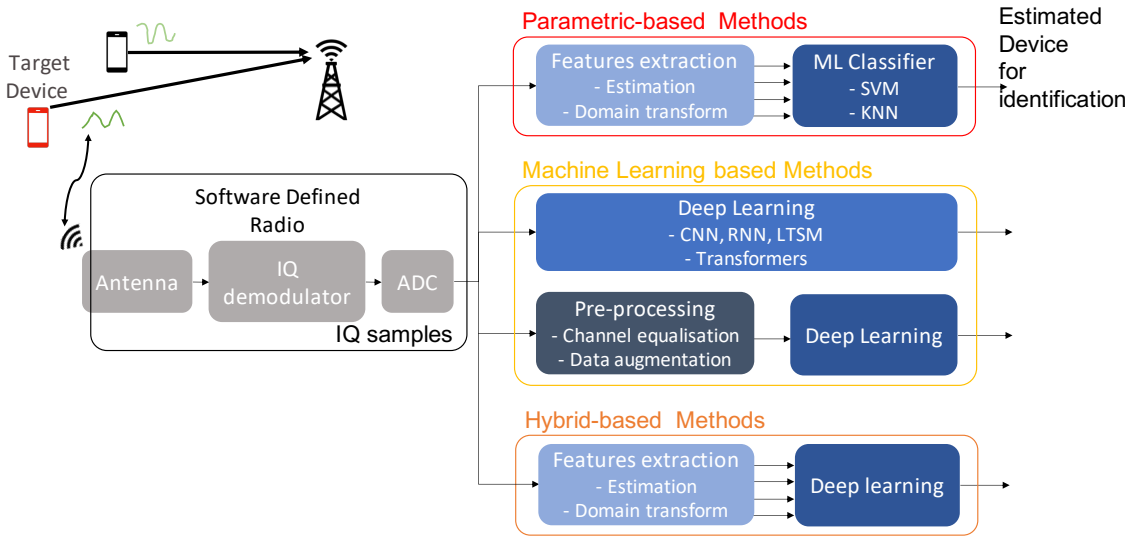


Figure 2.9 – SoA identification solutions.

eliminating the need for hardware changes. This makes SDR ideal for applications such as military communications, civilian radio, and amateur radio, where interoperability and flexibility are crucial. Additionally, SDR enables the development of advanced features such as dynamic spectrum access, cognitive radio, and adaptive modulation techniques. Moreover, the SDR allows us to develop a system independent of modulation to detect only faults thanks to IQ samples. In RFF identification, presented in Figure 2.9, the SDR is used to capture the signal as it can record large bandwidths and store the raw IQ samples before applying specific post-processing to help with signal classification. After recording, the complex signal become the input of the identification/classification system which attributes the signal to a device or a device group (authorized or not).

In the SoA, the RFF identification is based on three different family methods, represented in Figure 2.9. Primarily, the parametric methods combine feature extraction and classification thanks to Machine Learning (ML). Since 2018, the DL which takes raw IQ samples in input, has been massively explored especially for blind applications, DL is so the second method. Since 2021, some hybrid methods have appeared combining feature extraction and DL classification.

In parametric methods, the extraction step uses expert-defined features based on the physical properties of RF signals. These methods are adapted for a small number of devices (<100) but are not adapted for scale applications. Moreover, they required knowledge of the communication protocols [85], which is not realistic in the defense application

context, in particular in our use case. To address scale or blind applications such as defense or attack, the SoA proposes to take advantage of DL performance to perform classification. The hybrid methods are proposed to ensure the RFF classification by DL, the feature extraction highlights relevant device impairments that help the network to classify following the RFF.

2.5 Parametric-based methods

This section introduces the first identification solution based on the parametric methods. These methods exploit the intrinsic and unique nature of the impairments to identify the device and are composed of two steps: feature extraction and classification. Both parts of these methods can be done with multiple techniques. The next subsection presents the different features and extraction techniques and the one after presents the classification methods.

2.5.1 Features extraction

Feature extraction has been largely explored in the SoA and the possibilities are multiple due to the large choice of features, induced by the different impairments impact. The feature extraction methods are separated into two families depending on the part of the signals used to classify, Figure 2.10 presents a classification of extraction techniques following transient-based and asymptotic-based methods.

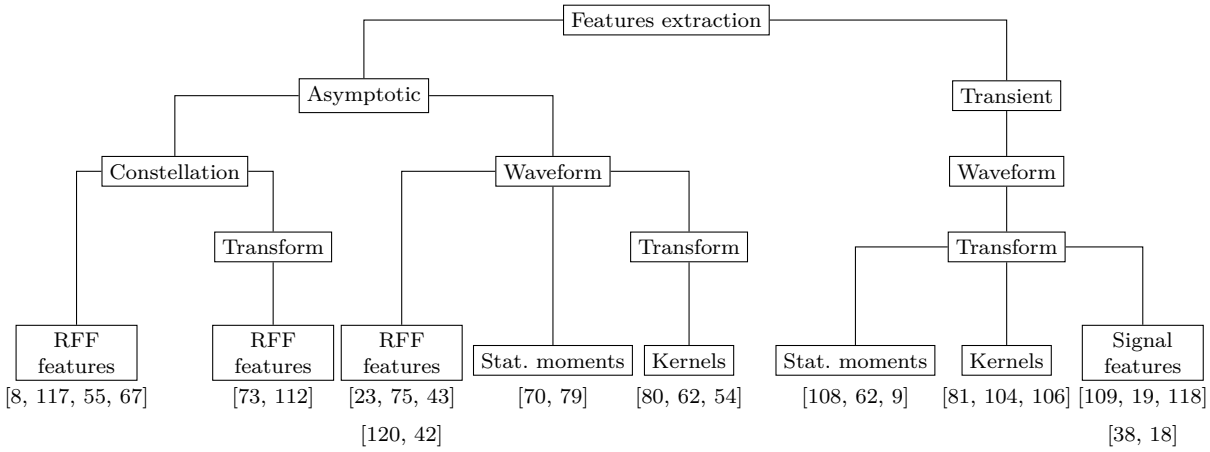


Figure 2.10 – SoA feature extraction classification.

The transient part of the signal opposite to the steady-state or asymptotic behavior, corresponds to the sudden change in the signal at the beginning of the communication seen in Figure 2.11. This change leads to nonlinear perturbations that are specific to an emitter. The fingerprint can be based on caracteristic features estimation [109, 118, 38, 19] or statistics estimation [18, 108, 62, 9] or domain transform kernel values [81, 104, 106]. The features estimations concern the transient particularity such as the energy envelope or duration of the transient signal.

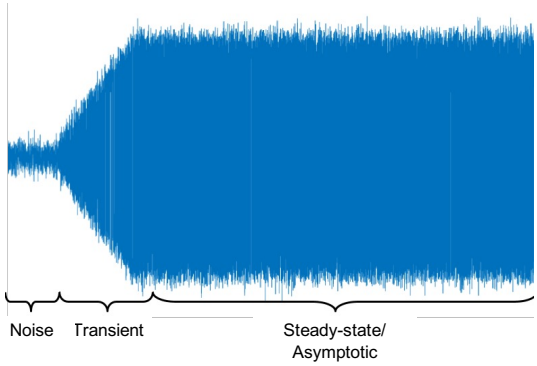


Figure 2.11 – Established signal representation.

The steady-state or *Asymptotic* part of the signal corresponds to the moment of emission after the transient part. The study of this transmission part is separated into two categories: i) waveform domain and ii) constellation domain, also called modulation domain in [8] contrary to waveform domain where the signal is not demodulated. Demodulation at the receiver requires synchronization and perfect knowledge of the modulation scheme, an assumption that is not always verified. Then each class is separated into two or three subclasses, i) estimation features such as CFO or PA coefficient for Waveform-based [75, 42, 43, 23, 120] and IQ imbalance for constellation-based [8], ii) transform-based extraction such as wavelet domain coefficients [54], iii) statistic moments of the signals [70, 79].

To summarize, four feature extraction techniques exist and can be applied to the transient part or asymptotic part of the signal to extract emitter characteristics.

— The first family of features corresponds to physical impairment value estimators, such as the PN or the CFO, that are often estimated directly from the signal or after a domain transform, called RFF features in Figure 2.10.

— The second family of features corresponds to statistical moment value estimators, such as variance, kurtosis, and skewness, called statistic moments in Figure 2.10.

— The third family corresponds to the use of domain transforms such as Wavelet or frequency domain and the use of the coefficients as fingerprints, called kernels values in Figure 2.10.

— The fourth family is only used for transient signals and corresponds to signals feature extraction not directly from the impairments estimator.

These different features can be combined to enhance the identification process. The next subsections present the different techniques used in the SoA presented in Figure 2.10, with firstly the transient-based research and then the asymptotic-based one. Tables 2.2 and 2.3 summarize the subsection. The second column presents the classifier used, such as K Nearest Neighbor (KNN) [17] and Support Vector Machine (SVM) [16] that are presented in the next subsection. The third column gives the domain used to extract features, and then the fourth presents the features. Finally, the fifth one lists the type of results, experimental or simulation-based.

Ref.	**Classifier**	**Transform**	**Features estimation**	**Exp. or Simulation**
Domain Transform kernels values				
[106]	Genetic Algorithm	Wavelet TF [2]	Wavelet coefficients	Simulation
[81]	-	Spectrum	-	Spectrum analysis
[104]	PSD correlation	PSD	PSD coefficients	Exp: 100 Wifi emissions
Domain Transform and statistic moments estimation				
[108]	KNN	Short Time FT	Energy envelope Statistics	Exp: 7 Bluetooth devices
[62]	SVM	PSD, FrFT	Statistics moment (2,3,4)	Exp: 10 walkie-talkie
[9]	SVM	Gabor TF	Standard deviation, variance, slope, and kurtosis	Exp: 4 transmitters
Domain Transform and signals features estimation				
[109]	PNN [3]	Hilbert TF	PCA: Amplitude profile	Exp: 8 WiFi transmitters
[19]	1NN	Hilbert TF	PCA: Spectral feature	Exp: 50 identical cards
[18]	Mahalanobis	Hilbert TF	Fisher LDA	Exp: (50 COTS)
[118]	SVM	Hilbert–Huang TF: EMD and Hilbert TF	Sum of energy Duration of transient signal Duration of the max. energy point:	Exp: 8 emitters
[38]	Multivariate Statistical Classifier	DWT	DWT coefficients, amplitude, phase, etc.	Exp: 30 transmitters

Table 2.2 – Transient-based methods.

2. Transform (TF)
3. Probabilistic Neural Network (PNN)

A. Transient-Based methods

The transient-based methods are decomposed into two steps, first transient extraction and then feature extraction with different techniques. Different transient detection techniques have been investigated such as amplitude-based and phase-based, for example, Ureten et al. [109] considered transient amplitude features while Hall et al. [36] used transient phase features and [104] proposed a variance-based approach. Soltanieh et al. [103] propose a review of RFF techniques and present 6 different methods to extract the transient part of the signal.

a. Domain transform kernels values: The transient-based features are structurally blind because they do not rely on modulation at all. For transient-based feature extraction, the authors propose to estimate some characteristic features from the waveform domain. The most popular technique is to use domain transform to highlight some differences in the signals. For example, in 1995 several works proposed Wavelet Transformation (WT) [15, 106] and used the wavelet coefficients as fingerprints. Then, Remley et al. [81] present in 2005 different Power Spectral Density (PSD) of received signals from different transmitters and show the difference. This work reveals the interest of the frequency domain for RFF identification, however, no ML classification is proposed. In 2008, the PSD is used as a featured extractor by Suski et al. [104] considering PSD coefficients.

b. Domain transform and statistic moments estimation: While it is possible to directly use coefficients as fingerprints, Ur Rehman et al. [108], apply a short time Fourier Transform (FT) and use the statistical moments of the energy envelope to characterize the devices and then used a KNN to classify the device. In the same idea, Lin et al. [62] propose PSD transform and create a fingerprint vector based on statistical moments of the PSD.

c. Domain Transform and signal features estimation: The third interesting domain transform is the Hilbert transformation [19, 109, 18] which creates complex-valued analytic functions. This transformation is used to compute the instantaneous attribute of a signal such as amplitude [109], phase or spectral features [19, 118] that are considered as fingerprints. Other transformations are used such as Gabor transform [9] combining with statistics features extraction or Discrete Wavelet Transform [38]. Hall et al. [38] propose to use Discrete Wavelet Transforms (DWT) coefficients, amplitude, phase, and many other features to compose the fingerprinting vector and then use a Multivariate statistical classifier to perform identification.

The features can be used independently or together to create a strong fingerprint vector based on different metrics such as statistics in [9] or time characteristic of transient signals and energy in [118] or Discrete Wavelet transform coefficients, amplitude, phase and others in [38]. The work of Xie et al. [115] presents a survey of Physical-Layer authentication in wireless communication. The first part of this paper presents the passive physical layer authentication which corresponds to RFF identification. This survey offers an interesting classification of parametric methods with a larger panel of work especially when they detail the number of features used to classify devices. The classification proposed by Xie et al. presents some very specific methods that are not presented here to alleviate the SoA classification.

Ref.	Classifier	Transform	Features estimation	Exp. or Simulation
Constellation - Impairment features estimation				
[8]	SVM KNN	-	I/Q origin offset, Frequency error, SYNC correlation	Exp: ORBIT nodes [69]
[117]	KNN	-	IQ imbalance estimation	Exp: 5 Tx simulation
[55]	Own classifier	-	Phase shifting	5 Tx nodes
[67]	Non parametric bayesian model	-	Phase shifting and frequency offset	Simu: 1 to 6 devices Exp: 4 ZigBee devices
Constellation - Domain Transform and impairment features estimation				
[73]	Own hybrid classifier	DCFT	DCFT, frequency , modulation and IQ offset	Exp: 54 ZigBee devices
[112]	Error probability	FFT	Non linearity coefficients	Exp: 6 Micaz sensors nodes
Waveform -Impairment features estimation				
[23]	LRT	-	PA estimation	Modelisation
[75]	LRT	-	PA estimation and DAC	Exp: 8 measures of PA
[43]	Hypothesis test	-	CFO estimation	Exp: 2 transmitters
[42]	MSE with Kalman predicted CFO	-	CFO estimation	Simu: false alarm detection
[120]	Visibility Graph	-	CFO estimation	Exp: 2 devices
Waveform - Statistic moments estimation				
[70]	MDA/MLE	-	Statistics moment (2,3,4)	Exp: 4 ZigBee devices
[79]	MDA/MLE	-	Statistics derive from amplitude phase and frequency	Exp: 7 ZigBee devices
Waveform - Domain Transform kernels values				
[80]	KNN	PSD	Normalized PSD coefficients	Exp: 3 WiFi transmitters
[62]	SVM	Bispectrum	AIB, CIB, RIB and SIB	Exp: 10 walkie-talkie
[54]	MDA/MLE	Dual tree Complex WF	Wavelet coefficients TD statistics	Exp: 4 Cisco devices

Table 2.3 – Steady-state based methods.

B. Asymptotic-Based methods

Unlike transient methods, asymptotic methods are divided into two families, constellation-based and waveform-based. In the same way, it is possible to extract different features by estimation or domain transform to characterize the transmitter fingerprint.

a. Constellation-Based

Impairment features estimation: In the constellation-based techniques, the signal is demodulated before features extraction offering the possibility to estimate the impairment values, such as phase shifting [55, 67], frequency offset [67, 8], or IQ imbalance [8, 117]. The use of such parametric methods is strongly limited by the knowledge about the transmission chain, protocol, modulation, and the superposition of impairments. In PARADIS [8], Brik et al. propose to extract features by measuring artifacts in wireless frames in the modulation domain and then use ML to identify the different devices. Their method requires demodulating the signal and requires a priori knowledge on the receiver side. They established different metrics to characterize the device identity such as (i) frequency error, (ii) SYNC correlation, (iii) I/Q offset, (iv) magnitude error, and (v), phase error, and then combined the metric to create a fingerprint vector. Then two classifiers are implemented and evaluated, one using the SVM algorithm and the other using the KNN algorithm. The authors evaluated PARADIS on the ORBIT indoor wireless testbed facility [69].

Recently, Yuan et al. [117] also proposed a novel OFDM RFF method that relies on the hardware property of the IQ imbalance and nonlinearity of the transmitter together. First, they estimate the parameters of the nonlinearity of the transmitter and Finite Impulse Response (FIR) of the wireless multipath channel, with a Hammerstein system parameter separation technique. Then, they use estimation techniques for IQ imbalance compensation. Finally, they combine the nonlinear coefficients and the IQ imbalance parameters to produce the RFF characteristic vector. Then the RFF is used by a KNN to classify the signal.

Domain Transform and impairment features estimation: In constellation-based methods, some authors propose to employ the Differential Constellation Trace Figure (DCTF) to highlight impairments. The DCTF is a graphical representation of a signal in a complex plane, where each point on the diagram represents a symbol transmitted in the signal. Peng et al. [73] propose an identification solution based on four modulation features, that are DCTF, carrier frequency offset, modulation offset, and I/Q offset extracted from the constellation trace figure. To classify the signals they develop a hybrid

classifier that adjusts feature weights according to the channel conditions. In [112], Wang et al. propose a wireless physical-layer identification model based on the complete wireless transmission chain. They only consider the non-linearity of the transmitter front-end, and other hardware impairments are considered as additional noises. The feature extraction consists of domain change here spectral domain, and dimensionality reduction with a Linear Discriminant Analysis (LDA). The classification consists of matching all the reference fingerprints and assigned to the identity with the smallest distance score.

b. Waveform-Based

Impairment features estimation: In the Waveform domain, the estimation techniques usually used in wireless communication for impairment compensation are used to create a specific fingerprint. The three families of feature extraction presented at the beginning of this section can be used to extract features in the waveform domain of the asymptotic signal. The first method targeted a particular impairment in the waveform domain such as the CFO and PA non-linearity, the second method focused on statistical moments, and finally the domain transformation such as the PSD.

CFO: The CFO is caused by the LO imperfections, and can be used as fingerprint [42, 120, 43]. Hou et al. proposed a RFF identification scheme based on time-invariant CFO analysis [42]. In other work, they propose a time-varying CFO scenario [43]. Using the CFO as a fingerprint is not interesting because the LO is sensitive to the temperature and the CFO is impacted by it [122]. The CFO on its own (or without refinement) is not a relevant signature.

Power amplifier impairments: The PA is the last component in the wireless transmitter chain and adds some non-linearity in the signal. By studying the possibility of using the imperfections of PA and DAC, Polak et al. [75] show that the PA nonlinearity dominates the RF chain, in particular the DAC imperfections. In a second work, they propose to use spectral analysis to identify a device thanks to its PA non-linearity in the case of artificial data distortion introduction in attack context [76]. For this extraction, the parameters of the Volterra model for the corresponding PA are estimated and then a Likelihood Ratio Test (LRT) is used to authenticate the devices [23, 75].

Statistic moments estimation: It is possible to use statistic moments of order 2 (variance), 3 (kurtosis), and 4 (skewness), and combine them to characterize the signal. Two different works, based on ZigBee devices identification propose to use statistic moments to identify the transmitter thanks to a multi-dimensional analysis and a Maximum Likelihood Estimation (MLE) algorithm to perform the classification [70, 79].

Domain Transform kernels values: While some authors propose to compute specific algorithms to estimate the value of impairments such as CFO and PA, others propose to use domain transform to highlight specific behavior and obtain a signature. The PSD coefficients can be used in the asymptotic domain to create a fingerprint, the work [80] proposes an analysis of the RF receiver front-end on the classification accuracy. For the classification, Rehman et al. [80] use normalized PSD coefficients extracted from the preamble part of the signals.

The second interesting domain transform is the wavelet domain. For example, Klein et al. [54] address intra-manufacturer discrimination using identical model devices manufactured by Cisco. The fingerprint is based on DT-CWT and the classification is done by a Multi discriminant analysis and Maximum Likelihood Estimation. (MDA/MLE) processing.

It is possible to combine different features and different techniques. For example [62] proposes to perform Specific emitter identification (SEI) on transient signals with both PSD and Fractional Fourier Transform (FrFT) and proposes the bispectral transform to analyze the asymptotic signal. The bispectral transform analysis of the signal has the advantages of phase retention, scale variability, and time shift-invariance. Finally, the RFF vector is used by a SVM model which classifies the signal.

C. Features extraction conclusion

This subsection has presented several RFF identification works based on parametric methods and especially the different features and manner to extract them. The variety of features and extraction methods is impressive, but the features extracted can be affected by environmental conditions and so impact the classification recognition. The next section presents the most important parametric-based classification methods in the SoA.

2.5.2 Parametric-based Classification

The parametric-based methods combine feature extraction with a ML classification stage. The ML algorithms can be classified into four categories: supervised algorithms, unsupervised algorithms, reinforcement learning, and hybrid algorithms [5]. The most popular techniques for RFF are supervised and unsupervised algorithms.

A) Supervised learning

Supervised learning has the particularity to require a correctly identified training set of observations, with predefined classes. The correctly identified training set is used to train the classifier. For parametric methods in RFF, identification means that the fingerprints or features have been collected during the training step and stored in a labeled library. During the test, the fingerprint of an unknown device is computed and compared to the existing library to identify the device. The most important databases used for RFF classification are presented in the next chapter. In the rest of this section, a brief description of the main supervised ML is proposed. For a more detailed description, we refer the reader to the cited references.

a. The K Nearest Neighbor (KNN) algorithm is a ML algorithm that classifies a data sample thanks to the labels of the nearest data samples (neighbors). This technique is computationally efficient during the training step. However, the classification phase may entail higher computational demands compared to other algorithms, which can be an important issue in real-time, and constraints RFF identification applications. To determine the distance and determine the nearest neighbors, it is possible to use the Euclidean distance or the Mahalanobis or Minkowski distances. The KNN classifier is used in different works to classify the device following their fingerprint features KNN [19, 108, 8, 80, 117]. Danev et al. [19] propose to use the 1-NN to estimate the similarity between testing and reference signatures from a given class due to the reduced training required to perform the classification. While [108, 80] propose a 3-NN to classify the devices.

b. Support Vector Machine (SVM) is a supervised learning algorithm used for classifying data points based on labeled training samples. These samples typically consist of observables paired with reference fingerprints. SVM partitions the labeled dataset into two distinct areas on a multi-dimensional surface through the utilization of a separating function. This function can take various forms such as linear, polynomial, or sigmoidal. Given this partitioning, SVM functions as a binary classifier, making it adept at distinguishing between two devices directly or validating the asserted identity of a device. The SVM offers several benefits for fingerprint classification such as great accuracy and resilience against outliers. Compared to alternative methods, SVM demonstrates a lower susceptibility to overfitting. Its efficiency in binary classification is particularly advantageous during the verification phase. However, SVM's drawback lies in its potentially slow learning process, often demanding a significant amount of training time. The SVM classifier is used in different works of the SoA such as [62, 118, 19, 9]. [8] performed

the classification with KNN and SVM and showed that the SVM is more effective than KNN probably due to the SVM data pre-processing where the input was mapped onto a higher-dimensional space.

c. Bayesian Classifiers are statistical classifiers and they predict the class membership probability, that is the probability that a given sample belongs to a particular class. A subtype of the Bayesian classifier is the Naïve Bayes Classifier, which assumes that all variables contribute to classification and are mutually correlated. This assumption may be true for certain categories of fingerprints, particularly those originating from shared physical components like RF equipment. Bayesian methods offer advantages such as efficient adaptation of probability distribution without overfitting and the ability to work effectively with a limited number of training samples, which is beneficial in scenarios where acquiring a large dataset of fingerprints for training is challenging. Hall et al. propose to use a Bayesian classifier to perform transient detection in [36] and then for fingerprint classification in [37]. However, Bayesian classifiers tend to be less accurate compared to other classifiers.

d. Likelihood Ratio Test (LRT) [75] and [23] present a classifier based on a Likelihood ratio test. After computing the features extraction they obtain the parameter vectors describing the nonlinear aspects of the user's transmitters. Then, the probability of error of the receiver is minimized by a LRT to classify the emitter. In the same idea, [112] proposes to minimize the probability error or the mean square error compared to a predicting value in [42]. Finally, to complete the MLE algorithm, some authors propose a multi-dimensional analysis [79, 54, 70] before improving classification accuracy.

B) Unsupervised learning

refers to a class of algorithms that operate without a training set, the algorithms must find the hidden structures within unlabeled or unclassified data. In the context of device identification or verification, unsupervised algorithms are employed to group similar fingerprints from various logical devices into clusters. Unsupervised learning can be used to combat counterfeiting or identity spoofing attacks. It also removes many of the constraints imposed by the need for labeled databases. Various techniques exist in the SoA such as K-Means Clustering, hierarchical clustering, unsupervised Bayesian Learning [67], and Principal Component Analysis (PCA) a multivariate method for data compression and dimensionality reduction [38]. PCA aims to extract important information from data and present it as a set of new orthogonal variables called principal components [109, 19].

The PCA can be used to extract the most relevant features and reduce the size of the fingerprint vector.

2.5.3 Conclusion

To conclude, parametric methods require knowledge about the targeted signals to be able to estimate the channel or to demodulate the signal and obtain IQ imbalance estimation for example. The application context of defense, which is particularly focused in this PhD does not offer ideal conditions for signal knowledge. This is why the identification based on DL seems more adaptable. Moreover, even if the authors try to focus on impairments, other environmental biases are integrated. For example, in [117] the non-linearity includes the channel non-linearity and so the relative position between transmitters and the receiver.

2.6 Deep Learning 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. The first papers that proposing these methods use time domain signals as input. In the next section, the recent hybrid methods are presented that combined specific emitter features extraction with DL to improve the robustness of classification. In this section, we focus the SoA on DL solutions based on time domain signals, with and without pre-processing. Many DL architectures exist, in particular CNN is used to extract and classify RFFs [102, 94, 92, 91, 97].

Firstly, a short definition of the different types of DL architectures such as the Feed-forward Neural Network (FNN), the CNN, the Recurrent Neural Networks (RNN), and the Transformers, that are the relevant network in the SoA, are presented. Then the particularities of the different works of the SoA are presented in particular the pre-processing solutions are described.

2.6.1 Network presentation

A neural network is composed of nodes, each node computes an output by multiplying inputs with weights, adding a bias, and then using an activation function such as linear, ReLu, gaussian, sigmoïd, etc. The output can be expressed as

$$y = f(\sum_i w_i x_i + bias), \tag{2.18}$$

where, w_i is the weight, x_i the input and f the activation function. Figure 2.12 presents a lonely node and two parallel nodes. These nodes are associated with creating some layers such as a fully connected layer or convolutional layer. The size of the network depends on the number of network parameters (weight and bias) and the complexity of the network depends on the number of operations such as multiplication done. The complexity aspects are presented in Appendix A.

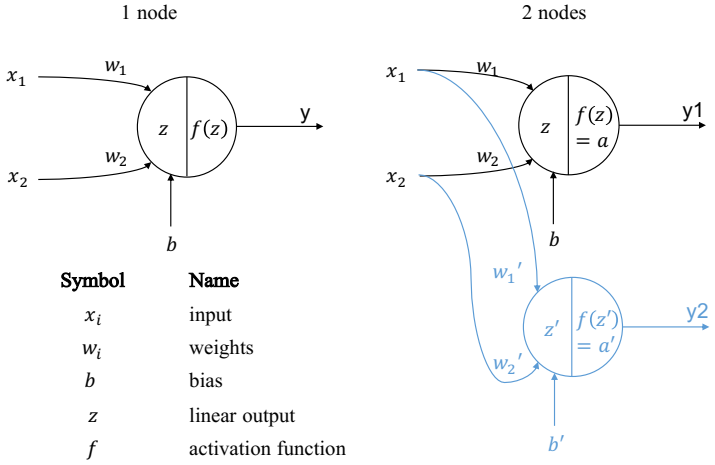


Figure 2.12 – Architecture of a neuron.

A Feedforward Neural Network (FNN), also called a fully connected neural network or dense network is a type of Artificial Neural Network (ANN) that consists of multiple layers of nodes, including an input layer, one or more hidden layers, and an output layer. A layer is composed of N parallel nodes. Information flows in one direction, from the input layer through the hidden layers to the output layer, without feedback loops. Figure 2.13 presents a fully connected network composed of 1 hidden layer, 5 input nodes, and 3 output nodes. All the nodes of layer L are connected to all the nodes of layers L-1

and L+1 with a corresponding weight associated with each connection. FNNs are widely used for supervised learning tasks such as classification and regression, where they learn to map input data to output predictions through a process of forward propagation and backpropagation [6, 88].

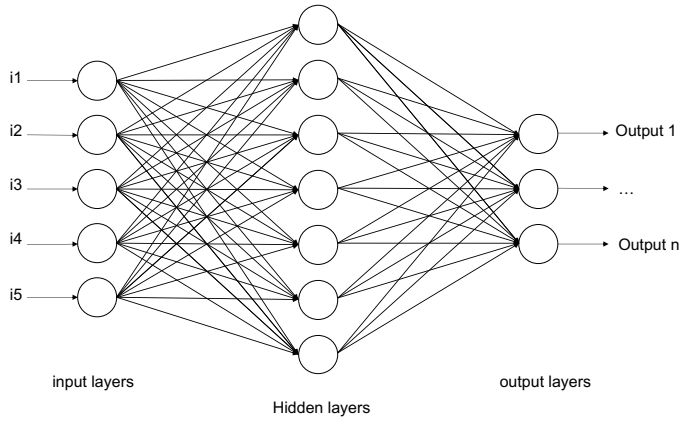


Figure 2.13 – Architecture of an FNN.

A Convolutional Neural Network (CNN) is a type of artificial neural network designed specifically for processing structured grid-like data, such as images. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers, as present in Figure 2.15. convolutional layers apply filters to the input data, enabling the network to learn hierarchical representations of features present in the input. The convolution is described in Figure 2.14, the blue and yellow cubes represent the input data, here the IQ samples in the context of RFF identification. The first and last two blocks of the sequences are created to preserve the same size of data between input and output, this is called zero padding, detailed in Appendix A. The first filter convolutes the data, and the result creates a vector of dimension 1, called a channel. Then the second filter convoluates the data and creates a second channel. Then the new vectors become the input of the next convolutional layer. The pooling layers downsample the feature maps generated by the convolutional layers, reducing their dimensionality and computational complexity. In Figure 2.15, a max-pooling layer is represented, this layer divided the data by 2, conserving the maximum value between 2 neighbor data. Finally, the CNN is often composed of fully connected layers at the end of the network, as it shown in Figure 2.15. In the classification context, the last layer is a softmax, which generates a normalized probability score, the total sum of the probabilities of which will be equal to 100%, or 1.

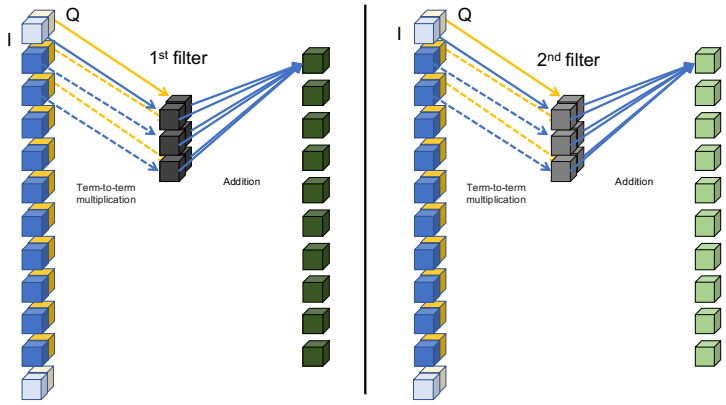


Figure 2.14 – Filter convolution in CNN for RFF application with IQ samples in input.

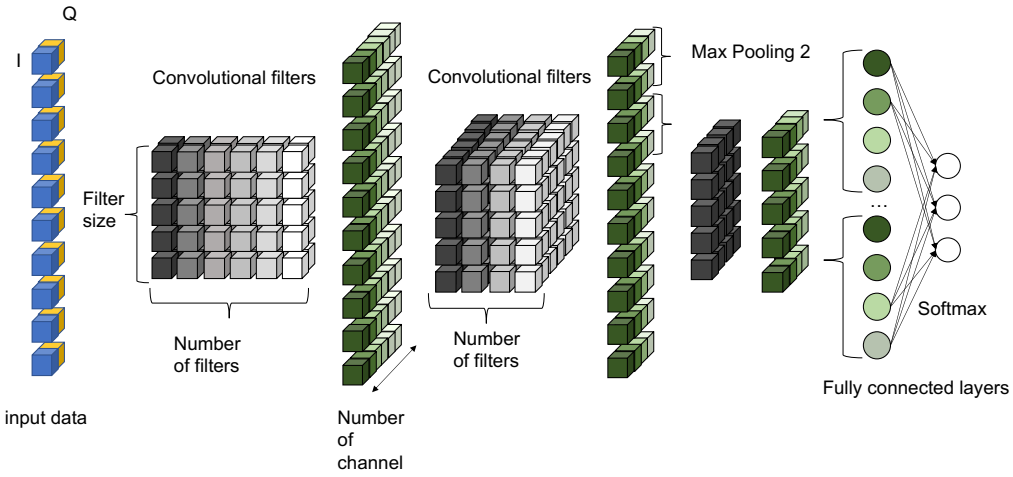


Figure 2.15 – Architecture of a CNN.

CNNs are widely used in image recognition, object detection, and other computer vision tasks due to their ability to automatically extract meaningful features from raw input data [61].

A Recurrent Neural Network (RNN) is a type of artificial neural network designed to process sequential data by incorporating feedback loops. Unlike feedforward neural networks, RNNs have connections that form directed cycles, allowing them to retain information over time. This architecture enables RNNs to learn patterns and dependencies in sequential data by processing each input in sequence and updating their internal state based on previous inputs. RNNs are commonly used in natural language processing, speech recognition, time series analysis, and other tasks where the order of the data is significant. Figure 2.16 presents how an RNN works with a language processing

example, the left part presents the principle, and the estimation of the letter will be used to estimate the next letter. The right part shows how RNN works. Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) are a type of RNN architecture, offering improvements in terms of their ability to capture long-range dependencies and mitigate the vanishing gradient problem. GRUs are simpler and more computationally efficient, while LSTMs are more powerful and versatile, making them suitable for a wide range of sequential data tasks [90, 97].

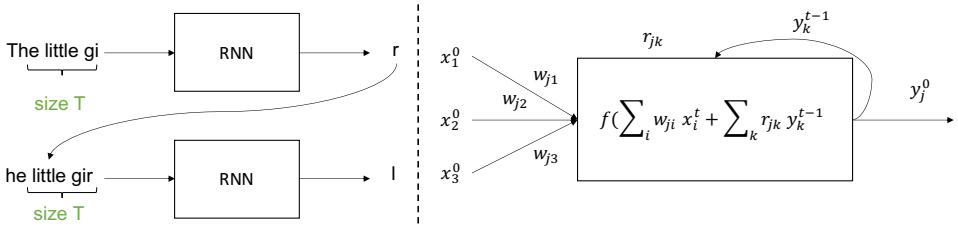


Figure 2.16 – Principle of a RNN.

Transformer architecture refers to a class of models designed for various Natural Language Processing (NLP) tasks. These models are characterized by their ability to process input data in parallel through multiple layers of self-attention mechanisms and feed-forward neural networks. Transformers have revolutionized NLP by enabling the modeling of long-range dependencies in text data and achieving SoA performance in tasks such as language translation, sentiment analysis, and text generation. The performances of transformers on temporal data interest researchers of the SoA for RFF identification [96, 97].

Generative Adversarial Network (GAN) is a type of artificial intelligence algorithm that consists of two neural networks, the generator and the discriminator, that are trained simultaneously through a game-like scenario. During training, the generator tries to produce increasingly realistic data to fool the discriminator, while the discriminator learns to become better at distinguishing real data from fake data. This process creates a feedback loop where both networks improve over time. The ultimate objective of Generative Adversarial Network (GAN)s is to train a generator network that can produce high-quality data samples that are indistinguishable from real data and can be useful for data augmentation. However, it can be used to separate trust devices and adversarial devices in the RFF context [87].

2.6.2 From signals to learning

DL has been increasingly used in classification for several years, especially for image classification. Inspired by these DL architectures, several authors propose to use a similar classifier for RFF identification. Figure 2.17 presents the general flow from signals to the trained network.

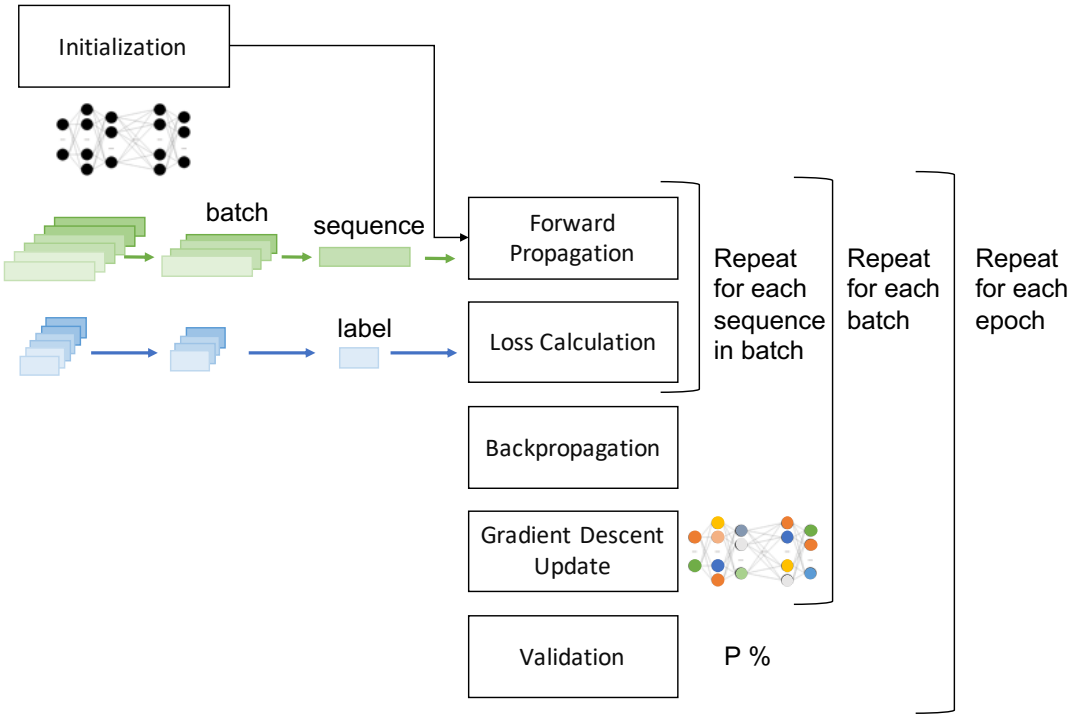


Figure 2.17 – Training process with DL techniques.

Firstly a database has been created by recording the signals from different transmitters thanks to a SDR. The database is so composed of complex signals from different transmitters, that each signal must correspond to a label to identify its emitter. Then the database is separated to form the training set and test set. To ensure good learning the database must be balanced between all transmitters.

Training a neural network involves adjusting its parameters, weights, and biases, so that it can effectively map the signals (input), to desired labels (output). Initially, the weights and biases of the neural network are usually set to small random values. These values determine how the network will initially respond to input data.

During forward propagation, the sequence of signal is fed into the network and computations are performed layer by layer until the output is generated. Each neuron in a layer

calculates a weighted sum of its inputs, applies an activation function to this sum, and gives the result to the next layer. Instead of updating the model parameters after each individual sequence (which can be computationally inefficient), the sequences of signal are grouped in batch which allows for more efficient computation by performing updates based on multiple sequences at once. Once the labels are generated for each sequence of the batch, they are compared to the desired labels using a loss function. The loss function quantifies how well the network predictions match the true target values. This loss function is generally a cross-entropy in classification problems but could be a mean square error function for regression problems. The core algorithm used to train neural networks is called backpropagation. It involves calculating the gradients of the loss function with respect to the network weights and biases. These gradients indicate the direction and magnitude of change required to reduce the loss. Using the calculated gradients, the network updates its weights and biases to minimize the loss. The most common optimisation algorithm used for this is gradient descent which iteratively adjusts the parameters in the direction opposite to the gradient to find the minimum of the loss function. However different optimizers exist that reduce the risk of trapping a local minima and are more efficient such as Adam (Adaptive Moment Estimation) which is the one used in this PhD.

The training typically involves repeating this process for multiple epochs until a stopping criterion is met. Within each epoch, the neural network iterates through all the batches of the training dataset. For each batch of the training dataset, the network performs forward propagation to compute predictions, calculates the loss between the predicted outputs and the true targets, and then performs backpropagation to compute gradients and update the model parameters (weights and biases). An epoch refers to one complete pass through the entire training dataset, during the training process. The stopping criterion could be reaching a maximum number of epochs, achieving satisfactory performance on the validation dataset, or observing no improvement in performance for several epochs.

During training, it is common to monitor the performance of the model on a separate validation dataset after each epoch. This helps track how well the model is learning and whether it is overfitting or underfitting the training data. Overfitting the training data is a common issue in DL which occurs when the network is too specialized for the training data and not able to perform on validation or testing set. After training is complete, the final performance is evaluated on a separate test set to assess how well the model generalizes to unseen data.

The presentation of the training network has revealed several hyperparameters that impact the training process and the results of the training. First, the database is sliced to obtain signal sequences, and the sequences are fed into the network. The size of the slicing windows depends on the network architecture and impacts the number of samples used to estimate the label. Then the batch size is important because it determines how many signal sequences the network sees between two updates. The batch size is important for network training because the number of updates per epoch depends on this hyperparameter.

During training, the loss function, learning rate, and optimizer can affect the evolution of the training. The learning rate must be sufficient to improve the training accuracy after each epoch but not too important to obtain precise learning. The loss function depends on the application context of DL, for classification, the cross-entropy function is adapted.

Concerning the network architecture the activation function and the dropout can significantly impact the learning. The activation function can be linear, rectifier linear unit (ReLU), Logistic, or Gaussian, and allows to highlighting of the value of a node. The dropout is a regularization technique used in DL models, particularly in neural networks, to prevent overfitting and improve generalization performance. The dropout is a value contained in $[0; 1]$. Depending on this value, a subset of neurons are randomly selected and temporarily removed, along with all of their incoming and outgoing connections, during the forward and backward passes.

Finally to prevent overfitting, a common method is to create new data thanks to the dataset by adding different levels and types of noise in the dataset. This method is called data augmentation.

2.6.3 Deep learning for RFF identification

Figure 2.18 presents the processing flow of RFF identification based on DL. Firstly a database has been created by recording the signals from different transmitters thanks to a SDR. The database is composed of different sequences of IQ samples and each sequence must correspond to a label to identify its emitter. Then the database is separated to form the training set and test set. The training set is pre-processed, with data augmentation post-acquisition and/or channel equalization, and most of the time the sequences are shaped with normalization, slicing, sliding, etc. Data augmentation is the process of increasing the size of the database. This technique can be performed during data acquisition by recording many signals from each transmitter. However, if this is not sufficient, it is possible to perform a virtual augmentation after the acquisition by adding some data

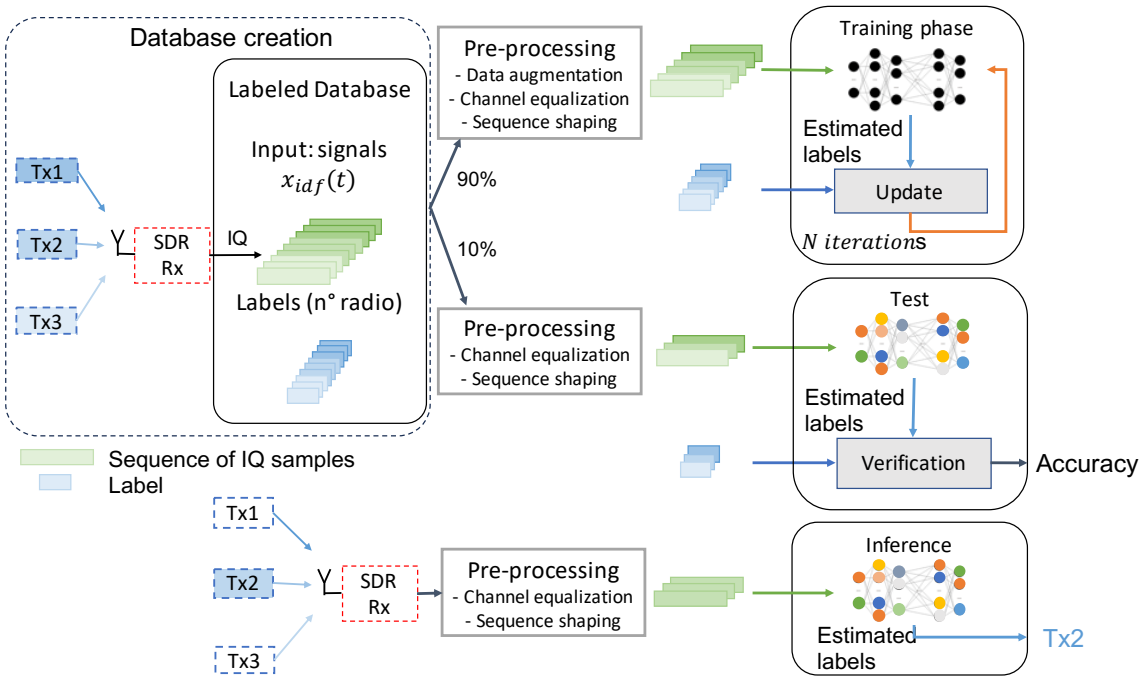


Figure 2.18 – Processing chain with SoA DL techniques.

from the acquisition, with some modifications, to the dataset using different techniques. The test set is pre-processed in the same manner as the training set, excepting the data augmentation. The testing set allows the detection of overfitting which occurs when a ML model learns the training dataset too well, capturing noise or random fluctuations in the data rather than the underlying pattern. Finally, in the inference step, the signals are captured and pre-processed and the network identifies the emitter by estimation.

This section gives a preview of the SoA of RFF identification-based DL which takes in input IQ sample sequence with and without pre-processing.

A. IQ samples based DL

The DL was introduced for modulation recognition in 2016 by O'Shera et al. [68], then inspired by this paper, Riaz et al. [84] propose in 2018 to use a DL architecture to performed RFF identification. They introduced a CNN with two convolutional layers composed of 50 filters of size 1×3 for the first layer and 2×3 for the second. The network takes in input a vector of l I/Q samples for a sequence of length l, here $l = 128$. The CNN is compared with conventional methods such as SVM with several features such as amplitude, phase, Fast Fourier Transform (FFT) value, etc. The CNN achieves promising results with 98% of accuracy compared to conventional methods with 55% of accuracy.

In the same year, [63] propose to perform RFF identification thanks to CNN, they create a dataset where each transmission between transmitter and receiver is included in the dataset several times with varying levels of simulated AWGN.

In 2019, Sankhe et al. [92] propose an approach to detect a unique transmitter from a large group of bit-similar devices, which means the same hardware, same MAC ID, and same protocol, using only IQ samples at the physical layer. They use a CNN composed of 2 convolutional layers and three fully connected layers, to identify 16 devices. The devices are Universal Software Radio Peripheral (USRP) X310 that are high quality. Sankhe et al. analyse two situations: static channel environnement and dynamic channel. In the first situation, they present good results with CNN architecture taking raw IQ samples in input without any channel estimation or particular pre-processing corresponding to the communication protocol of the targeted device. In dynamic situations, they add some controlled impairments to strengthen the RFF and improve the classification.

Using DL for RFF identification seems interesting. However, the signal can contain other relevant signatures such as a MAC-ID which is a more significant signature compared to hardware imperfections. In this case, the network will learn the MAC address and, in a spoofing context, the system is not protected. To overcome this issue, Jian et al. [48] demonstrate that the slicing technique introduced by Riaz et al. [84] produce a MAC-learning resistant classification, permitting a MAC spoofing-resistant RFF identification. They experiment on a deep CNN composed of five stacks of 2 convolutional layers one max-pooling layer and three fully connected layers at the end of the network. Another potential relevant signature is the propagation channel environment, since 2019 several authors have highlighted this issue and proposed several solutions to avoid it that are presented in the next subsection.

B. DL on equalized data

Sankhe et al. [92] propose to also study a dynamic propagation channel situation. They performed three experiments to show the channel impact. First, the network is trained with one location and tested on the same location and achieves near-perfect results, then the training is done on several locations and tested on the same locations, achieving good results but confusing some transmitters. Finally, the network is trained with data from one location and tested with data from another location, the classification is unpredictable. To overcome this channel issue, they propose to introduce controlled imperfections at the transmitter side thanks to feedback modifications that use channel estimation at the

receiver. This solution is interesting in securing the wireless network but not in defense applications because the transmitter is not accessible to introduce controlled imperfections. In 2020, Sankhe et al. [91] propose an extended version of the paper [92]. In this version, they propose an undercomplete demodulation to remove the effect of the channel and conserve the RFF imperfections. The channel is estimated thanks to the pilot training sequence. They present more precisely their contribution to add controlled impairments in particular for scalability and communication impact. The results give very good accuracy for wire transmission training and wireless tests in two locations, indoor and outdoor.

In the same idea, Restuccia et al. [82] propose a system for real-time channel resilient and adversary resilient optimization based on DL classification. The innovation lies in the use of a carefully optimized digital finite input response filter at the transmitter end. This adjustment of the filter strengthens the device fingerprint according to the current channel conditions and creates a more relevant signature for the network.

The DL technique has also been presented as a solution for scalable applications with many devices. In 2020 Jian et al. [47] present a massive experimental study by using a dataset from DARPA within WiFi and ADS-B signals from 10,000 transmitters captured in the wild. They performed two processing steps on WiFi signals, band filtering, and partial equalization, and used only raw IQ samples for ADS-B signals. As in previous papers, the slicing method is used, combined with sliding windows. In this study, the authors are interested in scalability, the size of the training set, the channel effect, the SNR, and the data transmission. The results, obtained with two different CNNs, show the difficulty of classifying a large WiFi population (1000) compared to a small one (100) with the same number of transmissions per device. Moreover, they propose to study the multi-burst during inference to classify a signal thanks to different sequences of this signal. The multi-burst is a great solution to improve accuracy per transmission, particularly in a large population context.

On the same idea, Al-Shawabka et al. [94] propose to analyze the impact of wireless channels. They create an important dataset to evaluate the impact of the channel on CNN-based RFF identification. Three different CNNs architectures are proposed and tested, and different pre-processing are tested such as FFT and WiFi equalization to avoid channel disturbance. The results conclude that the wireless channel impact negatively the classification accuracy. However, IQ data equalization improves accuracy by up to 23% compared to no equalization.

Finally, Shen et al. [97, 96] suggest pre-processing the data to obtain a Channel Independant Spectrogramm (CIS) and avoid channel problems.

C. Channel mitigation by data augmentation

While channel equalization seems to overcome channel learning, the SoA suggest using data augmentation to avoid channel issues. For example, in 2020 Soltani et al. [102] present two data augmentation solutions. The first one consists of physical data augmentation by adding different channels between the transmitters and the receiver. This solution has been simulated thanks to the channel model and additive Gaussian noise which simulated the noise of the receiver. The second data augmentation is done on received data from the DARPA dataset: after the transmission, they apply different channel models to the data. The results show better performance for the first solution. However, the data augmentation at the receiver side improves the results from 60% to 80%. In the same idea, Shen et al. [96, 97], propose to compare two types of data augmentation called, Offline and Online, where Offline corresponds to one unique augmentation during database creation while with online augmentation, new data is created at each iteration of the algorithm. The performances obtained thanks to online augmentation are better because the network is trained with a bigger dataset, this technique requires more computational resources, but less memory resources compared to offline augmentation.

D. Channel mitigation by data augmentation during aquisition

A real physical data augmentation is proposed by Morin et al. [66, 65], they randomize the position of the transmitter during the training dataset acquisition, in other words, the network could not attribute a particular channel environment to a transmitter because the position changed. They used the FIT/CorteXlab testbed with several nodes that allowed them to test physical layer identification techniques. This data augmentation is probably the most interesting compared to the post-acquisition one but requires time and many experiments to create the dataset. Nevertheless, the capturing room is an anechoïc chamber with spatial regularity.

Hanna et al. [40] propose a study on a large-scale WiFi dataset called WiSig, the study concludes with the difficulties of the network to generalize the learning. For example, they perform the training with data from a receiver and test with data from another receiver, and the classification accuracy is significantly degraded compared to testing with the

data from the same receiver. To overcome this issue they propose data augmentation by increasing the number of receivers during the training.

E. Other techniques to improve performance

In 2021, Zhang et al. [121] designed a robust RFF identification protocol thanks to the comprehensive study of RF impairments modeling in a wireless transmission context. The model includes oscillator imperfections, IQ gain and phase imbalance, and PA non-linearity. An experimental measure of the CFO over three months reveals an important variation that is not suitable for RFF classification as it interferes with other imperfections. They advise estimating and compensating the CFO to avoid RFF identification depending on temperature changes. Without this compensation, the CNN focuses on the most relevant feature which is the CFO.

While some authors focus the classification research on CNN, some others propose different architectures such as RiftNet [85], which is composed of a Dilated Causal Convolution (DCC) layer to extract features from the preamble, and from the other data. Moreover, this paper proposes to use the multi-burst decision to improve the classification acuracy [97]. Recently, the performance of transformer architecture in DL application aroused the interest of researchers, in several works. Shen et al. [97, 96, 98] propose to classify the signals with a transformers architecture. Initially, designed for sequence translation model, the transformers are composed of multi-head attention layers and allow to process of variable-length sequential data during inference. This characteristic has been studied and compared to other networks by Shen et al. [97]. They propose four neural networks that can process signals of variable lengths, namely flatten-free CNN, LSTM network, GRU network, and a transformer. The flatten-free layer consists of replacing the flattening layer with a global average pooling 2D layer, to fix the length of the dense layers input. The LSTM is a variant of RNN that are also study for RFF identification [102, 88, 93, 95]. The GRU is another variant of RNN, simpler as LSTM.

If the DL, in particular, CNN, achieves promising results, the number of parameters and the computation of the model are ignored by the authors. However, several applications presented in the previous section of this chapter require an embedded recognition system to be deployed. To address this issue [27] proposes a lightweight CNN which conserves high recognition accuracy.

2.6.4 Conclusion

In conclusion, DL methods contributions are summarized in Table 2.4, they are mainly based on IQ sample sequence pattern recognition. However, the RFF imperfections patterns are not easy to detect and some pre-processing is introduced in the SoA to improve the classification accuracy. Some pre-processing, such as channel equalization, requires knowledge of the target signals to estimate the channel. Identification based on DL seems to be more adaptable to our application context compared to parametric methods. However, a comprehensive study is needed to improve the classification accuracy with less knowledge about the target signals and a large and robust database is required to ensure RFF recognition.

2.7 Hybrid

While the previous section has presented DL solutions based on IQ samples, this section presents the hybrid RFF identification methods that combines the parametric features extractions and DL. The idea is to help the network to focus on the impairments and this pre-processing can avoid channel learning as [98, 97] where Shen et al. implemented a channel-independent spectrogram for LoRa preamble to obtain robust channel identification. The hybrid methods commonly required some knowledge about the signals to extract some RF impairments:

IQ imbalance: In their previous work Peng et al. [73] proposed to use DCTF as a feature combining with frequency, and IQ offset and a own classifier to perform the RFF identification. In 2020, [72] inspired by other work on DL they propose to use a CNN to classify the DCTF. The DCTF highlights in particular the IQ imbalance.

Power amplifier: Another paper [60] proposes to use a density trace figure to highlight the non-linear PA memory effect. Then a CNN is used to classify the density trace figures and so RFF identification based on PA impairments.

CFO: A recent paper proposes another IQ data representation which is called Double side Envelope Power Spectrum [24], this IQ representation highlights the oscillator behavior. They test the classification across different situations, by changing the channel, the location, and the time. The results are interesting however a study of the impact of the temperature is missing.

Conclusion The hybrid methods are interesting for RFF identification as they offer the computational complexity of DL combined with the extraction features, by decreasing

the potential number of parameters. Feature extraction can help the network to focus on RFF classification and not use other information in the signal to classify them.

2.8 Conclusions

This chapter presents a large overview of RFF identification. This identification based on hardware singularities is difficult because of numerous challenges linked to the extraction of features by parametric-based methods or thanks to DL layers [4]. The SoA presents parametric-based methods that use specific feature extraction and an ML classifier to perform the identification. Several extractions are proposed and combined to obtain characteristic vector features. The SoA presents different types of neural network structures, that are already used for computer vision or natural language processing, such as CNN structure or LSTM and transformers. However, these models are designed to capture features in specific domains, often related to image processing, and they are probably not the most performant network for RFF issues. In addition, training is defined by numerous hyperparameters that configure the training evolution, such as learning rate, batch size, dropout, and regularization parameters, making it difficult to fine-tune the network. Finally, the length of the input can impact the classification, in many works, the authors propose to slice the signals [91, 102, 40]. This allows to use fixed-size input structure network contrary to Shen [97].

While the classification accuracy in DL depends on the training step, it also depends on the data because the training and classification are impacted by the complexity and several issues of wireless transmission such as signal interference, propagation channel, the level of noise, the protocol of communication, etc. Therefore, there is a strong need for a large and robust database [46]. The next chapter presents the SoA of the databases proposed and used by the community for RFF identification.

Year	Author	Network	Signal Processing	Data Processing	Contributions
2016	O'Shera [68]	CNN	-	Slicing	Modulation recognition
2018	Riyaz [84]	CNN	-	Slicing	RFF identification
2019	Jian [48]	CNN	-	Slicing	Slicing for MAC-spoofing resilient
2019 2020	Sankhe [92, 91]	CNN	Equalization	Sliding	Add controlled imperfections to maximize the accuracy
2019	Restuccia [82]	CNN	-	-	Leverages FIR to maximize the accuracy under dynamic channel
2019	Roy [88]	CNN, FNN RNN, GAN	-	Normalisation	Used GAN to recognize trusted Tx
2019 2020	Morin [66] [65]	CNN	-	-	Increase channel variations to be channel changed resilient
2020	Jian [47]	CNN	Equalization	Sliding	Massive experimental study
2020	Al-Shawabka [94]	CNN	Equalization	Sliding	Experimental study of channel impact
2020	Soltani [102]	RNN CNN	-	Augmentation, Normalisation, Sliding	Explore data augmentation on Tx side (simulation) and Rx side (experiments)
2021	Shen [98]	CNN	CIS	Augmentation Normalisation CFO compensation	CIS Data augmentation
2021	Shen [96]	Transformers	Synchronisation CIS	Augmentation Normalisation CFO compensation	Multi packet interference process signals of variable length
2021	Zhang [121]	CNN	Synchronisation	CFO compensation	Tx, Rx Modelisation
2020 2021	Robinson [86, 85]	CNN + DCC	Filtered	Normalisation	Propose new NN
2021	Al-Shawabka [93]	CNN RNN-LSTM	Filtered	Augmentation	Evaluation of DL Data augmentation
2022	Hanna [40]	CNN	Equalization	Normalisation Slicing	Propose a large data base
2022	Yang [116]	3 CNNs	-	Normalisation	Voting scheme with I, Q and I + Q channel data
2023	Shen [97]	LSTM, GRU, Transformers	CIS	Augmentation	Process signals of variable lengths
2023	Feng [27]	Light CNN	Filtered	Normalisation Slicing	Propose a light CNN, compared to SVM and classic CNN

Table 2.4 – Summary table of DL-based work for RFF identification

Chapter 3

Database Challenges for RFF Identification

The previous chapter presented the SoA of RFF identification based on ML solutions, in particular on supervised DL. Since network training and performance evaluation are possible thanks to labeled databases, this chapter introduces the databases of the SoA and the challenges related to data acquisition. This chapter presents in Section 3.1 the SoA of databases used for RFF identification, particularly the databases used to train and test DL techniques. Then, Section 3.2 intro ces the challenges related to databases. Then in Section 3.3 two open databases are studied to reveal the current challenges. Finally, Section 3.4 presents the need for a tool to explore the RFFs and their impact on DL identification capacity.

3.1 State of the Art of RFF Databases

As it was presented, DL solutions achieve good results, but the classification accuracy of such methods dramatically depends on the database used to train the network. This phenomenon is well known in image classification, the training dataset has to represent correctly all situations that the network has to classify to perform correctly. For instance, with dog classification, the training dataset requires many types of dogs in different positions and views to recognize any dog images. Therefore, DL needs a large, diverse, and robust database to recognize certain features that allow classification. For RFF identification there is a strong need for large and robust databases, composed of raw labeled signals [46] from different transmitters to ensure RFF recognition in many environmental conditions.

Since 2019, the SoA on RFF identification with DL has increased and different databases to experiment RFF identification have been proposed in the literature. A selection of re-

cent papers on RFF databases is presented in Table 3.1. These databases are separated into two types: experimental-based and simulation-based. They are created with different wireless protocols, presented in the fifth column in Table 3.1 such as WiFi [40, 91, 92] and LoRa [96, 106]. Column 4 "Is data public?" gives information on public accessibility and column 6 gives information about the number of emitters where Device Under Test (DUT) are commercial off-the-shelf devices. Finally, column 7 gives additional information or contributions of the paper. The next subsections present different families of databases used in the SoA of RFF identification, particularly the database with real signals called experimental databases and databases with simulated signals called simulated databases. First, the real databases are introduced with private and public databases, and finally, we focus on simulated databases.

3.1.1 Experimental Databases

A. Private

The largest existing database for RFF identification was created by Defense Advanced Research Projects Agency (DARPA) in 2020. This database, used by authors from Northeastern University in Boston, is a private one and is used in many papers [47, 102, 91, 94, 48]. 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 [102]. 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. [73, 72] 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 by them for different works.

Many papers in the literature are based on data created for the study of the paper with few devices [84, 96, 36, 83] and the authors never give open access to their database. Consequently, the reproducibility of experimental results is not possible and makes it difficult to understand and explore the identification scenario proposed by the authors, in particular, if the experimental setup is not precisely described by metadata the reader can only try to reproduce similar results with its own database.

B. Public

The University of Boston created their own databases for experiments in papers [84, 91, 92]. First, they created a database with 5 USRPs B210 transmitters with different distances varying between 2 and 50 ft [84]. Then in 2019, they created the ORACLE database with 20 USRPs X310 transmitters [92] that emitted WiFi signal. They suggested introducing software-controlled impairments at the transmitter side to enhance identification robustness. This recommendation arises because the X310 transmitters are produced with low variability components, resulting in minimal RF front-end variations between the two devices.

In 2022, Hanna et al. proposed a new public database for RFF [40] called WiSig. WiSig is constructed with many signals and with information on how signals have been captured, such 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, probably not so much compared to other databases. They present WiSig as a RFF database to explore the identification in a static environment with different types of transmitters and different numbers of transmitters/receivers/signals/days.

In the same way, Al-Shawabka et al. present in [94] 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. [66] work on an 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 [45] a new public dataset that includes emissions from 10 Commercial Orbital Transportation Services (COTS) IoT emitters (2 laptops and 8

Reference	Year	Database	Is data public?	Protocol	Numbers of emitters	Additional informations or Contribution
Experimental Databases						
Hall [36]	2003	Own	No	Bluetooth	10 Txs	Exploit the phase to detect transient part
Riyaz [84]	2018	Own	No	WiFi	5 B210	
Sankhe [91, 92]	2019	ORACLE	Yes	WiFi	16 X310	Add control impairments with feed-back driven to increase differentiability
		DARPA	No	WiFi	140 DUTs	
Morin [66]	2019	FIT/ CorteXlab	Yes	WiFi	21 N2932	Physical data augmentation, to minimise the impact of the propagation channel
Peng [73]	2019	Own	No	ZigBee	54 DUTs	Adopt 4 novels modulation-based features effective in ZigBee node classification
Jian [47]	2020	DARPA	No	WiFi ADS-B	5000 DUTs 5000 DUTs	Investigate 2 CNNs for RFF identification under different environmental scenarios
Soltani [102]	2020	DARPA	No	WiFi	50 to 5000 DUTs	Study the interest of data augmentation
Al-Shawabka [94]	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 [96]	2021	Own	No	LoRa	10 DUTs	Improve low SNR classification accuracy with data augmentation
Elmaghbub [25]	2021	Own	Yes	LoRa	25 Pycom devices	Study the sensitivity to deployement variability
Reus-Muns [83]	2020	POWDER	No	WiFi, 4G, 5G	4 USRP X310	Incorporate the triplet loss with the deep CNN
Hanna [40]	2022	WiSig	Yes	WiFi	174 USRPs	Create 4 datasets for RFF identification varying days, N_{Tx}, receivers and signals
Jagannath [45]	2022	Own	Yes	Bluetooth	10 DUTs	2 days to do generalization.
Chillet [13]	2023	WiSig	Yes	WiFi	6 USRPs	Use TPG as a classifier
Elmaghbub [26]	2023	Own	Yes	WiFi	50 Pycom devices	2 datasets outdoor and indoor
Elmaghbub [24]	2023	Own	Yes	WiFi	50 Pycom devices	4 datasets: wire, wireless, different locations
Simulation based Databases						
Soltani [102]	2020	Own	Yes	WiFi	10 Txs	Model only IQ imbalance impairment
Zhang [121]	2021	Own	No	LoRa	50 to 200 Tx	Uniformly and randomly distributed IQ imbalances and PA nonlinearities
Chillet	2024	RiFyFi_VDG	Yes	WiFi	N_{Tx}	Virtual database generator IQ imbalance, PA, PN, CFO

Table 3.1 – Summary table of databases for RFF identification

commercial chips) that are captured with a USRP X300 device. The data set is split into two: Day1 and Day2, each recorded in a different time frame, location, and testbed setup to allow for critical generalization testing of the trained DL model.

Elmaghbub et al. propose different WiFi datasets [26],[24] composed of 50 Pycom devices. They create outdoor and indoor scenarios, 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 sig-

nals 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.

3.1.2 Simulation based Database

Soltani et al. [102] 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 [121], 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 reconfigured 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.

3.2 RFF Databases Challenges

The design space of the RFF database has been largely explored by the community. However, the data collection process poses several challenges, particularly related to design choices and pre-processing techniques. The design choices are related to the devices and the signal, such as protocols and environmental conditions during the recordings. The pre-processing techniques include data augmentation, shaping and normalization, and compensation techniques.

3.2.1 Design choices

A. Devices

In the literature, experimental signals are mostly generated using DUTs or SDR platforms for both transmitters and receivers. The work of Zhang et al. [122] shows that the

type of transmitter and receiver is important in RFF identification because the ability to discriminate two devices is related to the RFF difference between the two devices. For example, a USRP X310, a high-quality device, is manufactured with low variability components, resulting in minimal RF front-end variation between two devices. Sankhe et al. [91] show that two X310 are more difficult to separate than two B210. Furthermore, [80] studies the effect of the receiver on the classification capacity, a receiver could be sensitive to an emitter. Therefore, the similarity between devices affects the classification accuracy. For commercial devices such as smartphones or laptops, the quality of the embedded electric circuit can vary between constructors or device references. The quality of the electric circuit impacts the diversity of impairments around a mean value. This means that it is more difficult to identify two high-quality devices from the same constructor and the same references than two different devices. The massive experimental study conducted by Jian et al. [47] shows that the number of devices affects classification accuracy. This is probably due to the increased probability of having two devices with similar fingerprints.

B. Signals

The protocol chosen can affect classification accuracy due to the signal modulation used. For example, LoRa technology uses spread spectrum modulation techniques, whereas OFDM works by dividing the available spectrum into multiple orthogonal (non-interfering) subcarriers, each carrying its narrowband signal. This means that the effect of RFF does not apply to the same type of signal. In addition, the environmental conditions during signal acquisition can affect wireless transmission. Therefore, some identification applications use propagation conditions to identify the emitters such as the Received Signal Strength (RSS) and the Channel State Information (CSI) [123, 21]. However, these applications assume the static position of transmitters and receivers, which is not a necessary assumption for RFF identification.

In RFF identification, the environmental conditions can create a bias in the database, which can be considered by the classification stage as a principal discriminant characteristic of devices. For example, signals recorded with one location or during only one particular day may not be representative of all situations, which causes generalization problems and the inability of the network to predict the emitted device in other situations, other days for example. Therefore, the amount and diversity of data is also important to ensure the generalization of the training network and good accuracy of the inference

3.2.2 Pre-processing Techniques

A. Input data selection and shaping

The nature of the signals and the frame structure protocol affect the analysis of the results derived by the authors, making it difficult to compare different works. For example, Shen et al. [96] use only the preamble for WiFi data, while in ORACLE [92] the frame consists of a MAC address field with always the same address and a random payload. Jian et al. [48] proposes to slice the signals containing the MAC address to be resilient to MAC address spoofing. Alhazbi et al. [4] explain that selecting the most appropriate data segment to input into the identification network is a significant challenge. Ideally, the chosen segment should have consistent and repetitive patterns to ensure that the DL model learns from the unique characteristics of the RF signal rather than being influenced by the specific content of the wireless segment. The preamble of the radio packet, which contains synchronization-related information, is particularly attractive due to its consistency across different devices and packets within the same communication technology. However, these relatively short signal sequences provide limited data to learn the RFF. Conversely, using the payload of wireless packets, which is typically longer, presents challenges due to data scrambling and content variability which makes it difficult to correlate data. In the pre-processing techniques, the authors from the SoA use different sequence sizes, such as 128 IQ samples for [91] and 256 IQ samples for [40], obtained by slicing or sliding the signal. This sequence size is important because it must allow for RFF observability to enable RFF classification. The sequence is particularly short, for example at 20MHz it corresponds to $12\mu s$. In addition, to overcome overfitting on the amplitude difference of the signals, the authors from the SoA use dataset normalization, which is common for DL learning database.

B. Bias compensation

Wireless data is inherently dynamic and subject to time-varying channel conditions, hardware imperfections, and noise. These factors can lead to performance variations and scaling issues in the received data. Since DL models are sensitive to scale, these inconsistencies, if not addressed, can adversely affect the models learning and ability to generalize. In particular, a difference in amplitude can create a bias that can cause the quick convergence of the model to a solution that is primarily based on the magnitude of device identification. By normalizing all feature ranges of wireless data, DL models can accu-

rately capture the underlying data distribution pattern and mitigate bias towards features with large scales. This normalization approach accelerates model convergence, stabilizes neural networks, enhances model generalization capabilities, and ultimately improves DL model performance for identification tasks. To avoid some classification problems such as channel or receiver effects, some authors propose to pre-process the data before using the neural network, [91] proposing an under-complete demodulation that aims to remove only the channel effect from the raw IQ samples without compensating for the device imperfections. In the same idea, [47, 94] propose to perform channel equalization, and [98] propose a CIS representation.

In conventional wireless communications, the receiver performs other compensations such as IQ imbalance and CFO compensation to improve the bit error rate. In RFF identification, the aim is to preserve these impairments to perform identification. However, the CFO is not a stable impairment because the frequency error changes as a function of temperature. Shen et al. [96] and Zhang et al. [121] propose CFO compensation to overcome this problem and increase identification stability.

Compensating for the bias requires knowledge of the signals and the transmit/receive chain, which is not always possible.

3.2.3 Conclusion

In classification problems, the database used to determine how to separate the different classes is important. The construction of this database and the different design choices will impact the capacity to separate the transmitters. Moreover, using DL for classification increases the importance of the database, since the network is based only on the lessons it makes thanks to this database. For RFF identification, many parameters will affect the classification, and having information about this database will help to draw conclusions about network performance. The next section presents a primary study of two different databases.

3.3 Primary study of SoA databases for RFF identification with DL

This section proposes a preliminary study based on two different databases of the SoA. This study aims to show the limits of the actual databases to initiate a reflection

on needs and different degrees of flexibility. Figure 3.1 shows the different parts of the experiment. The database is composed of signals and labels for different scenarios (different acquisitions), the goal is to train the network with a particular scenario and evaluate the classification ability on all scenarios. The networks used to classify the signals are presented then two experiments are proposed.

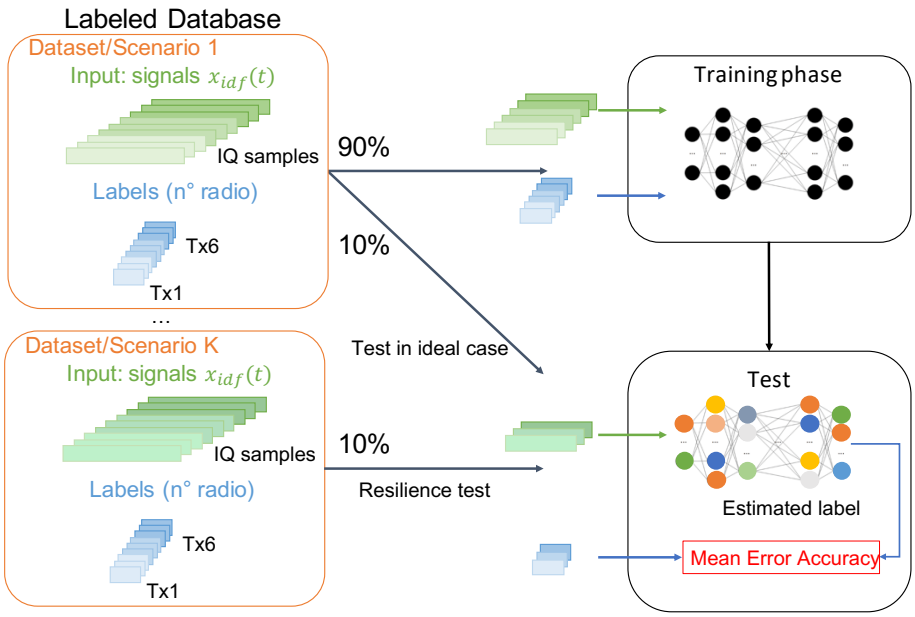


Figure 3.1 – Overview of the training and testing network with signals.

3.3.1 Networks: Presentation and evaluation

For this study, three CNNs are used to evaluate the database relevance to train different networks. 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 $t_p + t_n$ out of the total number of classifications. For the class c the accuracy can be expressed as:

$$Accuracy(c) = \frac{t_p(c) + t_n(c)}{t_p(c) + t_n(c) + f_p(c) + f_n(c)} \tag{3.1}$$

where $t_p(c)$ stands for the number of true positives for class c, $f_p(c)$ stands for false positives, $t_p(c)$ for true positive and $f_n(c)$ the number of false negatives. These terms

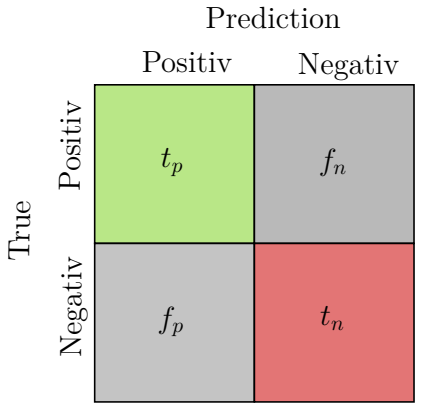
are presented in Figure 3.2 for a classification of 3 transmitters and considering the class number one.

— **True Positive** (or t_p), these are the elements of the class studied which have been correctly predicted.

— **True Negative** (or t_n), these are elements not belonging to the class studied which are not predicted as belonging to the class studied.

— **False Positive** (or f_p), these are items not belonging to the class studied but predicted as belonging to this class.

— **False Negative** (or f_n), these are the elements of the class studied predicted as not belonging to this class.



Guess / True	Tx_1	Tx_2	Tx_3
Tx_1	60	40	0
Tx_2	0	100	0
Tx_3	0	15	85

Figure 3.2 – Confusion Matrix and analyze class corresponds to transmitter 1.

The F1 score is calculated on the batch sequences as follows:

$$\mathrm{F1} = \mathop{\mathbb{E}}_{c \in \mathcal{C}} \left(\frac{2}{\frac{1}{P(c)} + \frac{1}{R(c)}} \right), \tag{3.2}$$

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

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 is the percentage of correct predictions for a particular class out of the total number of predictions made for that class. $R(c)$ is the recall for the

class c, which is the measure of the ability of a classification model to identify all positive occurrences of a particular class, representing the percentage of correct predictions for that class relative to the total number of actual occurrences of that class. The F1 score is a tool for measuring the overall performance of a classification model, combining both precision and recall. It takes into account true positives, false positives, and false negatives. More specifically, the F1 score measures the ability of the model to make accurate predictions for all classes, avoiding misclassifications and identifying all true occurrences of each class. The F1 score is interesting when the dataset is not balanced. In our case, the F1 score is very close to the accuracy value because the dataset is completely balanced. In this PhD, the F1 score is expressed between 0 and 1 or in percentage in the confusion matrix to improve the readability.

The three networks chosen in this section have been already proposed and studied for RFF identification in the SoA [1]. The first one called Sankhe_2020 is a CNN inspired by the network proposed by Sankhe et al. [91], 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. After the 4 convolutional layers, the CNN has 3 fully connected layers with 256 nodes, 128, and the number of classes. After the two first fully connected layers, a dropout layer is added with dr $\in [0; 1]$. For the ending layer, a softmax layer is added. In input, the network takes complex-based band signals without pre-processing. These signals are split into two raws I and Q, and N corresponds to the input size. This architecture is presented in Figure 3.3, the notation $N_{\text{Filters}}CL(a, b)$ stands for the number of filters in the Convolutional Layer with (a,b) the size of the filters. For instance, 128 CL(7,1) is for a layer of 128 convolutional filters of size (7,1). The notation *MP 2* stands for Maxpooling 2. The green rectangles and expressions below represent the data and the format of data between each layer.

The second network called Sankhe_2019 [92], is composed of 2 convolutional layers, both layers are composed of 50 filters of size 7. After the convolutional layers, the CNN includes 3 fully connected layers with 256, 80, and the number of classes N_{Tx}. After the two first fully connected layers a dropout layer is added with dr $\in [0; 1]$. This architecture, shown in Figure 3.4, is composed of fewer filters but requires more parameters.

Finally, the third one was proposed by Gutierrez del Arroyo [35], called Arroyo_2022 in this PhD, is composed of three convolutional layers each followed by a max-pooling layer. The first convolutional layer is composed of 64 filters of size 10 the second layer has

1. The architectures presented here can be different from the structure used by the SoA authors if description information is missing in the referenced paper.

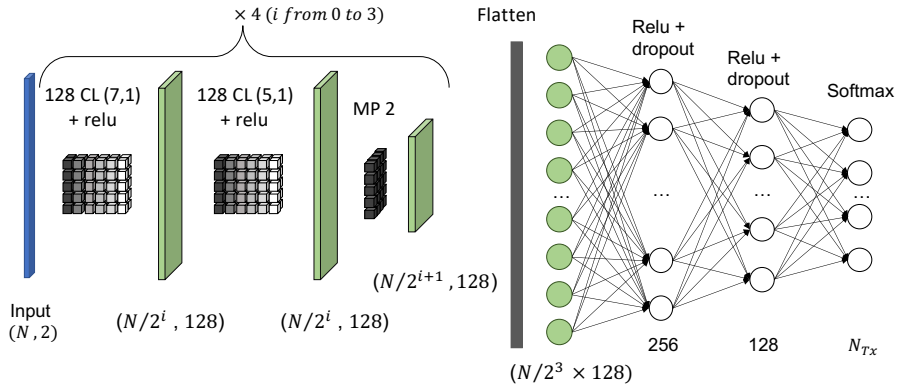


Figure 3.3 – Deep Learning architecture Sankhe_2020.

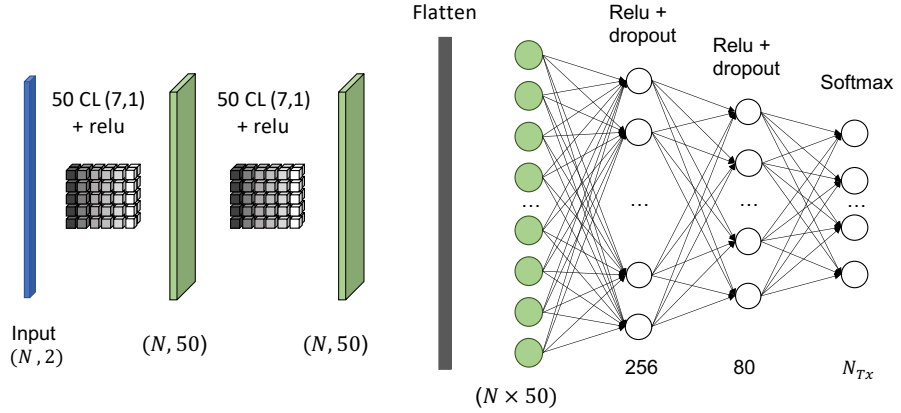


Figure 3.4 – Deep Learning architecture Sankhe_2019.

32 filters and the third one has 16 filters. This architecture is presented Figure 3.5, the architecture has fewer parameters than the previous ones.

For $N = 256$ as done in [40], the first architecture has 1,232,774 parameters, the second has 3,316,402 parameters and the third has 60,114 parameters. All the experiments have been realized with a dropout of 0.5 and a learning rate at 10^{-4} after an empirical exploration of the learning parameter which allows correct learning. The chosen activation function is ReLu for all the networks and the optimizer is Adam. The batch size has been empirically explored and set to 64.

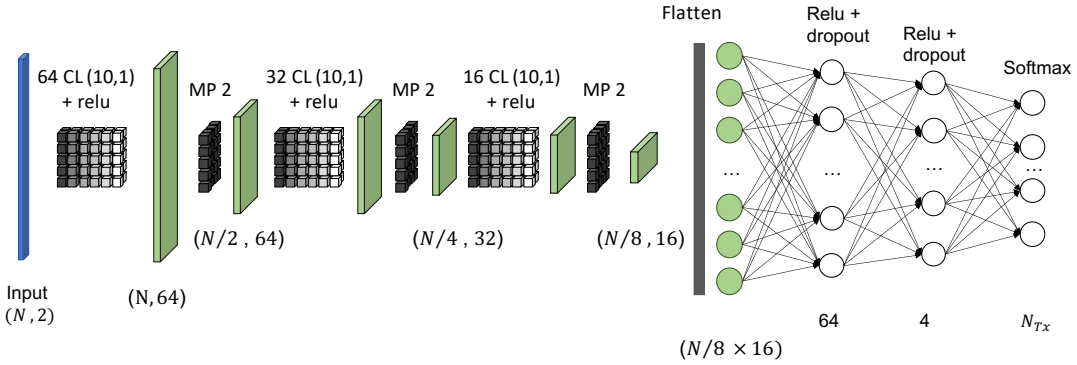


Figure 3.5 – Deep Learning architecture Arroyo_2022.

3.3.2 First use case: Study of the WiSig database

A. Database

WiSig is a recent database [40] that has been built with many signals and a lot of information about how the signals were captured, such as transmitter location and type of radio used. They provide a large WiFi dataset captured by 41 USRPs with 20 MHz bandwidth from different references. The signals come from 174 WiFi transmitters in four different captures over one month. The authors have created different databases with many transmitters (150), many receivers (32), and many signals (1000 for each transmitter). For our experiments, we chose the ManySig database with 6 transmitters and 12 receivers. We have represented the locations of the transmitters and receivers to study the influence of the channel, which can be observed in Figure 3.6. Each transmitter, shown in the blue square in Figure 3.6, has transmitted 1000 signals of 256 IQ samples. All the transmitters are Atheros AR5212 and AR5213. The receivers are placed in the room, so the propagation channel may differ from one radio to another. The database is split into two parts, 90% (5400 signals) for training and 10% (600 signals) for testing. Both sets of data are balanced as they contain signals from all transmitters with a balanced ratio.

B. Presenting the experiments

In this study, the receiver Rx1 is used and two different datasets are used: the equalized one and the non-equalized one. The aim is to analyze the results obtained with equalized and non-equalized data by training a network with data from certain day(s) and testing on

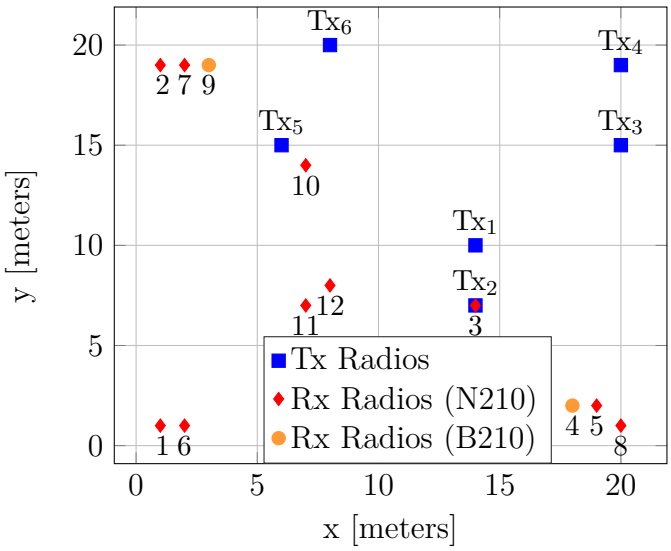


Figure 3.6 – Locations of Tx and Rx in the Orbit grid, for ManySig dataset.

other days. In their paper Hanna et al. [40] present an experimental study of the number of days used during training for equalized and unequalized data. The results are presented in Figure 3.8a and have been recreated by us in Figure 3.8b using the description of the CNN used by Hanna et al. [40], presented in Figure 3.7.

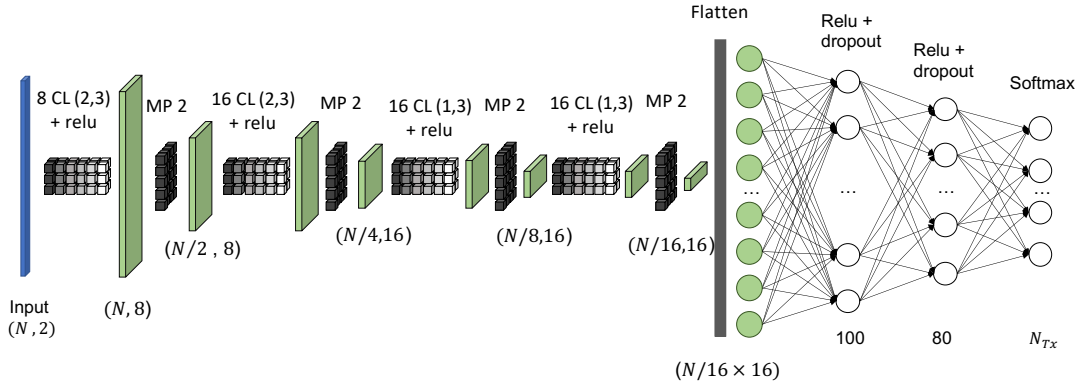
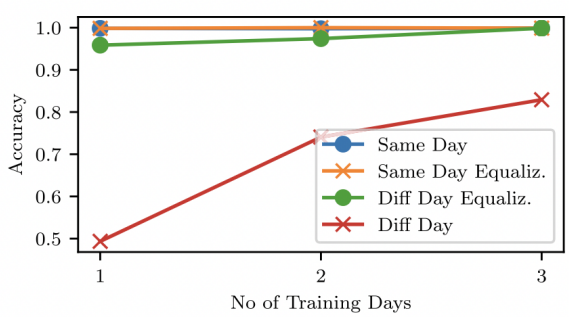
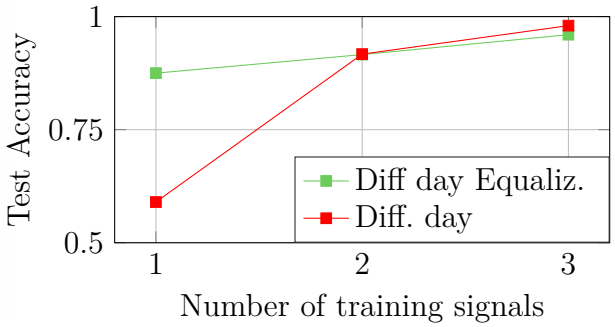


Figure 3.7 – Deep Learning architecture Hanna_2022.

This architecture is composed of 2 layers of 8 and 16 convolutional filters of size 2 by 3 each followed by a max-pooling layer. Then 2 layers of 16 convolutional filters of size 1 by 3 each followed by a max-pooling layer. Finally, three fully connected layers with ReLu and dropout are implemented with 100 neurons, 80 and the number of classes N_{Tx}.



(a) Results presented by Hanna et al [40].

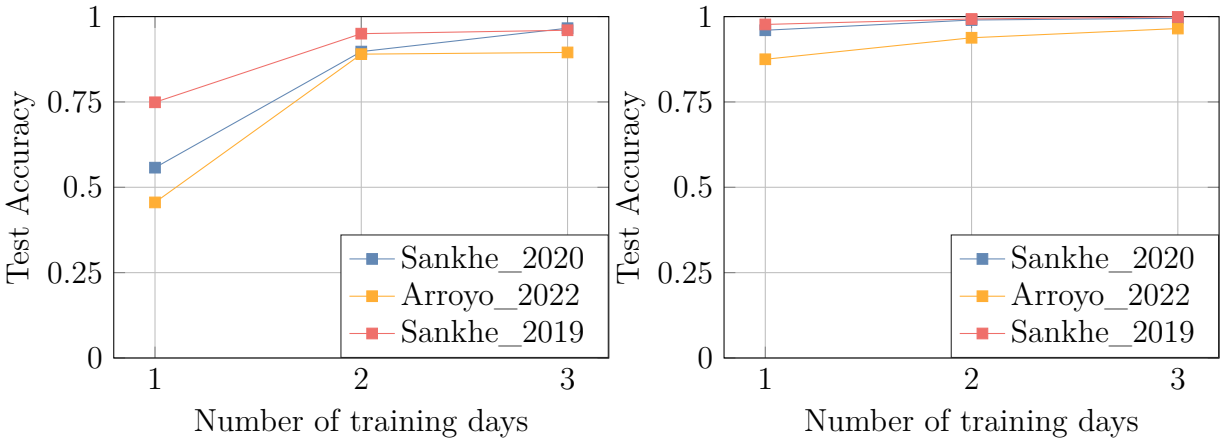


(b) Results obtained with the CNN from Hanna et al [40].

Figure 3.8 – Accuracy obtained in test (day 4) in the function of the number of training days, day 1, days 1 and 2 or days 1, 2 and 3.

The last day (4) is reserved for testing and the first three days are used for training, Figure 3.8a represents the accuracy obtained in the test while the network is trained with data from one day (the day 1), two days (the days 1 and 2) and three days (the days 1, 2 and 3). A difference between the two figures can be observed but the conclusions done by Hanna et al. are still valid about the importance of equalized data to obtain better performance on other days. The average results obtained with our three networks are then shown in Figure 3.9. We trained the three networks 5 times with the non-equalized datasets 3.9a and the equalized datasets 3.9b. First, the evolution of the curves shows that increasing the number of training days improves performance. Moreover, this figure seems to show the conclusion related to the importance of equalization, especially the difference in the one-day training set. The results obtained by Hanna et al. are equivalent and they conclude on the interest of the equalisation to improve the classification performance.

However, we propose to verify the results by comparing the accuracy obtained in the test on data from day 4 when the training is done with data from day 1, day 2, or day 3. Table 3.2 shows the accuracy obtained on the test set (day 4) while training with data from day 1, 2, or 3, for equalized and not equalized data. The first row represents the value of the first blue squares in Figure 3.9. The accuracy of rows 2 and 3 shows that it is not possible to conclude on the equalization interest in this context because depending on the training day, the performance on the test is very different. Indeed, the test accuracy obtained when the training is done with data from day 3 performs better with non-equalized data. Because of this conclusion, we propose to analyze more precisely the behavior of the networks on this database.



(a) No equalization.

(b) Equalization.

Figure 3.9 – Accuracy obtained in test (day 4) in the function of the number of training days, day 1, days 1 and 2 or days 1, 2 and 3.

Test / Train	No Equalization	Equalization
day_1	60%	90%
day_2	84%	80%
day_3	95%	78%

Table 3.2 – Accuracy obtained in test with data from day 4 for different training situations with Sankhe_2020.

C. Non-equalized data

The first experiment consists of training the network with the dataset of a particular day and testing the network with each day's dataset. Tables 3.3 shows the mean error classification accuracy obtained for the three different networks for each situation. The rows represent the training day and the columns represent the testing day. These experiments aim to show the ability of the network to perform classification when the situation is not the same. The possible changes between days 1, 2, 3, and 4 are the temperature, the pressure, and the presence of other interfering signals out of the capturing room, but the transmitters and receivers have not moved between recordings. Tables 3.3 show the percentage of error classification. For all tables, the diagonal is around 1 or 2%, which means that the network classifies perfectly all the signals of the test day n, while the

Train \ Test	day_1	day_2	day_3	day_4
day_1	1	60	50	50
day_2	62	0	17	25
day_3	45	7	0	5
day_4	38	18	6	0

(a) Sankhe_2020.

Train \ Test	day_1	day_2	day_3	day_4
day_1	1	19	14	21
day_2	36	0	2	5
day_3	34	1	1	2
day_4	24	3	1	0

(b) Sankhe_2019.

Train \ Test	day_1	day_2	day_3	day_4
day_1	2	43	24	31
day_2	55	1	9	16
day_3	44	6	1	7
day_4	36	14	6	1

(c) Arroyo_2022.

Table 3.3 – Mean error accuracy in percentage obtained for different days in test and train with no equalized data.

training is done on the training dataset from day n. However, for the three networks and in particular, for the Sankhe_2020 network, the classification of the signals of days 2, 3, and 4, while the training is done on day 1, is particularly bad. This is probably due to a change in environment during the recording. Excluding the first row and column of the matrices, the results obtained are interesting and show a low percentage of error. Sankhe_2019 architecture gives better results than the other two.

D. Equalized Data

The same experiment was carried out with equalized data to compare the results. Results are shown in Tables 3.4. Firstly, the difference between day 1 and the other days seems less pronounced. However, the performances obtained on days 2, 3, and 4 for the training days 2, 3, and 4 are on average worse with 11.8% than the performance obtained with non-equalized data on average 5.5%, regardless of the training or test day.

Train \ Test	day_1	day_2	day_3	day_4
day_1	1	22	29	2
day_2	29	0	24	17
day_3	43	22	0	24
day_4	30	7	13	0

(a) Sankhe_2020.

Train \ Test	day_1	day_2	day_3	day_4
day_1	1	22	26	1
day_2	22	0	16	17
day_3	44	18	0	12
day_4	29	3	6	0

(b) Sankhe_2019.

Train \ Test	day_1	day_2	day_3	day_4
day_1	2	30	40	20
day_2	39	1	24	26
day_3	41	29	1	28
day_4	27	10	17	1

(c) Arroyo_2022.

Table 3.4 – Mean error accuracy in percentage obtained for different days in test and train with equalized data.

E. Conclusion

The Wisig database offers the possibility to train a network and perform classification. However, the transmitters and receiver are always in the same location, reducing the opportunity to train the network in a dynamic context. In addition, this study shows that even in a static context, it is difficult to analyze the results because of external factors that cannot be controlled. These external factors can affect the recording data and make it difficult to recognize the RFF. Hanna et al. proposed to equalize data to improve classification and in particular the generalization to overcome the external factor issues. However, our study reveals that using equalization on data from days 2, 3, and 4 augments the error classification accuracy from an average of 5.5% without equalization to 11.8%. It is difficult to conclude that the network learns to recognize RFF and not the location of transmitters thanks to the propagation channel signature, and difficult to conclude about the real necessity of channel equalization. For example, it would be interesting to have a different signal acquisition with different transmitter positions to test the resilience of the network. In the next section, another database is used to evaluate the training possibilities and performances.

3.3.3 Second use case: Study of the Oracle database

A. Database

In the ORACLE database [92], all the transmitters consist of bit-similar USRP X310 radios transmitting frames conforming to the IEEE 802.11a standard, generated using the MATLAB WLAN System toolbox. The generated data frames contain random payloads but share identical address fields before being streamed to the selected SDR for over-the-air wireless transmission. The receiver SDR samples the incoming signals at a rate of 5 MS/s at the center frequency of 2.45 GHz. In total, they collect over 20 million samples for each transmitter. The experiments are carried out in an open area with fewer reflections, as shown in Figure 3.10. The separation between the transmitter and receiver is gradually increased from 2 ft to 62 ft, with intervals of 6 ft (1 ft = 30.48 cm). The dataset consists of recordings of collected raw IQ samples from 16 high-end X310 USRP SDRs with the same B210 SDR as the receiver. The recordings are organized into different folders with different transmitter/receiver separation distances in feet, with two different record moments called run 1 and run 2.

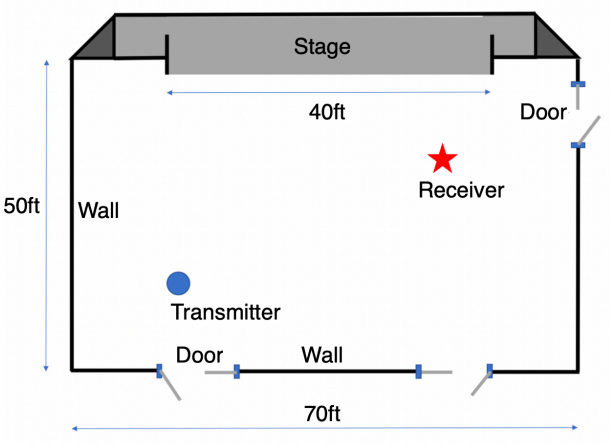
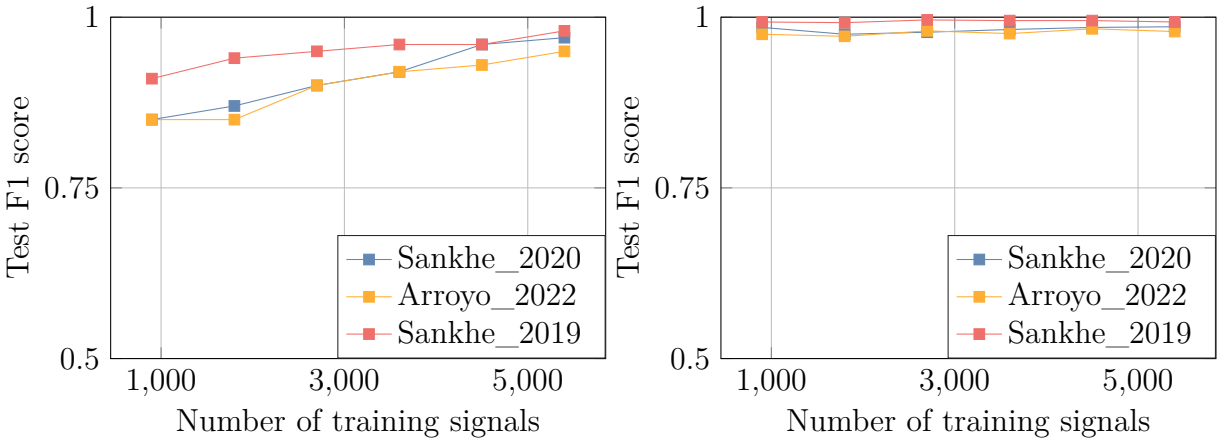


Figure 3.10 – Experimental environment in Oracle [91].

B. Presenting the experiments

To obtain consistent results compared to the WiSig database, we only use 6 transmitters. In the first experiment, only 2 ft and 62 ft data are used to train the network. The training is done with 900 signals to 5400 signals per transmitter, from run 1 and the test is performed on 100 signals per transmitter from the same recording set, run 1.

Figure 3.11 presents the average F1 score on the 3 networks obtained in the test as a function of the number of signals in the training set. The data from the 2 ft distance gives a worse performance in the test compared to the 62 ft distance. However, increasing the number of signals in training allows us to reach better performances in both cases for the three networks.



(a) 2 ft.

(b) 62 ft.

Figure 3.11 – Average F1 score obtained in test in function of the number of training signals for 2ft distance.

Then, we propose to use both sets of recordings, called run 1 and run 2 by Sankhe et al. [92]. The training is realized with data from run 1, with 900 signals per transmitter, and the test is realized with data from run 2. The training is realized among 5 seeds to obtain a mean percentage of correct classification accuracy. This experiment has been done for the three networks and different distances. Table 3.5 presents the mean percentage of accuracy obtained in the test with data from run 2. First, the three networks have the same behavior, particularly for extreme distances. The three networks are not able to classify the transmitter in the 2 ft situation, and the more the distance increases, the more the networks can correctly separate the transmitters. To understand this behavior, the experimental recording condition has to be analyzed, but we miss some important information such as the position of the transmitters during the transmission. The distance information does not allow us to know how the transmitters are positioned in the room. Figure 3.12 presents a scenario where all transmitters are in the same position for different distances and Figure 3.13 presents a second scenario where the transmitters are in different positions. With the second assumption, the propagation channel between

the transmitters and the receiver is different at 62 ft than 2 ft which can be a relevant difference between transmitters and allows the network to differentiate the transmitters thanks to the propagation channel.

	2ft	**20ft**	**38ft**	**56ft**	**62ft**
Sankhe_2020	13%	63%	41%	75%	75%
Arroyo_2022	15%	16%	52%	59%	80%
Sankhe_2019	14%	15%	42%	68%	80%

Table 3.5 – Test accuracy obtained with Run 2 data using a network trained with Run 1 data, with different train/test distances.

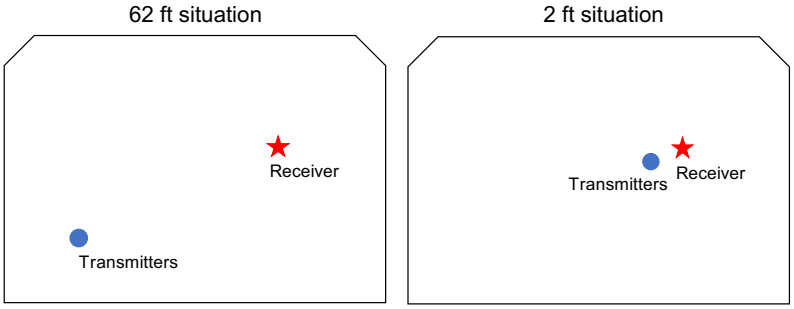


Figure 3.12 – Assumption 1 of transmitters and receiver location for Oracle database.

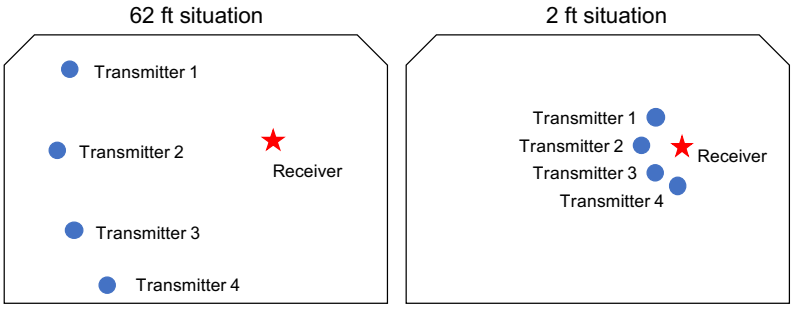


Figure 3.13 – Assumption 2 of transmitters and receiver location for Oracle database.

C. Conclusion

The study of the Oracle database reveals the importance of having database design information to understand the network behavior, without this information, we can only make some assumptions about the results. Here the study shows that it is easier to separate the devices in long distance context than short distance however, the more important the distance is, the more important the interferences are so the network probably uses other characteristics than RFF to classify the devices. This primary study highlights the importance of data and shows that the networks seem to achieve fairly similar performance on classification accuracy with the same behavior.

3.4 Need for Virtual Database Generator

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 addition, the experimental database must provide different test contexts to validate RFF learning resilience. 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 very useful to study RFF identification scenarios and to design a robust RFF identification protocol [121]. However, the authors only give access to the final database [102] that could be useful to reproduce the experiments but limits the exploration possibilities. The community misses, therefore, a generic virtual database generator. The next chapter presents the virtual database generator that creates a database based on the scenario description to study the DL RFF identification process and explore database design space such as the number of signals, the type of signals, and the impact of each impairment.

Chapter 4

Proposed Virtual Database Generator

This chapter presents the **R**ad**i**o **F**requenc**y** **Fi**ngerprint **V**irtual **D**atabse **G**enerator RiFyFi_VDG, which is the first contribution of this PhD. This generator, coded in the Julia language, was conceived and developed to help the community address and understand RFF identification with the DL technique, and provides a tool for exploring database design, as a digital twin. The RiFyFi_VDG can create a database of signals in a few seconds from different simulated transmitters based on RF transmission models and parametric impairments models. The databases created by the generator can be used to understand RFF identification with the DL technique and allow the exploration of impairments, database design, and learning models. Creating a virtual database requires digital communication models, hardware impairments models, and wireless propagation channel models, that are detailed in Section 4.1. Section 4.2 presents the practical use of the database generator with examples. Finally, we propose an overview of the global framework used for RFF identification called RiFyFi framework coded in Julia language too in Section 4.3. A part of this work was done at Tampere University in Finland in 2023, under the supervision of Pr. Mikko Valkama and Pr. Elena Simona Lohan.

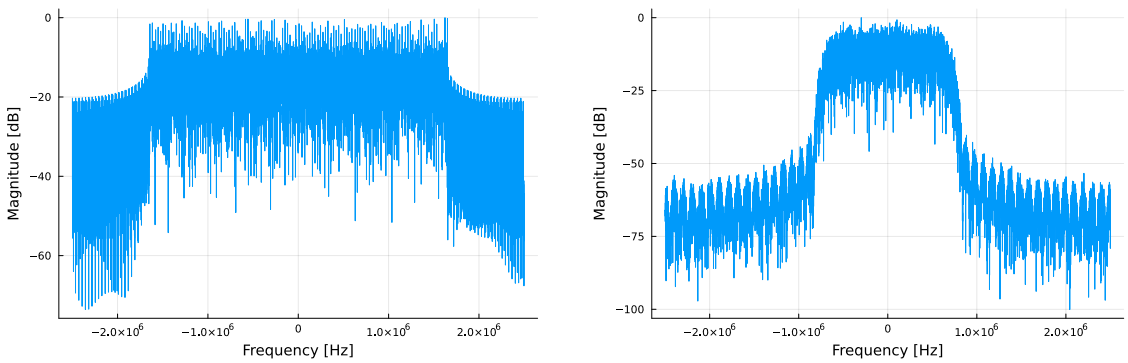
4.1 Virtual Database and Radio Model

RiFyFi_VDG is a Julia package integrated into the RiFyFi system, allowing virtual database creation thanks to wireless communication models, hardware impairments models, and wireless propagation channel models. This section describes the models implemented in this generator.

A. Symbols

First, the wireless communication model between an emitter and a receiver requires creating a signal for transmission. Here, two signal modulations are implemented in RiFyFi_VDG: single-carrier modulation and OFDM. For single-carrier, we consider that the binary sequence is modulated by QAM symbols and then followed by a single-carrier modulation. For OFDM, we consider that the binary sequence is modulated by QAM symbols and then followed by an OFDM modulation with subcarrier-based pilot insertion. OFDM modulation is massively used in standard communication in WiFi for example, and so particularly in the RFF database as it is shown in Table 3.1 and its signal varies greatly in amplitude which makes it interesting for the analysis of non-linear imperfections [1].

In this PhD, we mainly focus on an OFDM transmission, similar to a WiFi communication system, we have implemented and studied the other protocol to present the flexibility of the RiFyFi_VDG. After the symbol modulation, the binary sequence becomes a complex sequence and both parts of the complex signal are separately processed and modulated at the carrier frequency f_c. Figures 4.1 show the PSD obtained for an OFDM modulation and a single-carrier modulation, thanks to RiFyFi_VDG. For single-carrier modulation, the roll off is $\beta = 0.3$, with an oversampling factor of 4 and a square root raised cosine filter of 6, and a 4-QAM. For OFDM the total number of subcarriers is 512 with 336 active subcarriers (224 for data and 112 for pilot).



(a) OFDM. (b) Single-carrier.

Figure 4.1 – Power spectral density illustrations for OFDM and single-carrier modulation.

1. It is possible to use other communication, a proof of concept has been proposed with a LoRa interface [11]

B. Transmitter impairment models

In this subsection, the initial model proposed in Chapter 2 is completed with impairment models of each hardware component. As a remember, the DAC, the LO, and the PA, distort the signal and create the signature called the RFF of the transmitter denoted $\mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}$. The emitted signal can be modeled by:

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{RFF}_{\mathrm{Tx}}}\left(x(t)\right), \tag{4.1}$$

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{PA}} \circ \mathcal{F}_{\mathrm{LO}} \circ \mathcal{F}_{\mathrm{DAC}}\left(x(t)\right). \tag{4.2}$$

In this section, the objective is to detail the impairment models behind (4.2). The impairments modeling is described in Figure 4.2 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. [121] this study is focused on the main features: CFO impairments $\Delta\omega$, gain and phase IQ imbalance g_I, g_Q and θ, PN $\Phi(t)$, and PA nonlinearity. We decided to omit DAC deficiencies because their deficiencies are not relevant for RFF identification, as shown by Polak et al. [75]. The pattern of each depreciation is described below.

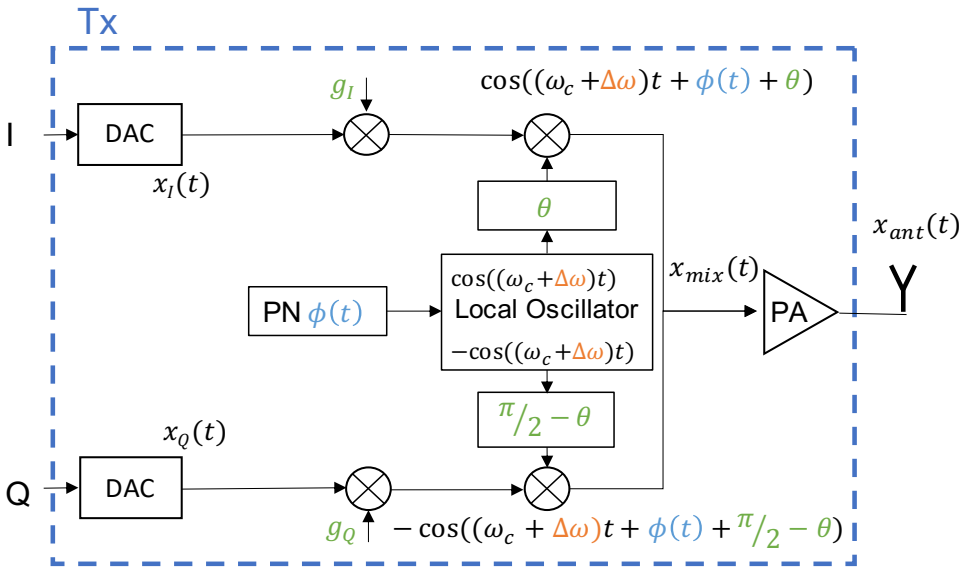


Figure 4.2 – Homodyne transmitter chain architecture with impairments.

Before the LO, the analytical signal is modeled as:

$$\underline{x}(t) = x_I(t) + jx_Q(t), \tag{4.3}$$

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

Carrier Frequency Offset (CFO) impairments: The LO modulates the signal at the ideal carrier frequency, f_c. However, CFO impairments introduces a frequency offset Δf, resulting in the effective carrier frequency, f_0, noted as:

$$f_0 = f_c + \Delta f. \tag{4.4}$$

For the sake of brevity, the models are expressed in terms of angular frequency with $\omega_* = 2\pi f_*$. Based on (2.3), the modulated signal $x_{mix}(t)$ with such impairment, is expressed as a gain and phase error by:

$$x_{mix}(t) = x_I(t)\cos((\omega_c + \Delta\omega)t) - x_Q(t)\sin((\omega_c + \Delta\omega)t), \tag{4.5}$$

which can be equivalently written as:

$$\underline{x}_{mix}(t) = \underline{x}(t)e^{j(\omega_c+\Delta\omega)t}, \tag{4.6}$$

$$x_{mix}(t) = \Re(\underline{x}_{mix}(t)), \tag{4.7}$$

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 4.2 in the form:

$$\underline{X}_{LO}(t) = g_I\cos(\omega_0 t + \theta) + jg_Q\cos(\omega_0 t + \frac{\pi}{2} - \theta), \tag{4.8}$$

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

where θ is the phase impairment, and g_I and g_Q the gain impairments. The expression can be simplified as done by Valkama et al. [111]:

$$\underline{X}_{LO}(t) = K_1 e^{-j\omega_0 t} + K_2 e^{j\omega_0 t}, \tag{4.9}$$
$$\text{where } K_1 = \frac{g_I e^{-j\theta} + g_Q e^{j\theta}}{2}, \quad K_2 = \frac{g_I e^{j\theta} - g_Q e^{-j\theta}}{2}.$$

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

$$\begin{aligned} \underline{x}_{mix}(t) &= \underline{x}(t) \times \underline{X}_{LO}(t), \\ &= \underline{x}(t) K_1 e^{-j\omega_0 t} + \underline{x}(t) K_2 e^{j\omega_0 t}. \end{aligned} \tag{4.10}$$

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 (PN) 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 [122]. The LO PN $\phi(t)$ may be modeled by [78]:

$$\phi(t) = \sqrt{c} B(t), \tag{4.11}$$

where $B(t)$ denotes a standard Wiener process and parameter c describes the LO quality called diffusion rate [78]. $B(t)$ is defined as $B(t_2) - B(t_1)$, where t_1 and t_2 are the duration of the noise with variance $\sqrt{t_2 - t_1}$. $\mathcal{N}(0, 1)$, where $\mathcal{N}(0, 1)$ is a normal law with zero mean and variance 1. In the rest of the PhD, we consider the digital Wiener PN model parameterized by its state noise variance σ_ξ^2 [31].

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

$$\underline{x}_{mix}(t) = \underline{x}(t) K_1 e^{-j(\omega_0 t + \phi(t))} + \underline{x}(t) K_2 e^{j(\omega_0 t + \phi(t))}. \tag{4.12}$$

Power Amplifier (PA) impairments without memory: At the end of the transmission chain, the PA amplifies a low-power signal to a higher-power one. Here we propose a memoryless model of PA where the past of the signal does not affect the amplification of the present. To model the memoryless nonlinear effect of the PA in our system, the Saleh

model used in SoA [121] is chosen. The non-linearity is modeled as amplitude/amplitude (AM/AM) denoted $A(t)$ and amplitude/phase (AM/PM) distortions denoted $\xi(t)$.

$$A(t) = \frac{\alpha_{AM}|\underline{x}_{mix}(t)|}{1+\beta_{AM}|\underline{x}_{mix}(t)|^2}, \qquad (4.13)$$
$$\xi(t) = \frac{\alpha_{PM}|\underline{x}_{mix}(t)|^2}{1+\beta_{PM}|\underline{x}_{mix}(t)|^2},$$

where $|.|$ denoted L1 norm. $\alpha_{AM}, \alpha_{PM}, \beta_{AM}, \beta_{PM}$ are the parameters of Saleh model [121]. Finally, the signal $\underline{x}_{PA}(t)$ after the PA is modeled as:

$$\underline{x}_{PA}(t) = A(t)e^{j(\angle \underline{x}_{mix}(t)+\xi(t))}, \qquad (4.14)$$

where $\angle$ represent the angle of $\underline{x}_{mix}(t)$.

Power amplifier impairments with memory: The power amplifier can be modeled with a memory effect, the signal $\underline{x}_{PAM}(t)$ after the PA is modeled as [74]:

$$x_{PAM}(t) = \sum_{\substack{p=1 \\ p\ odd}}^{P} f_p(t) * (|\underline{x}_{mix}(t)|^{p-1}|\underline{x}_{mix}(t)|), \qquad (4.15)$$

where P is the nonlinearity order of the model and $f_p(t)$ denotes the p^{th}-order response of the polynomial model.

C. Note on the impact of the carrier frequency:

The models proposed and used in our database generator are valid whatever the carrier frequency value, but the parameterization of the model will depend on the carrier frequency. For example, the CFO depends on the carrier frequency following:

$$\Delta f_{max} = \frac{ppm}{10^6} f_c, \qquad (19)$$

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 work corresponds to a CFO of 300 Hz at 2.4 GHz.

D. Channel models

The channel model implemented in our database generator is a wireless flat-fading transmission with random delay spread. The maximum of the delay spread is set at 36, which corresponds to the CP of the OFDM considered here. The signal obtained after the channel is modeled as:

$$\underline{y}(t) = h(t) * \underline{x}_{PA}(t) + n(t), \tag{4.16}$$

where $*$ is the convolution operator, h represents the impluse response of the propagation channel and, $n(t)$ is a AWGN. The power of the tap follow a rayleigh model centered around 1 (e.g Rice model), to ensure that neough power is always received at the reception stage.

$$h(t) = \begin{cases} 1 + \alpha\gamma, & \text{if } t = \tau \\ 0 & \text{else.} \end{cases} \tag{4.17}$$

where α corresponds to the rayleigh model random variable and γ respresent a ponderating factor with the value of 0.3. Figure 4.3 presents the different parameters influence.

The doppler effect is not considered and the channel power does not change in function of time in other word we considered that the devices are fixed during the transmission. In order to model changes in environmental propagation, we consider that different flat fading channels can be encountered. To achieve this, we generate different channels applied to a few consecutive sequences of 256 IQ samples, with each channel having a different random power and delay spread

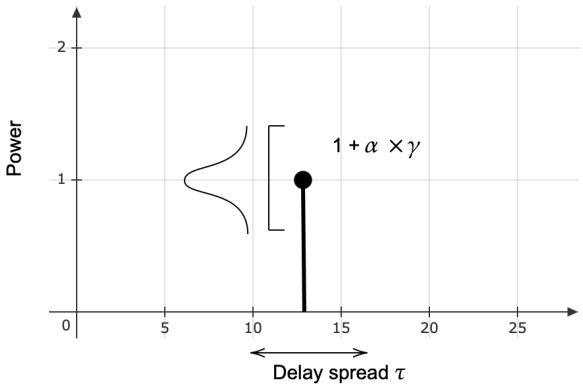


Figure 4.3 – Parametric propagation channel model.

E. Conclusion:

All these impairment models are implemented in RiFyFi_VDG, a device is defined by *device scenario* which corresponds to 8 parameters: the gain and phase imbalance, the value of the CFO, the variance of PN, and the 4 parameters of PA Saleh model. The value of each parameter as well as the similarities between devices is discussed in Section 4.2.

4.2 Practical use: from models to scenarios

This section presents the practical use of RiFyFi_VDG and how to create different database scenarios thanks to the parametric generator and models. First, the device scenario is presented in Subsection 4.2.1. Then the database parameters are presented in Subsection 4.2.2. Finally, each step of signal creation is presented with different examples with the symbols scenario in Subsection 4.2.3, the fingerprint scenario in Subsection 4.2.4, and finally the propagation channel scenario is presented in Subsection 4.2.5.

4.2.1 Impairment similarity scenarios

To simulate the behavior of different transmitters, the values of the impairments have to be different for each device with more or less similarity between transmitters. The *impairment similarity* is a critical point of the SoA, because recognizing two devices from the same manufacturer is more difficult than two devices from two manufacturers. The *impairment similarity* scenarios are described in a JSON file. It is possible to create a random devices scenario or load a devices scenario with a particular percentage of similarity between transmitter impairements, created before. The percentage similarity scenarios are described in the next chapter. Some authors of the SoA propose to create a grid to make sure that the space between two impairment values is sufficient. In Zhang paper [121], the impairments follow a uniform random distribution within an interval. In this PhD, we study the impact of the similarity between two RFF devices.

4.2.2 Database design parameters

As the SoA shows, the RFF identification conditions are multiple such as the type of data within the frame used to identify the transmitter [48], the level of noise [47], the number of signals, the number of different propagation channels in the database, the number of transmitters and the similarity between them. It is difficult to determine which are the influencing parameters that can affect the data and behavior of the network.

Therefore, exploring these different scenarios, by changing only one of the settings at a time with a single framework seems interesting and could help in designing a real database. With RiFyFi_VDG for each database creation, it is possible to choose:

— the number of transmitters N_{Tx},

— the number of signals per device in the database N_{signals},

— the number of IQ samples in a signal $ChunkSize$,

— the frame type and the modulation,

— the activated impairments,

— the similarities between RFF emitters/ impairment similarity scenario,

— the level of noise,

— the channel scenario for training and test sets,

— the repartion of training and test set.

All of these parameters are part of the database generator in Figure 4.4.

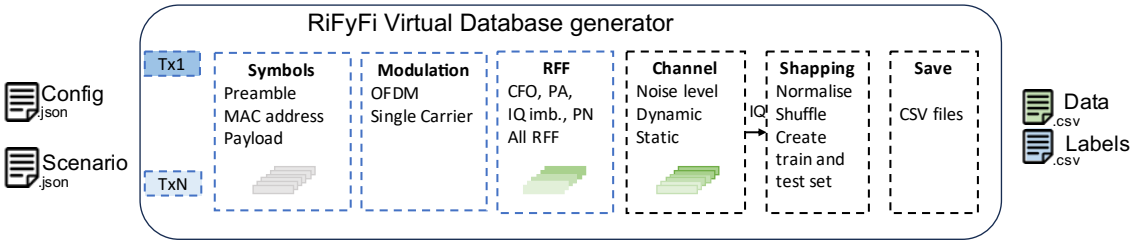


Figure 4.4 – Parametric database generator chain.

In RiFyFi, a *signal scenario* consists of key parameters separated into 4 types: symbol generation, type of modulation, RFF, and propagation channel as shown in Figure 4.4. First, the *symbol*, represents the type of signal/frame used for identification: Preamble, MAC address, or payload (Pre, MAC, Pay). Then, it is possible to choose the type of *Modulation* between at least two possibilities: OFDM or single-carrier. 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 the propagation conditions, such as noise or channel model. Finally, it is possible to add a receiver model with its own RFF [2]. At the end, two matrices are created, one with data

2. Note that in this work the impact of the receiver will not be explored, and a unique receiver without any impairments is considered.

and the other with labels due to the supervised learning context. The data matrix is composed of IQ samples of size ($ChunkSize$, 2, $N_{\text{signals}} \times N_{\text{Tx}}$) is created. This matrix is transformed by the shaping block to obtain the required format for training, the data is shuffled and split to create both, training and test sets. Then the database composed of data and labels is saved in 2 CSV files for train and 2 CSV files for test to be used by the network. The labels matrix is composed of 2 dimensions ($N_{\text{Tx}}, N_{\text{signals}}$).The labels are saved in vector format with only the corresponding Tx label for each signal (in the same order as in the data file).

4.2.3 Symbols scenario and Modulation

This subsection addresses the type of binary sequence and the modulation used. Considering the binary sequence, three scenarios are proposed:

— Preamble, all sequences are the same,

— MAC address, each transmitter has a particular sequence,

— Payload, all sequences are different.

Then the binary sequence is modulated to obtain symbols, we have proposed two scenarios:

— OFDM

— Single Carrier

For OFDM, a symbol is composed of 548 IQ samples with an FFT size of 512 and a CP size of 36. For single-carrier, a symbol is composed of 4 IQ samples.

Depending on the modulation, the number of symbols that have to be generated is determined to obtain a sequence of 64 $ChunkSize$ IQ samples, this sequence is called a burst. The symbols are randomly generated and then, depending on the desired type of frame (Preamble, MAC address, or Payload), the other bursts are created to obtain $N_{\text{signals}} \times ChunkSize$ IQ samples. Figure 4.5 presents the three modes.

Creating a Preamble-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. In RiFyFi_VDG, the random generation of symbols is controlled by a seed, therefore to obtain a preamble the seed is always the same. Figure 4.5 presents the different modes. For Preamble, mode the long gray rectangle represents a burst and is repeated for all transmitters.

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.

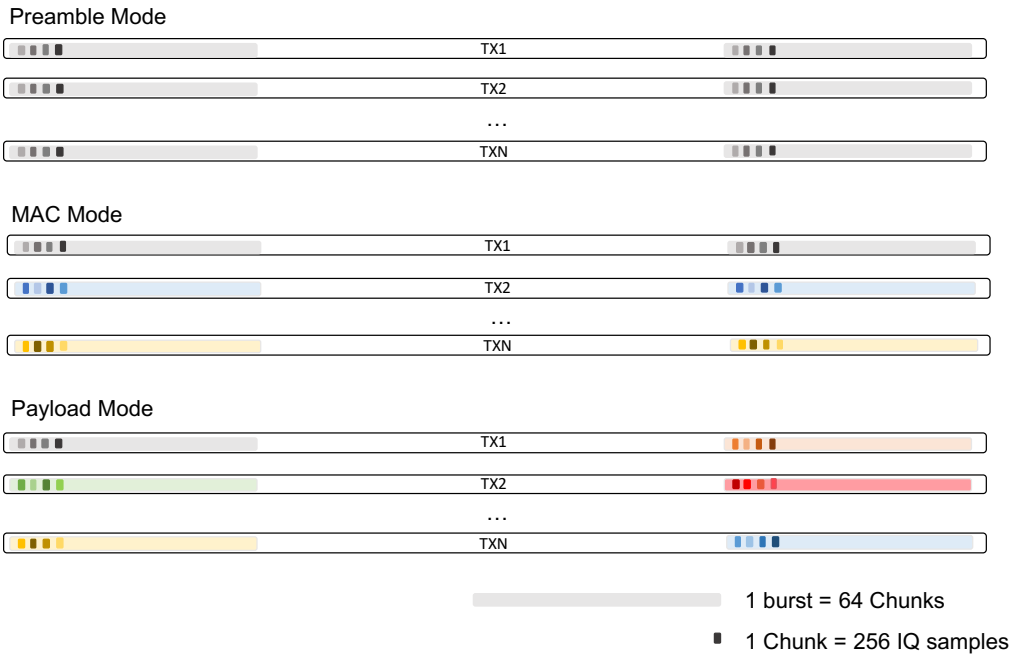


Figure 4.5 – Generation of different types of sequence/frame.

To simulate this scenario, the seed is changed for each transmitter. In Figure 4.5 the color of the rectangle changes for each transmitter.

The last possibility is to generate different sequences for each transmission, this scenario is called Payload where the identification can only be done through RFF. To simulate this scenario, the seed is changed for each burst and each transmitter. In Figure 4.5 the color of the rectangle always changes.

Example

We propose to create a database of two devices: $N_{\text{Tx}} = 2$, with 1000 signals per transmitter: $N_{\text{signals}} = 1000$. Each signal corresponds to 256 IQ samples: $ChunkSize = 256$ IQ samples. The emitted signal is a preamble transmitting with OFDM modulation. The sampling frequency is 5.2608×10^6 Hz.

Figure 4.6 presents two bursts of the "Example" data from transmitter 1 [3] without impairments. The first burst which corresponds to 16384 complex IQ samples is in blue and the second is in yellow. As shown in Figure 4.6 the preamble time is $3.2ms$ and is

3. or transmitter 2, because at this step of the data creation, the signal of transmitter 1 and 2 are similar because we choose Preamble mode

repeated to obtain 256,000 IQ samples. The peaks observed in the figure are due to the pilot of the preamble.

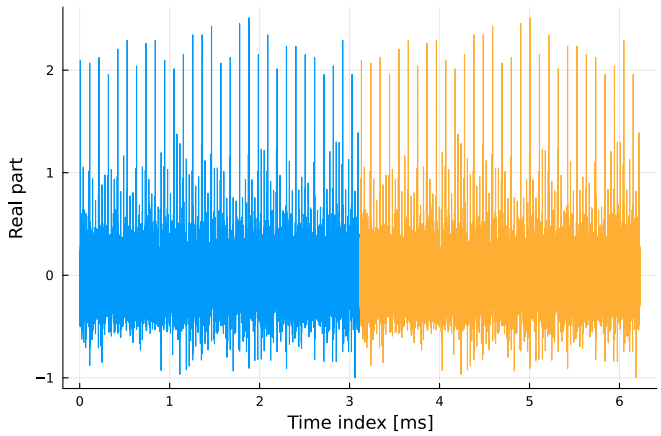


Figure 4.6 – Part of the signal without impairments.

4.2.4 Fingerprint Scenario

In fingerprint scenarios, it is possible to activate different impairments which is not possible with real devices. However, it offers exploration possibilities. First of all, it is possible to activate one or multiple impairments, to combine their effects. Six scenarios are created:

— CFO: only CFO impairment

— Imbalance: only gain and phase IQ imbalance impairment

— PN: only PN impairment

— PA: PA impairment with Saleh model

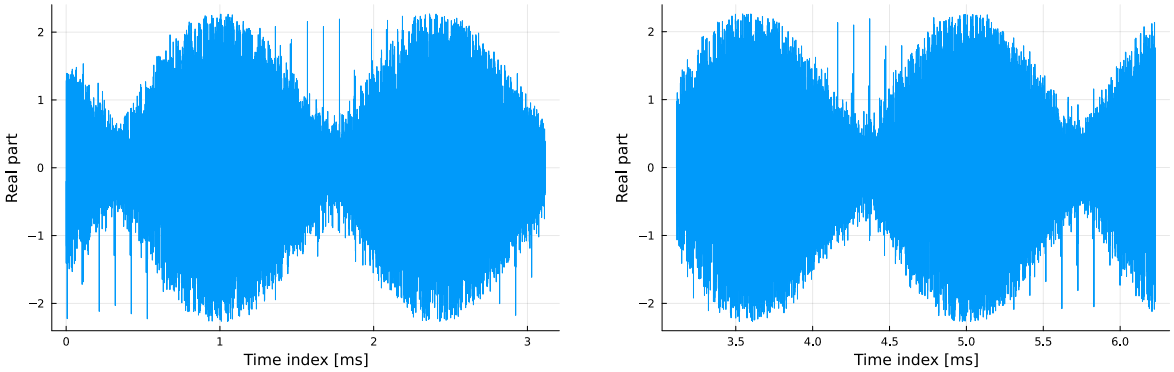
— PA_memory: PA with measured memory model

— All_impairments: CFO, imbalance, PA with Saleh model and PN.

In the next chapter, the impact of impairments is independently studied, and then the most realistic scenario is addressed: All_impairments.

Example

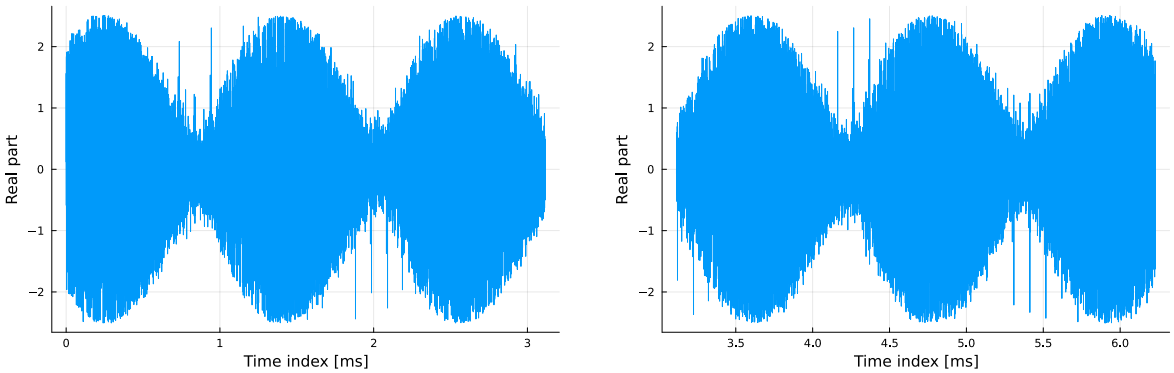
In the example context, all the impairments are activated with clearly different *impairment similarity scenarios* presenting in Table 4.1.



(a) View of burst 1, with impairments.

(b) View of burst 2, with impairments.

Figure 4.7 – Parts of the signal of the transmitter Tx1 with impairments.



(a) View of burst 1, with impairments.

(b) View of burst 2, with impairments.

Figure 4.8 – Parts of the signal of the transmitter Tx2 with impairments.

Figures 4.7 and 4.8 present two different parts of the signal of transmitters one and two respectively. The figures show the impact of the impairments, in particular the CFO impact which creates this waveform. The CFO of Tx2 is more important than the CFO of Tx1 and this phenomenon is observable in Figures 4.8. Indeed the CFO leads to a frequency modulation whose frequency is higher for the second transmitter. It is really easy to change the parameters of Table 4.1 to understand the impact of the impairments on the signal. Unlike many applications in image classification, such as dog or cat recognition, here it is difficult to know by eye which transmitter the signal comes from.

Impairment	Parameters	Value Tx1	Value Tx2
CFO	Δf	270 Hz	330 Hz
Imbalance	g_Q	1.350 dB	1.650 dB
	θ	0°	5°
PN	σ_ξ^2	9×10^{-8}	1×10^{-7}
PA	α_{AM}	1.943	2.375
	β_{AM}	1.037	1.267
	α_{PM}	3.603	4.404
	β_{PM}	8.194	10.014

Table 4.1 – Values chosen for impairment parameters.

4.2.5 Channel or Noise Scenario

Finally, it is possible to add AWGN or propagation channels to the transmitted signal to model the transmission over the air. Figures 4.9 and 4.10 present different scenarios. The signal is the first part of Tx1 with an SNR of 10 dB in Figure 4.9a, and an SNR of 0 dB in Figure 4.9b. The noise affects the waveform of the signal and will affect the classification accuracy. The propagation channel models implemented in this database generator are based on a power delay profile, such as a static multipath channel. Here we define two classical channel profiles which are the ETU and EVA models, presented in previous section.

Example

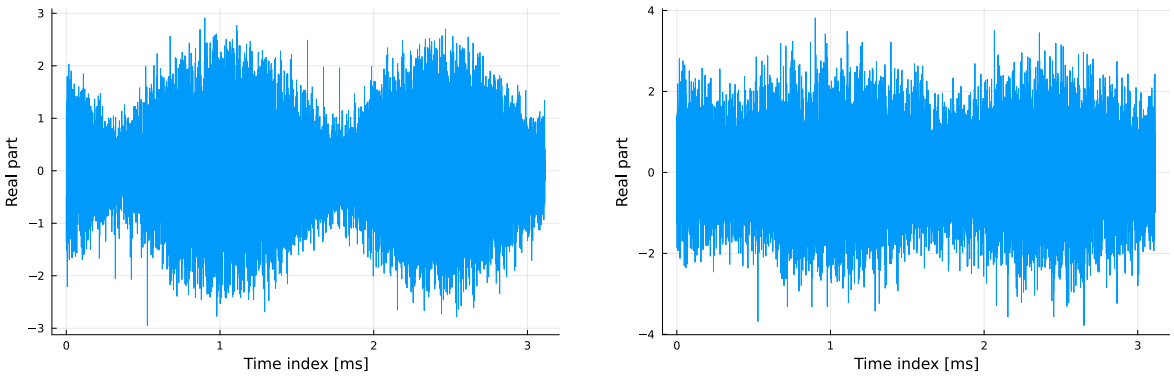
In the example context, we propose to add an additive noise with 10 dB of SNR, and 0 dB, Figure 4.9.

Example

In the example context, we propose to add an ETU propagation model with 9 taps, Figure 4.10.

4.2.6 Conclusion

The RiFyFi_VDG is easy to use and to adapt with several parameters to create various scenarios for training the network or testing and evaluate the classification abilities.



(a) SNR: 10 dB. (b) SNR: 0 dB

Figure 4.9 – Parts of the signal of the transmitter Tx2 with impairments and noise.

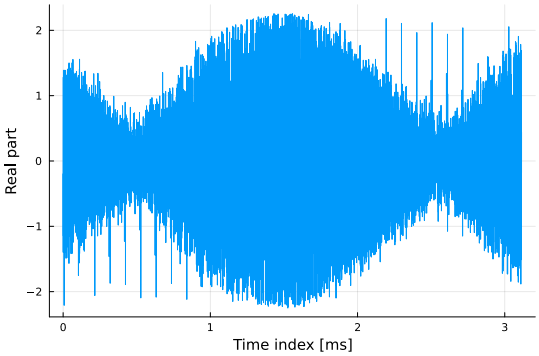


Figure 4.10 – Part of the signal of the transmitter Tx2 with impairments and multipath channel.

RiFyFi_VDG is an accessible tool in [11]. An example script to use RiFyFi_VDG is available in order to reproduce an example.

4.3 RiFyFi System overview

In this section, we present our flexible framework for RFF identification coded in Julia language and explain some technical points about the development of RiFyFi framework to make it completely flexible. Julia [29] is a high-level language, efficient in: abstraction and execution, with many DL and telecom libraries [56]. The framework is composed of i) a database management block and ii) a classification stage based on DL. The global framework, presented in Figure 4.11 offers the flexibility to load data from an existing database or use RiFyFi_VDG to create a new virtual database based on a scenario description.

To ensure the flexibility of the framework, we create some data structures: one for database description which integrates a structure for augmentation parameters, and one for network definition which integrates a structure for training parameters. The structures called Param_Data and Param_Network, are the input of the different framework parts, seen in Figure 4.11.

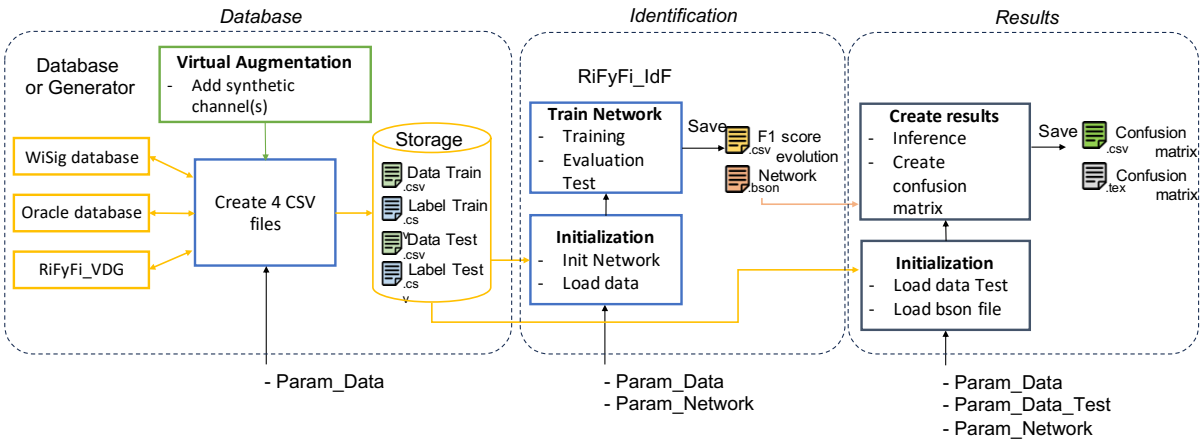


Figure 4.11 – RiFyFi Framework flow.

4.3.1 Databases in RiFyFi

This subsection describes the database management block with the database specificities. This block takes as input the Param_Data structure and creates the corresponding database by loading existing signals or creating virtual ones. The database is shuffled and separated for train and testing before being saved in CSV files with a particular name that describes the database. Different Julia packages have been created to load the data, one for each database: Oracle, WiSig which are composed of fixed data, and RiFyFi which is considered as an infinite possibility database. A virtual augmentation package has been created to add a propagation channel to the data. This package can be used with RiFyFi_VDG to simulate wireless communication or with a real database such as WiSig to augment the data.

First, the user has to describe the database parameters to create the Param_Data structure which is the input of the first bloc *Database* in Figure 4.11. This structure depends on the database: for example, the WiSig database proposes different receivers so the structure has to define the receiver(s) used. However, the Oracle database only has one receiver but different distances. That is why a Julia package is created for each database to

allow data structure personalization. To use another database with RiFyFi, the user has to create a new Julia package, inspiring from the existing ones. The common parameters of Param_Data structure are the name of the databases: WiSig, Oracle, VDG; the number of transmitters; the number of signals; the ChunkSize, and the channel parameters.

— For the WiSig database the Param_Data structure contains information about the receiver(s), the day(s) of capturing data, and the equalization or not.

— For the Oracle database, the added information is the distance(s) between transmitter and receiver and which one of both captures is chosen.

— The virtual database requires more information such as the type of frame, type of modulation, the activated impairments, and the name of the impairment similarity scenario to load the correct JSON file.

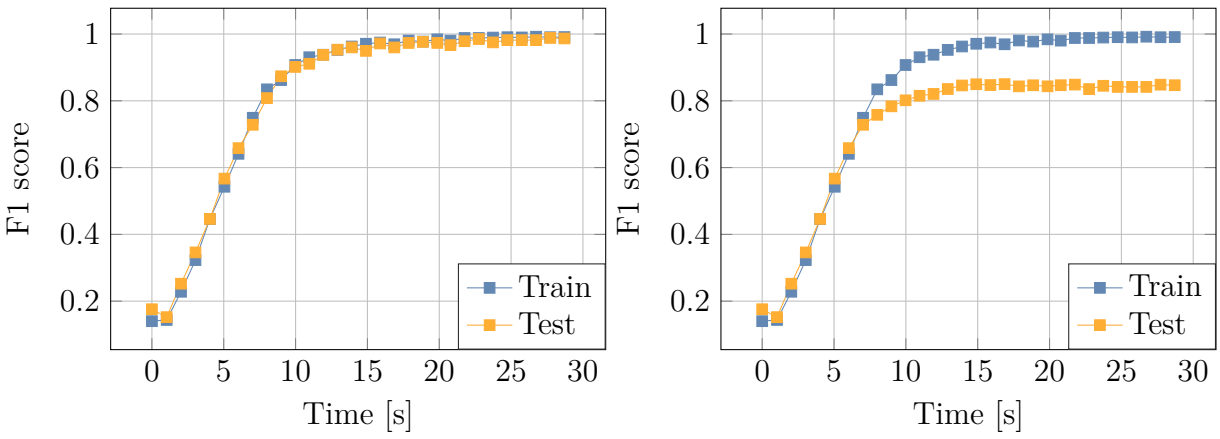
The channel parameters are used by the Virtual Augmentation package, seen in Figure 4.11. The objective is to add a channel model to augment the database at the receiver side in a real data context or simulate a propagation channel in a virtual data context. This allows to evaluate the impact of channel variation exploration in different contexts. For instance, Al-Shawabka et al. [94] experiment with different propagation channel contexts with variation and conclude with the need to have a robust system to channel variation. However, 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 RiFyFi_VDG flexibility allows the different scenarios exploration to find or create a robust identification system depending on the application context.

4.3.2 RFF Identification training in RiFyFi

The *Identification* stage in Figure 4.11 takes as inputs the description of the required database and the description of network architecture and training parameters. The description of the network consists of the name here three SoA architectures that are implemented, the input size of the network, the number of transmitters it has to classify, and the value of dropout. The training parameters concern the learning rate, the number of epochs, and the batch size. The network architectures implemented in RiFyFi are Sankhe_2020, Sankhe_2019, and Arroyo_2022 and have been described in Chapter 3.

First, the training and testing databases are loaded with the labels, then the network is initialized with the corresponding architecture. The input size depends on the ChunkSize of the signal and the output depends on the number of classifying devices, the outputs of the network are the probability of belonging to a class. Finally, the network is trained with different learning parameters. 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 each epoch, the F1 score is computed for training and testing data and saved with execution time to obtain the curves shown in Figure 4.12. These curves represent the F1 score evolution during training at each epoch for training and test sets. These curves allow us to compare the training evolution in different contexts or with different parameters or architectures. For example, two situations are presented the first one is an ideal training with close performance on training and test sets. The second situation shows a performance stall in the test, that is related to an overfitting of training data.

The training ends when the shutdown condition has been reached which is when the network reaches $P\%$ of mean accuracy on training data. This is probably not the best condition but it allows the behavior comparison of the networks. The framework saves the network status in `.bson` file and saves the performance evolution during the training phase in a `.csv` file.



(a) Ideal context.

(b) Overfitting context.

Figure 4.12 – F1 score curves examples which can obtain with RiFyFi framework.

4.3.3 Evaluation in RiFyFi

Finally, to evaluate the network it is possible to use another database or the testing one, the network predicts the label, and thanks to the true labels a confusion matrix is created to show the capacity of the network to classify the signal in the correct class. The ideal confusion matrix is 100% on the diagonal and 0% for the other case. However, the network can confuse the transmitters and classify some signals from transmitter 4 as transmitter 2.

4.4 Conclusion

The RiFyFi framework has been implemented and improved all along this PhD, to explore the network architecture, the learning parameters, the different databases, and finally the RiFyFi_VDG generator has been developed to explore more specifically database design. The next chapter presents an investigation of the individual impact of impairments with the network introduced in [91] aimed to reveal the most discriminant impairments for this network. The selection of an [91]-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 [47, 102, 91, 84].

Chapter 5

Understanding RFF with Virtual Databases: Experiments and Results

This chapter presents the results of the first contribution of this PhD. The major contribution is the virtual database generator presented before and the global open-source Julia framework for RFF identification. The previous studies presented in Chapter 2, highlight the similar behavior of the 3 CNNs chosen. For the sake of brevity, we suggest using only one network architecture in this chapter, which is Sankhe_2020. In Section 5.1 the impairments are studied separately to draw some preliminary conclusions, and then in Section 5.2 several conglomerate studies are proposed, with in particular changing the frame type and modulation, in the presence or absence of a propagation channel. Finally, Section 5.4 concludes this first contribution.

5.1 Investigation of the individual impact of impairments

In this section, impairments are separately studied with different confidence intervals, described in this section, around a fixed mean value, inspired by the SoA [121] and defined in Table 5.1. In [102], 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, because the values of IQ imbalance are too important. Zhang et al. [121] set the range of gain and phase imbalances to [-1 1] dB and [-5 5] degrees, which are more realistic values. For the PA they used a Saleh model with the values presented in Table 5.1 which vary within ±5%. In our work, different intervals are explored. The confidence interval is a metric to model the

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

Table 5.1 – Mean value chosen for impairment parameters.

disparity between the electric circuits embedded in the transmitters. The RFF identification complexity depends on the similarities between the RFF transmitters. For a given number of transmitters, a large confidence interval reduces the similarity between two transmitters. However a small confidence interval increases the RFF similarity between devices, and, therefore, it makes the identification difficult. For this study, some learning parameters are empirically adjusted upstream for each impairment to compare them in favorable situations. The parameters are specifically the dropout (dr), the learning rate and the batch size, which is always set to 64 in this chapter. 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 simultaneously. The results presented in this section are obtained by means of 5 different training networks with different seeds to ensure the results obtained. The different colors in the tables evaluated the performance (shades of green, orange, or red).

5.1.1 CFO

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

$$\Delta f_{Tx1} = \bar{\Delta f}(1 - p\%), \tag{5.1}$$

$$\Delta f_{Tx2} = \bar{\Delta f}(1 + p\%), \tag{5.2}$$

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

Figure 5.1 presents the F1 score evolution during the training phase and Table 5.2 summarizes the results with the mean F1 score obtained during the training phase on the training set and test set at different epochs. For the next impairments and for the sake of conciseness we only use the tables to present results.

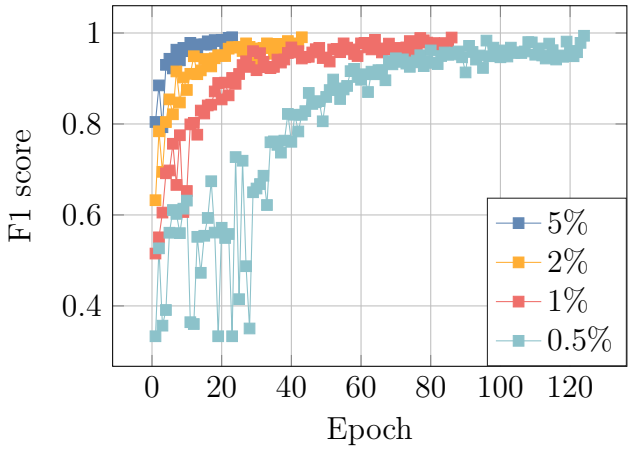


Figure 5.1 – F1 score evolution during training for the different CFO scenario similarity, 2 transmitters and 900 signals per transmitter for train.

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 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 the RFF transmitter similarity scenario and the capacity of the network to separate transmitters.

F1 score at	20 epochs		50 epochs		100 epochs		315 epochs	
p	Train	Test	Train	Test	Train	Test	Train	Test
5%	98%	98%						
2%	93%	91%	98%	98%				
1%	53%	51%	92%	89%	97%	95%		
0.5%	48%	46%	52%	47%	57%	56%	98%	87%

Table 5.2 – Mean F1 score evolution during training phase for different CFO scenarios.

5.1.2 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.

$$g_{QTx1} = g_{QTx3} = \bar{g_Q}(1 - p\%) \tag{5.3}$$

$$g_{QTx2} = g_{QTx4} = \bar{g_Q}(1 + p\%) \tag{5.4}$$

$$\theta_{Tx1} = \theta_{Tx2} = \theta_{max} \tag{5.5}$$

$$\theta_{Tx3} = \theta_{Tx4} = \theta_{min} \tag{5.6}$$

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 2 in Appendix B presents gain and phase values for each transmitter. The results are obtained without dropout and a learning rate at 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%

Table 5.3 – Mean F1 score evolution during training phase for different IQ imbalance impairments, $\gamma = 10^{-4}$.

Table 5.3 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 test performance

drops even after long training. Then, comparing the first row of the 10% similarity scenario and the first row of the 3% scenario shows a slight difference in the F1 score value at the same time. Moreover in combination with {2°, 3°} and gain over 3%, results show an over-learning on training data as it stops around 50% on Test data. The analysis of the confusion matrix in Table 5.4b, 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 5.4a, which highlights a classification issue, as the network only seems to recognize three classes.

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

(a) g: 1% 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

(b) g: 10% and $[2°; 3°]$ combination.

Table 5.4 – Confusion Matrix for test data for IQ imbalance impairment.

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

5.1.3 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 dropouts. However, the results are always bad: the F1 score on the test set is about 25%, 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.

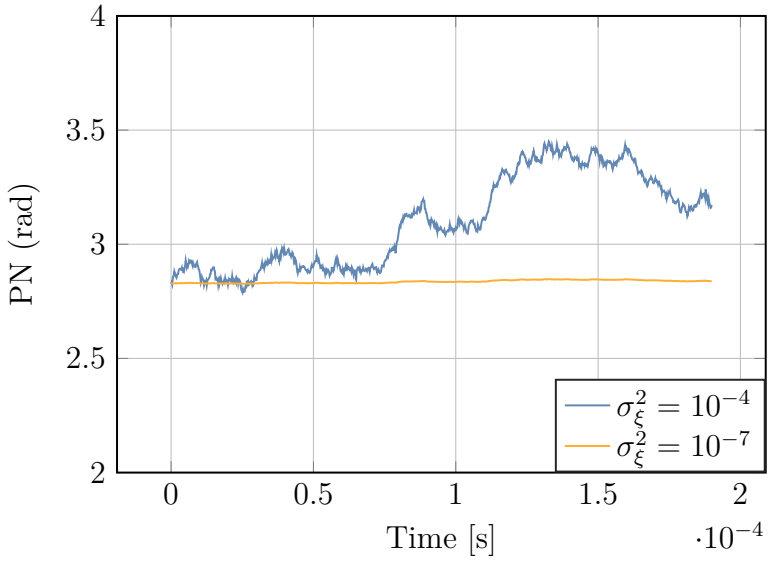


Figure 5.2 – PN realizations.

5.1.4 Power Amplifier

To study the PA effect, two types of model introduced in Section 4.1 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 [2]. 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 4 of the Appendix B for different impairment similarities expressed as:
For α_{AM} and α_{PM},

$$\alpha_{Txi} = \alpha_{Txi+2} = \alpha(1 - p\%) \text{ with i} = 1, \tag{5.7}$$

$$\alpha_{Txi} = \alpha_{Txi+2} = \alpha(1 + p\%) \text{ with i} = 2. \tag{5.8}$$

For β_{AM} and β_{PM},

$$\beta_{Txi} = \beta_{Txi+1} = \beta(1 + p\%) \text{ with i} = 1, \tag{5.9}$$

$$\beta_{Txi} = \beta_{Txi+1} = \beta(1 - p\%) \text{ with i} = 3. \tag{5.10}$$

Different experiments are done and present instability of the network during the training phase. To reduce this problem the dropout is set at $dr = 0.25$ and the learning rate is

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%

Table 5.5 – Mean F1 score evolution during training phase for different PA impairments, $\gamma = 10^{-5}$ dr $= 0.25$.

decreased at $\gamma = 10^{-5}$. Table 5.5 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.

5.1.5 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.

5.2 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 centered around $\sigma_\xi^2 = 10^{-7}$and $\sigma_\xi^2 = 10^{-4}$ are explored. The values chosen for each parameter of the 6 transmitters are calculated following (5.13).

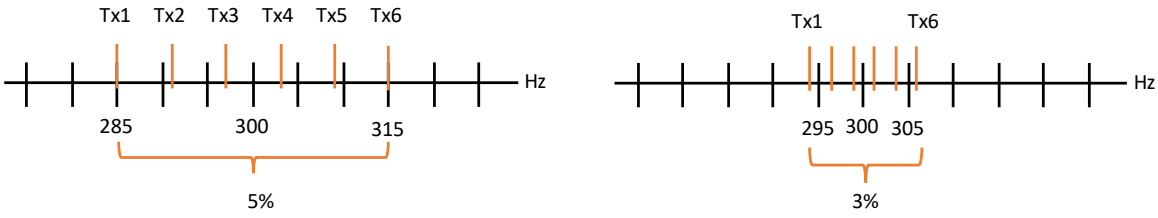
Table 5 in Appendix B presents the parameter values calculated for $p = 5\%$ similarity, for example. Except for θ, the parameter values $P^p_{Tx_k}$ for device $k \in [1, N_{Tx}]$ are computed as:

$$P^p_{TX_k} = P^p_{min} + k\frac{(P^p_{max} - P^p_{min})}{N_{Tx}}, \tag{5.11}$$

$$\text{with } P^p_{min} = \text{Mean Value}(1 - p\%), \tag{5.12}$$

$$P^p_{max} = \text{Mean Value}(1 + p\%), \tag{5.13}$$

with P^p_{min} the minimum of the impairment parameter in the p% similarities scenario and P^p_{max} the maximum. Figures 5.3 present an example of repartition value for the CFO impairments at 5% and 3%. In Figure 5.3a the CFO value is between 285 Hz and 315 Hz, while in Figure 5.3b the similarity is more important because the CFO is between 295 Hz and 300 Hz.



(a) CFO at 5%.

(b) CFO at 3%.

Figure 5.3 – Repartition of CFO values around the mean (300Hz), for 6 transmitters and different percentages.

Four different similarity scenarios are studied, 5%, 3%, 2% and 1%. For θ parameters, P^p_{TXk} follows (5.11) but P^p_{min} and P^p_{max} depend of the similarity scenario. For $p = 5\%$ and 3%, we set:

$$P^p_{min} = 0° \qquad P^p_{max} = 5°, \tag{5.14}$$

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

$$P^p_{min} = 1° \qquad P^p_{max} = 4°. \tag{5.15}$$

5.2.1 Preamble scenario

In this section, the preamble mode is studied, which corresponds to sending the same sequence for all transmitters, several times.

A. 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 5.6 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 in Table 5.6 indicate that increasing the state noise variance worsens both classification and generalization challenges due to the additional noise introduced into the signal. At 30 epochs the network has reached 98% in the testing dataset 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 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 study, phase noise is set to 10^{-7}.

F1 score at		30 epochs		60 epochs		280 epochs	
σ^2_ξ	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 5.6 – Mean F1 score evolution during training phase for Preamble and different similarity scenarios, $\gamma = 10^{-4}$, dr = 0, 900 signals per transmitter for train and 100 signals per transmitter for test.

Then different similarity scenarios: 5%, 3%, 2% and 1% are compared. The network has no difficulty in classifying the 6 transmitters in the 5% scenario. As the similarity increases, the network needs more time to learn and classify the devices. The complexity

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%

Table 5.7 – Mean F1 score evolution during training phase for Preamble and different similarity scenarios, $\gamma = 10^{-5}$, dr = 0.25, $\sigma_\xi^2 = 10^{-7}$, 900 signals per transmitter for train and 100 signals per transmitter for test.

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 5.7. 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 only specializes 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 5.4 presents the F1 score obtained in the 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.

In the Preamble scenario, the network specializes on the training data: for another preamble used in the test, the network obtained about 25% of the F1 score 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 well in this situation.

B. 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 sequence is the same

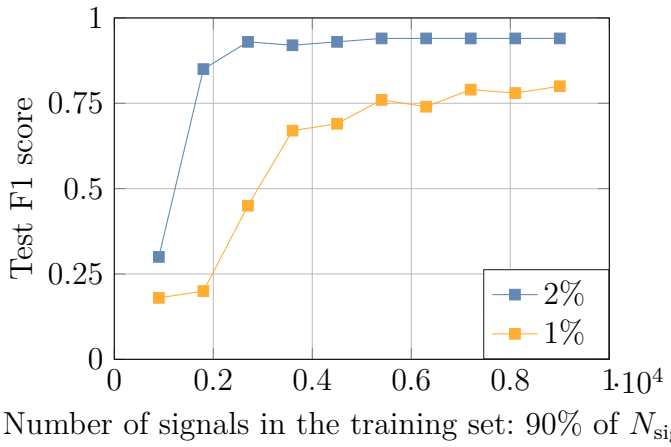


Figure 5.4 – F1 score obtained in test in function of the number of signals used to train the network when training has reached 98% of F1 score, in Preambule scenario with $\gamma = 10^{-4}$ $\mathrm{dr} = 0$.

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

The results presented in Table 5.8 are obtained without dropout and with a learning rate at 10^{-4} with 900 signals in train set. The F1 scores are very close to the results presented in the previous Table 5.6. The convergence speed is comparable to the convergence speed obtained with OFDM and decreases when the similarity between devices increases. 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 PhD.

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

Table 5.8 – Mean F1 score evolution during training phase for Preamble and different similarity scenarios for single carrier modulation, $\gamma = 10^{-4}$, $\mathrm{dr} = 0$, $\sigma_\xi^2 = 10^{-7}$, 900 signals per transmitter for train and 100 signals per transmitter for test.

C. 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 5.9 presents the results obtained in the test when the network has reached 98% of F1 score on the train set for different situations. The best performances are obtained when the PA is set to 10% with 94% of F1 score and reveals the importance of PA in RFF identification.

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

Table 5.9 – F1 score obtained in test when training has reached 98% for different RFF at 10%, $\gamma = 10^{-5}$ dr $= 0.25$.

D. Does the dynamic CFO impact the classification?

The previous result highlights the most relevant features and Table 5.9 shows that the CFO is not really impacting in our context. In Section 4.1, 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 SoA extensively covers this topic and leads to the conclusion that the CFO has a significant impact on RFF identification [28, 122, 24, 44]. 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:

$$\Delta f_{Txi+1} = \Delta f_{Txi} + \delta_f, \tag{5.16}$$

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

Figure 5.5 presents the F1 score obtained in the test when the CFO has shifted between the training phase and test phase for different dispersion scenarios. The other impairments are set at 5% similarity, and the learning parameters are still $dr = 0$ and $\gamma = 10^{-4}$ with 900 signals per transmitter in training set. This shift, called frequency variation, and noted ν_f, can model the impact of a temperature variation and is expressed as:

$$\Delta f_{Test_{Txi}} = \Delta f_{Train_{Txi}} \pm \nu_f, \tag{5.17}$$

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

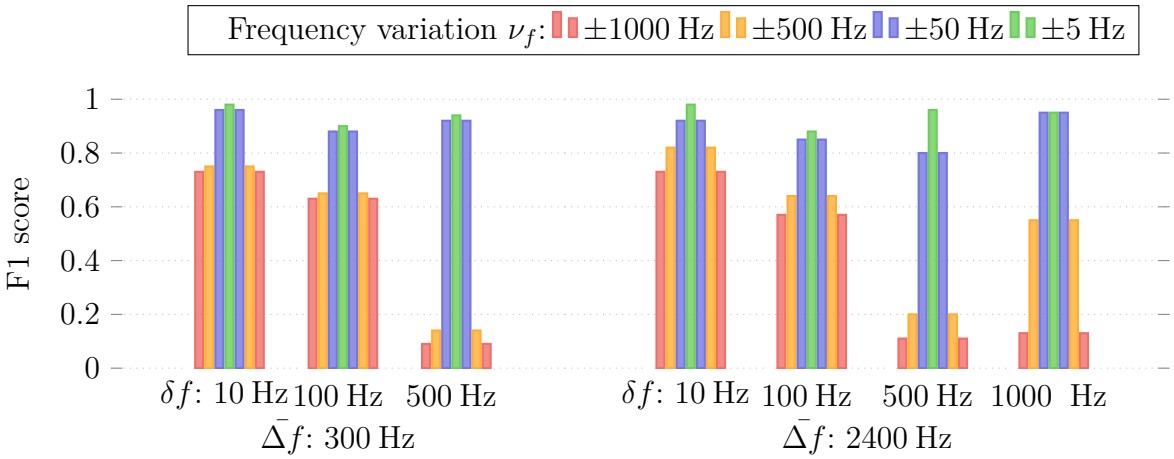


Figure 5.5 – F1 score obtained in test in function of the CFO dispersion values δ_f for two different $\bar{\Delta f}$, and 6 transmitters.

The left part of Figure 5.5 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 dispersion 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, the accuracy drops, meaning 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 at the right part of the figure, the conclusions are the same as 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 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 study, we keep the parameterization of a precise oscillator: $0.13ppm$ (temperature compensated X oscillator, or an oscillator whose frequency is controlled by digital/analog compensation).

E. 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 a 5% interval 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 required more epochs to converge because the complexity of the problem has increased.

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%

Table 5.10 – Mean F1 score during training phase for 5% similarity and 12 devices and 6 devices with $\gamma = 10^{-4}$ dr $= 0$.

Conclusion

— A strong similarity between RFF transmitters increases the network convergence time and decreases the F1 score performance.

— Changing the learning parameters can improve the classification in the test but increasing the number of signals seems the best option.

— The PA is the most relevant impairment for identification.

— For a 500 Hz dispersion of CFO between transmitters, the CFO becomes the most relevant impairment.

— The network classification accuracy is affected by a variation of 1000 Hz, but is still around 75% of F1 Score, for 100 Hz dispersion between transmitters.

5.2.2 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 5.11a presents the result of classification in the test without MAC spoofing. The rows of the confusion matrix are the true labels while the columns are the labels estimated by the network. The numbers represent the percentage obtained for each case. Table 5.11b 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 learns 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 [48].

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	100	0	0	0	0	0
Tx_2	8	87	2	1	1	1
Tx_3	0	0	100	0	0	0
Tx_4	0	0	0	100	0	0
Tx_5	0	0	0	0	100	0
Tx_6	0	0	0	0	0	100

(a) Without spoofing.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	0	0	100	0	0	0
Tx_2	8	87	2	1	1	1
Tx_3	0	0	100	0	0	0
Tx_4	0	0	0	100	0	0
Tx_5	0	0	0	0	100	0
Tx_6	0	0	0	0	0	100

(b) Tx1 spoofed MAC address from Tx3.

Table 5.11 – Confusion Matrix for test data in MAC scenario.

5.2.3 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.

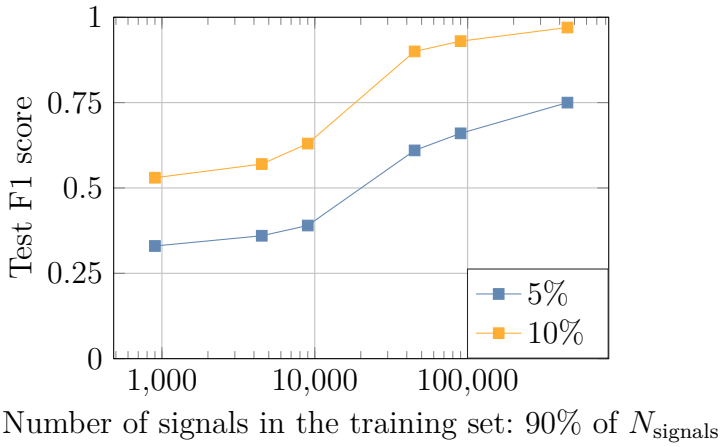


Figure 5.6 – F1 score obtained in test in function of the number of signals used to train the network when training has reached 98% of F1 score.

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 5.6 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

Number of signals	5%	10%
900	27s	25s
4,500	160s	110s
9,000	8min	220s
45,000	45min	19min
90,000	2h	38min
450,000	+10h	2h12

Table 5.12 – Time required for the network to reach 98% of F1 score on training data, in Payload context.

similarity scenarios. In blue, we represent the 5% similarity scenario database, and yellow represents the 10% similarity scenario. Table 5.12 completes the results by adding the time of training to reach 98% of the F1 score.

Figure 5.6 shows a great improvement of the F1 score in the test when the number of signals is increased. However, Table 5.12 presents the time required to achieve the different training and the time convergence speed increase when the number of signals increases 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 amount of data may be limited by the difficulty of collecting data. The time to train the network may be limited by the need for a short response time.

5.3 Network channel resilience study

SoA motivations

Chapter 3 reveals the impact of propagation channel in classification accuracy and the need to have a channel resilient database to train the network to recognise the impairments. In the SoA, several works have shown the need for data augmentation to improve the resilience of the propagation channel [102, 92, 66, 13]. For example, Morin et al. [66] showed that incorporating varied multipath channel parameters in the training dataset improved network resilience in different transmission scenarios by up to 44% compared to static training, referred to as "plain" in their experiments. However, data augmentation requires a long process to record signals in varied transmission conditions to build

the training dataset. Therefore, we use the RiFyFi virtual database generator to investigate the network robustness, and enable comprehensive monitoring of all aspects of the experimental setup.

5.3.1 Impact of propagation channel on the classification accuracy in Preamble mode

A. Network resilience evaluation

To evaluate the resilience of the network under different conditions, multiple datasets, called scenarios, are required to test the network under different environmental conditions compared to the training one. The resilience of a DL network is evaluate thank the following process :

— 1. Creating a database (model-based or experimental) following a propagation channel scenario S_{train};

— 2. Using this database to train the DL network;

— 3. Creating a new database with another scenario S_{res};

— 4. Using this new database to evaluate the resilience of the DL network in the scenario S_{res}.

Different crucial aspects of channel impact are studied in Preamble context to create an RFF database: the number of signals required to perform DL classification, the impact of RFF similarities between emitters, and the propagation channel impact. In this section, all the experiments are done with 5 transmitters, the learning rate is set at $\gamma = 10^{-5}$, and the dropout at 0.25.

The propagation channel considered models a wireless flat-fading transmission over a path with random power,and delay spreads powers are generated. Due to the power variations, we have to considered a new normalisation of the dataset. In the previous result without considering the propagation channel, the entire dataset was used to calculate the mean and variance, which were then applied for normalization. However in this way the network use principally the amplitude to classify the transmitters. In this section we propose to normalize the data by group of few consecutives sequences, that have the same propagation channel, means that each group of sequences of each transmitters has the same mean value at the end because all sequence are normalised following his own statistics. This new normalisation may have an impact on the network training/classification.

First of all, we consider two scenarios: scenario S_1 for S_{train}, which models a wired transmission, is used to train the network and evaluate its static performance, and then scenario S_2 for S_{res} models a wireless flat-fading transmission over a path with random power, and several channel powers are generated. The training is done with data from S_1 and the test is realize on both scenario S_1 and S_2. The S_2 set is composed of 100 signals per device and per channel, and we simulate 100 different channels for each device. Table 5.13 present the F1-Score obtained in the different similarities for both testing scenrario. The results show that the network could not identify the transmitter in S_2 case. Moreover, because of this new normalisation, we have to significanthly increase the number of signal in the training database to avoid overfitting and obtained around 90% of F1-Score on Test S_1.

	Training on S_1		
p%	$N_{sig.}$	Test S_1	Test S_2
10%	10000	98%	27%
7%	10000	92%	24%
5%	10000	91%	23%
3%	10000	89%	23%

Table 5.13 – F1-Score obtained in different similarity scenarios to evaluate the resilience, Preamble mode.

B. How many channel should we have?

In this section, we apply data augmentation during the training step by generating propagation channel diversity. The database is extended by adding signals with impairments by different numbers of channels. The objective is to evaluate/estimate how many different propagation channels are required in the training database to ensure the channel resilience of the network in different environment.

The network is trained with a dataset composed of $N_{chan.} \times N_{Tx} \times N_{signals}$, where $N_{signals} = 1800$, $N_{Tx} = 5$. Then the F1-Score is evaluated on a dataset composed of 100 different channels with 200 signals. In this experiment, the network learning rate is set to 10^{-5}, the dropout dr to 0.25 and the batch size to 64.

Figure 5.7 presents the F1-Score obtained in S_2 Test for different level of channel diversity, in this case the channel diversity is important as $N_{chan.}$ is large. The results

obtained show that data augmentation allow to increase the performances accuracy obtained in different propagation channels conditions. Moreover, different floors appear for each similarity scenario, after 100 channels per transmitters means that over 100 channels per transmitter are required. Right part of Table 5.14 gives the F1-Score obtained by mean 5 seeds of data generation and training for 100 propagation channels for different similarity scenario.

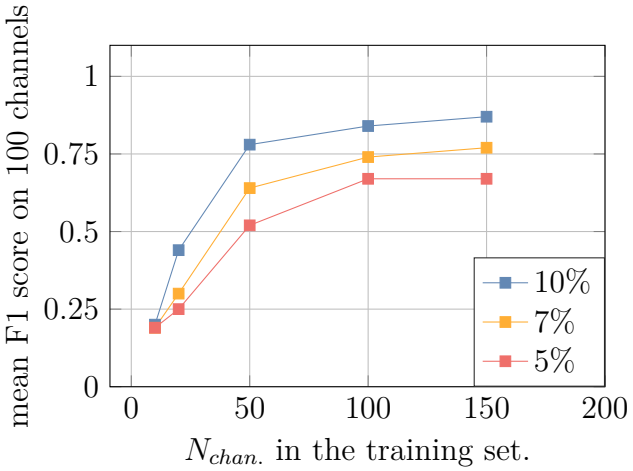


Figure 5.7 – Mean F1 score obtained in test in function of the number of channels used to train the network when training has reached 98% of F1 score.

	Training on S_1			Training on S_2		
p%	$N_{sig.}$	Test S_1	Test S_2	$N_{sig.}$	$N_{chan.}$	Test S_2
10%	10000	98%	27%	1800	100	84,5± 0.8%
7%	10000	92%	24%	1800	100	74.8± 0.6%
5%	10000	91%	23%	1800	100	67.6±0.2%
3%	10000	89%	23%	2700	100	69.2±0.3%

Table 5.14 – F1-Score obtained in different training and similarity scenarios to evaluate the resilience, Preamble mode.

These simulations show that over 100 channels per transmitter are required to ensure intersting performance resiliency, meaning capturing signals from each transmitter at 100 different locations, which is a too long and complex process for many RFF identification applications. For instance, let's consider that changing the propagation channel context

for a transmitter takes just 5 seconds. This means a new record with a different propagation channel is created every 5 seconds. Collecting sufficient data for a single transmitter requires over 500 seconds (approximately 8 minutes). However, if the process isn't automated and manual context switching takes a full minute, the time required per transmitter would exceed 1 hour and 30 minutes or even more.

5.3.2 Diversifying data to ensure robustness and resilience

A. Methodology

We propose to change the paradigm of data augmentation to reduce the complexity and time required to build a training database while ensuring network resilience. The channel diversity used in traditional data augmentation generates entropy, which helps the network to be resilient to environmental influences. The top of Figure 5.8 presents conventional data augmentation, used in the previous section, with a fixed data preamble and channel diversity. In our approach, diversity is directly introduced into the transmitted data, increasing the entropy of the signals used for identification. The bottom of Figure 5.8 shows our solution with an ideal propagation channel, such as a wired one, which simplifies the process of recording signals from different transmitters with sufficient diversity.

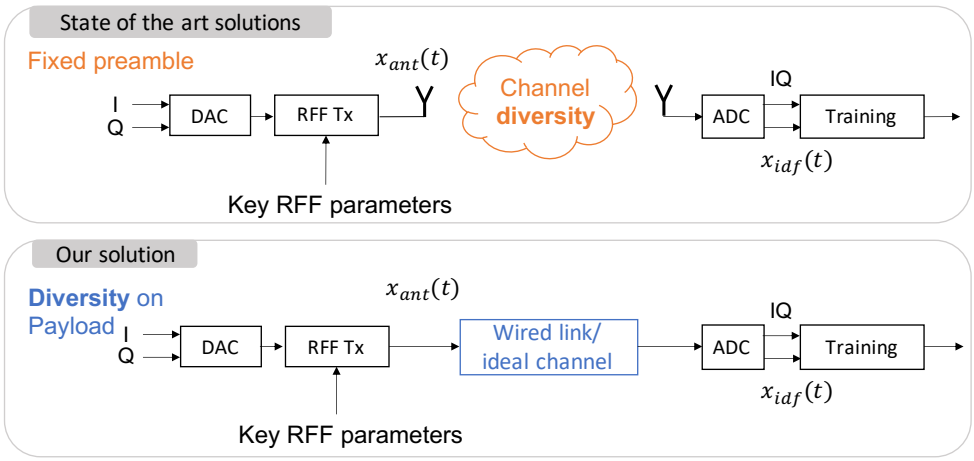


Figure 5.8 – Increase entropy of training dataset by adding diversity in transmitted data.

In practice, long sequences of random symbols are generated and transmitted and are thus disturbed by RF impairments. This transmission mode, called payload, is used to train the network to recognise the RFF in a noisy and diverse environment. To present a proof of concept, experiments are firstly conducted using model-based databases. These

databases are created according to the previously described parameters, including different similarity scenarios and a payload mode. Two transmission scenarios are used: S_1 for the wired transmission model, and S_2 for the wireless propagation model with multiple paths and noise levels.

B. Simulation-based proof of concept with data diversification

Three large databases with 10%, 7% and 5% similarity have been created, each consisting of random OFDM symbols to obtain 180,000 signals of 256 IQ samples. Each dataset are normalized following the new normalisation by group of sequences. The network is trained with each dataset in S_1 scenario using a learning rate of 10^{-5} and a dropout dr=0.25. After training, the networks are evaluated in both scenarios S_1 and S_2. Table 5.15 shows the network F1-score following both scenarios. We chose to create 180,000 signals to match the diversity achieved with 1,800 signals and 100 channels per signal, as in traditional data augmentation.

	Training on S_1		
p%	$N_{sig.}$	Test S_1	Test S_2
10%	180000	77%	59%
7%	180000	60%	47%
5%	180000	45%	39%

Table 5.15 – F1-Score obtained in different training and similarity scenarios, Payload mode.

The results show that a very large dataset is indeed necessary to perform identification. However, only one record per transmitter is required, which is a great simplification compared to the usual data augmentation, where 100 recordings per transmitter are typically required. Using this approach, recording 180,000 signals per transmitter takes approximately 30 seconds, making it at least 16 times faster (assuming an optimistic recording time of every 5 seconds) than the previously described method. Table 5.16 shows the confusion matrix obtained in test in scenario S_2 for a 10% similarity. The network occasionally confuses transmitters, in particular each transmitter i are often confused with the $i-1$ and $i+1$ due to the close proximity in the impairment definition.

In the next chapter some experimental databases are created to validate the solution proposed in this section.

Guess / True	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	69.7	25.4	3.0	0.5	1.4
$TxTrue_2$	17.9	51.0	25.5	3.8	1.8
$TxTrue_3$	2.4	17.6	53.4	22.1	4.4
$TxTrue_4$	1.3	1.7	18.1	49.8	29.1
$TxTrue_5$	1.1	0.4	2.5	22.0	74.0

Table 5.16 – Confusion Matrix obtained for 10% similarity scenario in test S_2, Payload mode.

5.4 Conclusion

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.

In Sections 5.1, and 5.2.1, our analysis showed in preamble context:

— a strong similarity between RFF transmitters increases the network convergence time and decreases the F1 score performance,

— changing the learning parameters can improve the classification in the test but increasing the number of signals seems the best option,

— the PA is the most relevant impairment for identification.

In Section 5.2.1, our analysis of the CFO showed

— for a 500 Hz dispersion of CFO between transmitters, the CFO becomes the most relevant impairment,

— the network classification accuracy is affected by a variation of 1000 Hz, but is still around 75% of F1 score, for 100 Hz dispersion between transmitters.

In Section 5.2.3 our analysis of the Payload scenario showed

— a large number of signals in the Payload scenario can mitigate the issue of the propagation channel.

In Section 5.3.1 our analysis of the propagation channel showed

— the channel presence can deteriorate significantly the classification accuracy,

— a large number of signals with propagation channel diversity are required which is time consuming,

— similar RFF devices can be a countermeasure to avoid RFF identification.

In Section 5.3.2 our proposition to change the data augmentation paradigm showed

— the improvement of channel resilience by increasing data diversity instead of channel diversity,

— the reduction of time required to produce the real database with enough diversity.

RiFyFi_VDG can help to pre-evaluate the required database design with a lot of flexibility. This generator is an open source tool available in [11]. These works have been published in [12, 14].

Chapter 6

From Virtual Data to Real Data

In this chapter, we propose to create our own datasets based on the lessons learned from our experience with the generator. The objective of this part is to validate the previous conclusions made thanks to transmitters model-based dataset. Section 6.1 presents the experimental setup for the different scenarios of datasets used in this chapter. Then Section 6.3 presents and discusses the results obtained with Preamble mode, and Section 6.4 presents the Payload mode. Finally, Section 6.5 proposes some perspectives of this work to improve the classification accuracy.

6.1 Experimental scenarios description

The lessons of the previous chapter highlight the need for numerous signals when the similarity between transmitters is important and the best resilience of the payload mode in different scenarios. The goal is to verify both points. First of all, five SDRs are used as transmitters, and one, always the same, is used as receiver. Table 6.1 presents the name of the transmitter and the references with the label considered in the datasets. The transmitters are all different excepted the two E310, however for one of them the use of GPS has been desactivated so the CFO is probably less stable compared to the other one. The power emission of each transmitter is adjusted to obtain the same power in reception. This step ensures that the network will not focus on the power difference which changes depending on the conditions. We decide to create different datasets to evaluate the capacity of the network to recognize the transmitters in a real world proof of concept. For each transmission scenario, we have two different modes: the Preamble and the Payload.

In the Preamble scenario the transmitters all transmit the same sequence of 30 OFDM symbols, and repeat this sequence for 8 seconds. For Payload mode, each transmitter transmits random OFDM symbols for 30 seconds. The time defined for Preamble and

Label	Name of transmitter	Constructor	RF daughter board
Tx1	Blade RF	Nuand	AD9361
Tx2	ADALM Pluto	Analog device	AD9363
Tx3	E310 (no gps)	ETTUS	AD9361
Tx4	E310	ETTUS	AD9361
Tx5	X310	ETTUS	UBX-160

Table 6.1 – Transmitters used to create the experimental dataset.

Payload is different because thanks to the generator we know that we need more data to train a network with Payload mode.

Then we define five different scenarios, presented in Table 6.2:

— S1, the transmitters are sending the data through the same wire, same attenuator for each radio, and transmitters are assumed cold (i.e the transmitters are turn on just to perform the transmission). This scenario will be used to train the network, all remaining scenarios are only used to test resilience.

— S1bis, the scenario 1 has been repeated another day to ensure the reproducibility of the ideal scenario.

— S2, same as S1 but the transmitters are turned on in the morning and the recording is done in the afternoon so the components of the transmitters are considered hot.

— S3, the transmission is done over the air (OTA) in anechoïc chamber and all the transmitters are placed the same location for the transmission.

— S4, same as S3 but the transmitters have different locations.

— S5, over-the-air scenario in an office with different locations.

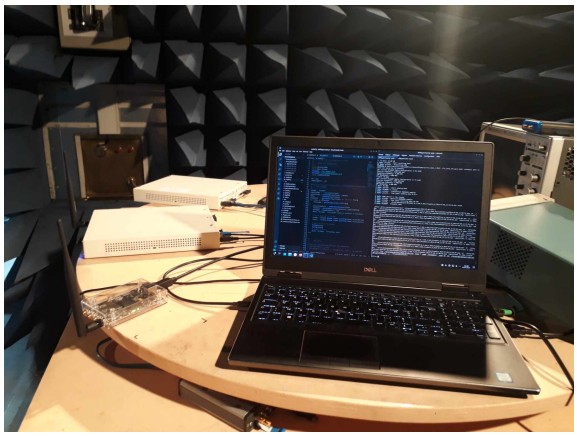
— S6, a room with several metallic objects which can potentially increase the effect of the propagation channel.

6.2 Experimental overview

The network used in this chapter is the Sankhe_2020. The network is trained with data from scenario 1, different training are done with different sizes of datasets. From 9,000 signals per transmitter to 45,000 for Preamble mode, and from 9,000 signals per

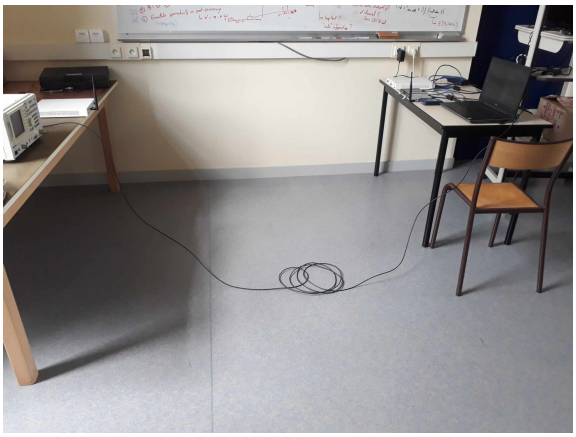
Train/test	Scenario	Channel	hot/cold	Tx location
Train	1	Wired	cold	-
Test	1bis	Wired	cold	-
Test	2	Wired	hot	-
Test	3	OTA Anechoic chamber	cold	same place
Test	4	OTA Anechoic chamber	cold	different locations
Test	5	OTA office	cold	different locations
Test	6	OTA metallic room	cold	different locations

Table 6.2 – Experimental scenarios description.



(a) Scenario 4.

(b) Scenario 5.



(c) Scenario 6.

Figure 6.1 – Photos of experimental setup for different scenarios.

transmitter to 180,000 for Payload mode. The training ends when the network reaches 98% of the F1 score on training data. Then each network is evaluated through the testing dataset of each scenario. The testing datasets are composed of 5,000 signals per transmitter, and correspond to the end of the recorded signals as shown in Figures 6.2 and 6.3. The learning parameters are set to $\gamma = 10^{-4}$ for the learning rate, dr = 0.5 for the dropout, and the batch size is set to 64.

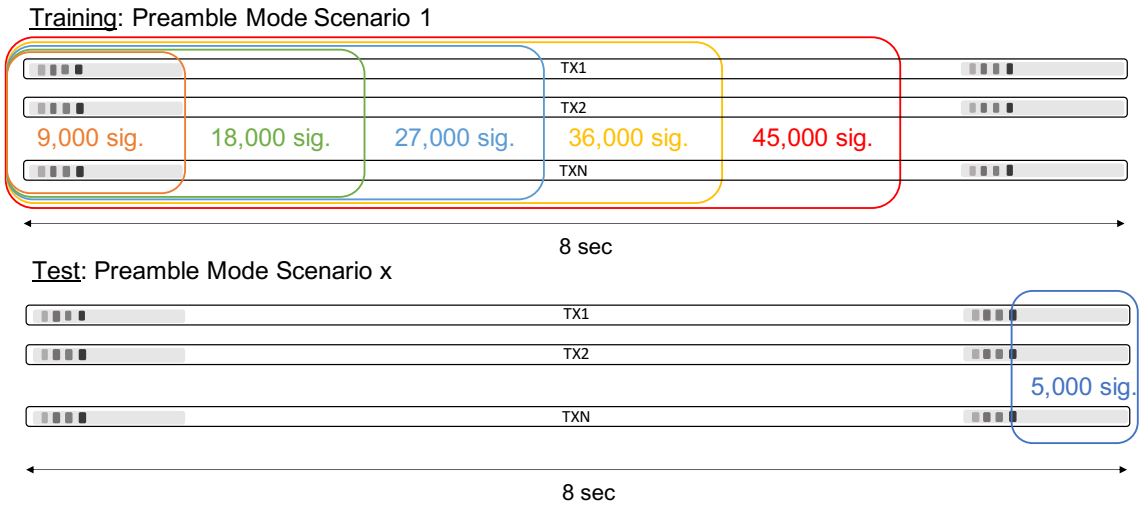


Figure 6.2 – Training and Test dataset repartition on capturing signal for Preamble mode.

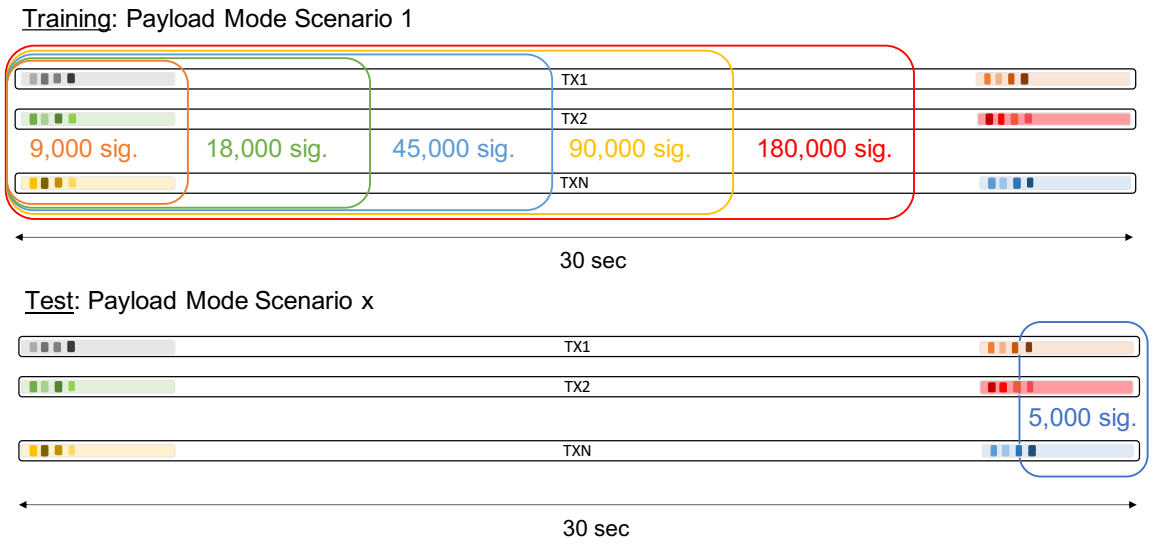


Figure 6.3 – Training and Test dataset repartition on capturing signal for Payload mode.

6.3 Experimental results in Preamble mode

Table 6.2 presents the mean F1 score obtained on 3 seeds of the network for each scenario depending on the size of the training dataset. The column 1 corresponds to the results obtained with the test part of scenario 1, and column 1bis corresponds to a second recording done a few days after to ensure the reproducibility of the ideal case. Columns 1 and 1bis show that increasing the number of signals increases the F1 score obtained in the test in both cases.

The comparison of the F1 Score obtained with scenario 1 or 1bis and scenario 2 shows a drop in classification performance. Using the SDR on or off has thus an impact on the performance. However, this drop is reduced by increasing the number of signals in training up to 36,000.

	F1 score obtained for testing scenarios						
Number of signals	1	1bis	2	3	4	5	6
9,000	76.7%	89.3%	74.5%	43.7%	40.0%	27.0%	20.1%
18,000	95.8%	96.4%	86.7%	41.0%	40.0%	20.3%	25.0%
27,000	97.9%	97.2%	90.0%	34.5%	40.0%	19.8%	24.0%
36,000	98.4%	97.5%	92.2%	40.8%	40.0%	20.0%	26.7%
45,000	97.8%	97.3%	89.3%	41.5%	40.0%	24.9%	20.0%

Table 6.3 – F1 Score obtained in Test for each scenario in Preamble mode, depending on the number of signals used to train the network.

The network evaluation on scenarios 3 to 6 shows bad performance, and in particular, the performance decreases with the scenario difficulty. However, this result was expected because the network was trained with an ideal scenario, and adding a propagation channel in the test affected the classification. We propose to show one confusion matrix per scenario to understand the network behavior.

Tables 6.4 present the confusion matrices obtained in the test for each scenario. Tables 6.4a and 6.4b reveal great performance with wired transmission. In scenario 2, transmitter 1 is sometimes confused with transmitter 3. Tables 6.4c and 6.4d show a different behavior: the transmitters 2 and 5, respectively the Pluto and the x310 are perfectly recognized but the other transmitters are confused, the confusion is not the same for both scenarios which means that a particular effect of channel or power affected the recogni-

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	93.5	0.0	6.4	0.0	0.1
$TxTrue_2$	0.0	99.9	0.0	0.1	0.0
$TxTrue_3$	1.9	0.0	97.6	0.5	0.0
$TxTrue_4$	0.0	0.1	0.3	99.6	0.0
$TxTrue_5$	1.9	0.0	0.0	0.0	98.1

(a) Scenario 1

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	71.1	0.0	28.9	0.0	0.0
$TxTrue_2$	0.0	92.0	0.0	8.0	0.0
$TxTrue_3$	2.9	0.0	96.5	0.6	0.0
$TxTrue_4$	0.0	5.8	0.0	94.2	0.0
$TxTrue_5$	0.3	0.0	0.0	0.0	99.7

(b) Scenario 2

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.0	13.7	0.2	86.1	0.0
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	92.9	0.0	7.1	0.0	0.1
$TxTrue_4$	7.1	0.0	92.3	0.5	0.0
$TxTrue_5$	0.0	0.0	0.0	0.0	100.0

(c) Scenario 3

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.6	0.0	0.0	0.0	99.4
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	61.7	0.0	0.0	0.0	38.3
$TxTrue_4$	0.0	100.0	0.0	0.0	0.0
$TxTrue_5$	0.0	0.0	0.0	0.0	100.0

(d) Scenario 4

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.0	99.6	0.2	0.1	0.0
$TxTrue_2$	71.7	0.0	0.1	0.0	28.2
$TxTrue_3$	0.7	0.0	33.7	65.6	0.0
$TxTrue_4$	0.4	97.9	1.2	0.5	0.0
$TxTrue_5$	0.9	0.0	0.0	0.0	99.1

(e) Scenario 5

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.0	100.0	0.0	0.0	0.0
$TxTrue_2$	0.0	99.7	0.2	0.1	0.0
$TxTrue_3$	0.0	78.1	0.6	21.3	0.0
$TxTrue_4$	0.8	0.0	0.0	0.0	99.2
$TxTrue_5$	0.0	99.3	0.4	0.3	0.0

(f) Scenario 6

Table 6.4 – Confusion Matrices obtained for each scenario with a network trained with 50,000 signals per transmitter, from scenario 1 in Preamble mode.

tion. Finally Table 6.4e is really difficult to interpret, but Table 6.4f all transmitters are classified as Tx2 excepted the transmitter 4.

6.4 Experimental results in Payload mode

Table 6.5 presents the results obtained for Payload mode for each scenario depending on the size of the training dataset, and Tables 6.6 present the corresponding confusion matrices. The column 1of Table 6.5 corresponds to the results obtained with the test part of scenario 1. Compared to the Preamble mode, the Payload mode requires more signals to reach the same accuracy, which is coherent with RiFyFi conclusion. The behavior obtained in columns 1, 1bis and 2 are really similar to the Preamble mode. In particular the results for the scenario 2 are improved means that the diversity ensure the robustness of the identification. The scenarios 3 to 5 obtain better results in Payload mode than the results obtained with Preamble mode, which means that increasing the diversity of the signals increases the robustness of the network. Reaching 50% of F1-score is not totally satisfying however regarding the confusion matrices presented in Table 6.6e for scenario 5, we notice that some transmitters are perfectly to correctly recognize aand some other are not. In future experiments it could be interested to classify different types of devices. The others confusion matrices whow a correct recognition of different devices in particular for the scenario 1 and 2.

	F1 score obtained for testing scenarios						
Number of signals	1	1 bis	2	3	4	5	6
9,000	42.4%	47.8%	56.9%	37.8%	36.9%	22.3%	38.6%
18,000	75.4%	73%	78.5%	53.2%	45.6%	42.2%	37.3%
45,000	91.1%	89%	88.7%	57.6%	54.9%	48.4%	22.4%
90,000	94.8%	92.9%	93.3%	59.6%	53.5%	50.8%	9%
180,000	96.4%	94.5%	95.2%	60.46%	53.5%	50.4%	14.5%

Table 6.5 – F1 Score obtained in Test for each scenario in Payload mode, depending on the number of signals used to train the network.

However, these experiments highlight the need to pre-process the data to improve the quality of the signal and so the classification accuracy in particulary in scenario 6. In scenario 6 the received power is largely reduced and disturbe the network classification.

Contrary to the most approaches of the SoA, in this PhD channel equalization is not a wanted solution because it requires data knowledge that we do not want to depend on.

6.5 Conclusion and experimental perspectives

This study confirmes the previous conclusion done thanks to the generator RiFyFi_VDG, with a drop in performance probably due to the important similarities between the devices. We have proposed an alternative and less complex approach for creating RFF DL identification training databases. Our solution shifts the focus from channel diversity to data diversity to enhance network resilience to varied propagation environments. The results and analysis show significant performance improvements compared to similar Preamble data recording methods. In addition, our solution is at least 16 times faster than SoA data augmentation methods. While this solution alone does not fully address all identification needs, it reliably distinguishes between different types of transmitters, making it particularly useful in applications where precise recognition is less critical and identifying the transmitter type is sufficient. The datasets created for these experiments allow us to evaluate the level of resilience of the networks depending on the different scenarios, which is not offered by the SoA databases with this level of granularity. In a noisy context with a drop in received power, it is important to note that the signal quality is deteriorating and needs to be improved to help the network to recognize the transmitter. That is why future work can be done to find a pre-processing to filtrate the signal without deteriorating the RFF. Finally, in future works, it will be interesting to use the preamble mode and estimate the impairments of our SDRs. Then the impairments will be used as parameters of RiFyFi_VDG. This will allow us to implement a digital twin of each device and then train a network with a virtual database in payload mode. Then the network capabilities are evaluated to identify the real devices.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	93.8	0.0	0.5	5.7	0.0
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	0.7	0.0	97.3	2.0	0.0
$TxTrue_4$	5.9	0.0	2.3	91.7	0.2
$TxTrue_5$	0.1	0.0	0.0	0.3	99.7

(a) Scenario 1

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	94.7	0.0	0.7	4.5	0.1
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	0.5	0.0	97.7	1.7	0.0
$TxTrue_4$	9.5	0.0	3.1	87.3	0.1
$TxTrue_5$	0.2	0.0	0.0	3.3	96.5

(b) Scenario 2

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	53.9	0.2	0.4	17.4	28.1
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	3.7	0.0	92.7	3.5	0.0
$TxTrue_4$	23.1	0.0	30.9	45.9	0.1
$TxTrue_5$	42.9	0.0	9.8	26.3	21.0

(c) Scenario 3

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	46.7	0.0	53.3	0.0	0.0
$TxTrue_2$	0.0	100.0	0.0	0.0	0.0
$TxTrue_3$	0.0	0.0	99.7	0.2	0.0
$TxTrue_4$	0.0	100.0	0.0	0.0	0.0
$TxTrue_5$	39.4	0.0	5.3	31.6	23.7

(d) Scenario 4

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.0	46.3	0.1	11.1	42.5
$TxTrue_2$	1.9	90.8	0.2	6.8	0.2
$TxTrue_3$	5.0	0.0	79.8	15.1	0.1
$TxTrue_4$	0.7	0.0	1.4	61.8	36.1
$TxTrue_5$	36.5	0.0	6.7	35.7	21.1

(e) Scenario 5

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5
$TxTrue_1$	0.0	0.1	0.0	99.9	0.0
$TxTrue_2$	0.1	1.3	0.0	98.6	0.0
$TxTrue_3$	0.8	0.0	2.4	96.9	0.0
$TxTrue_4$	3.8	0.0	35.4	60.8	0.0
$TxTrue_5$	0.0	0.3	0.0	99.6	0.0

(f) Scenario 6

Table 6.6 – Confusion Matrix obtained for each scenario with a network trained with 180,000 signals per transmitter from scenario 1 in Payload mode.

Chapter 7

Comparison of Machine Learning for lightweight RFF identification

If DL techniques are promising and show very good classification results, they also exhibit an important complexity both at training and inference steps, dependent on their architecture [124]. In the IoT context, RFF authentication might be with stringent complexity and energy constraints. This chapter presents works realized during the PhD and has been done with the help of student internships, Baptiste BOYER and Emma BOTHEREAU, and addresses the complexity issues. The common thread of these studies is lightweight RFF identification. Section 7.1 presents the lightweight RFF identification motivations. The first study concerns the Tangled Programm Graph. In Section 7.2 we propose to use TPG-based classification to achieve a lightweight and accurate RFF identification scheme. This study has been published in a publication at PIMRC 2023 [13]. The second study presented in section 7.3 concerns pruning applied to classic neural networks. The results of this study led to the beginning of a new PhD for 2023-2026 by Emma BOTHEREAU.

7.1 Lightweight RFF Identification motivations

The previous chapters are mainly focused on DL and they promise and show very good classification results. Nonetheless, they exhibit an important complexity both at the training and the inference steps, depending on their architecture [124]. In the IoT context, RFF authentication might be with stringent complexity and energy constraints. Therefore, the lightweight RFF identification is an interesting axe of this PhD. The lightweight identification is a vast subject and different levers can be used to improve the embedded characteristic. For example, to obtain the result of previous chapters, a GPU has been used. However, in most IoT contexts the devices can only embed a CPU. Therefore having an identification solution working on a CPU can be interesting. For DL techniques, it is

possible to realize the training part on GPU and then use a CPU for inference. However, in a dynamic context where transmitters appear or disappear, we can imagine that having a retrainable network in real-time can improve the accuracy of classification. That is why having a light network to implement it on a CPU is needed. The next section presents the second major contribution of this PhD, which addresses the complexity issues by proposing to use TPG instead of CNN for the RFF identification.

7.2 Efficient RFF Identification with Tangled Program Graph

The lightweight TPGs are a recent light-by-construction ML technique based on genetic programming principles [101]. Previous works demonstrated that for comparable performance with a SoA of DL, TPGs inference required 2 to 3 orders of magnitude less computations complexity, and 3 to 5 orders of magnitude less memory [50].

7.2.1 A brief introduction of TPG

Introduced in 2017, TPG is a successful Reinforcement Learning (RL) model [51] that is built on SoA genetic programming techniques. Unlike neural networks whose topology is generally chosen by an expert data scientist, TPGs are grown from scratch for each learning environment, and their topology and computational complexity adapt automatically to the complexity of the learned task. TPGs have proven to be competitive with SoA neural networks, providing several order of magnitude improvements in computational complexity and memory requirements on various use cases, with gains at both training and inference [50].

RL, presented in Figure 7.1, is based on an agent that observes the environment and makes actions to change the state of the environment. As these actions change the environment, the agent will propose other actions based on a reward and new observations. A TPG is structured as a directed graph whose vertices and edges, called teams and programs, respectively, specify a control flow of an RL agent, and not a data flow as in neural networks. The control flow of the TPG stems from its root vertex, each time a new state of the learning environment is observed. All programs associated with outgoing edges of the root team are executed with the current state of the environment as their input. An example of TPG and the semantics is given in Figure 7.2.

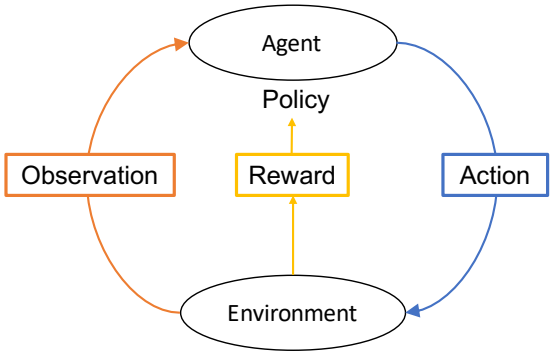


Figure 7.1 – Reinforcement learning principle.

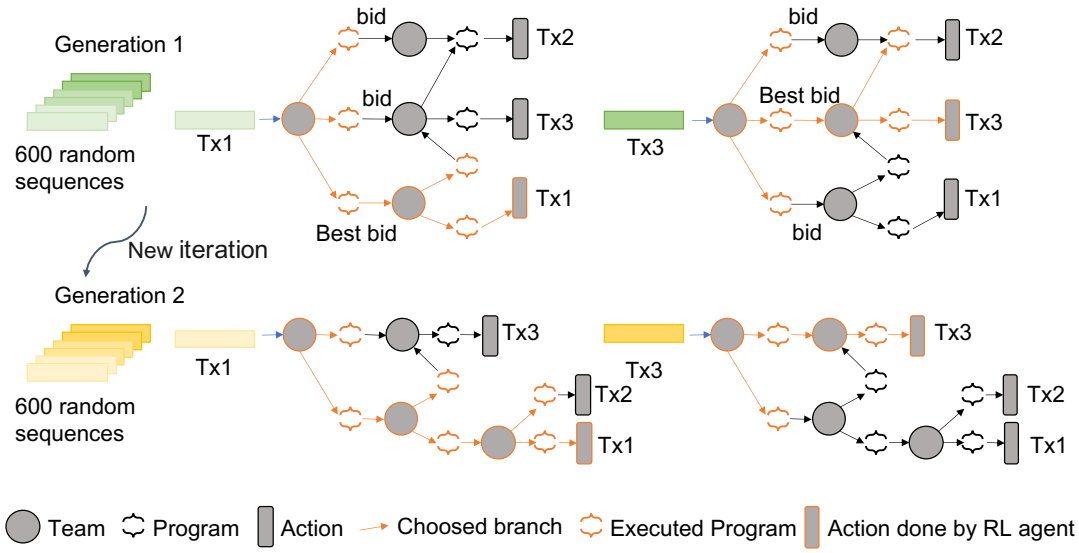


Figure 7.2 – Semantics of the TPGs.

A program is a genetically evolved assembly sequence of instructions taking as inputs the different variables exposed by the environment, here the IQ samples, and returning a single value, called a bid, per program. The programs are composed of basic instructions such as the addition and the multiplication of time domain signals to analyze. Once all programs have completed their execution, the edge associated with the largest output bid is identified, and the execution of the TPG continues following this edge. Eventually, the edge with the largest bid leads to a leaf vertex, associated with a specific action of the RL agent on the environment. After this action, the RL agent can observe again the environment.

The training process for TPG is not based on gradient descent, like DNNs, but on a genetic algorithm. The genetic algorithm is a bio-inspired optimization algorithm, based

on a randomly generated population of learning agents. An initial graph is randomly created with different roots where each root represents a different policy, here a policy corresponds to a classification. Each learning agent observes the environment and takes an action following the above description. Then, after objective evaluations which affect some rewards to each individual policy, the algorithm selects the roots associated with the greatest rewards and removes the other ones from the graph. The best ones are used to create a new population of root teams, which are introduced in the graph by randomly copying and mutating surviving ones. This new population corresponds to a new generation in Figure 7.2.

The TPG is grown from scratch, along all generations, for each learning environment, and complexity is added to the model if it leads to a greater reward. This makes the complexity of the TPGs dependent on the complexity of the learned task [52, 51].

7.2.2 TPGs for RFF classification

Despite being initially proposed for RL, TPGs are also used for classification. In this case, an Action represents a class membership decision. For example, TPG-based classification applied on the CIFAR-10 dataset achieves interesting results [101].

TPG-based classification leads to a similar framework as the one described in Figure 2.18. The network is the TPG, the update phase is done by a genetic algorithm and an iteration corresponds to a generation. The inputs of TPG for each prediction are a set of 256 IQ samples as with DL. At each generation, each root of the TPG takes 600 random sequences of 256 IQ samples in input. The reward is based on the F1 score which is calculated on the 600 sequences.

The genetic update of TPG could create a solution where one class is not classified [101]. Because the global accuracy or F1 score is given as a reward, this may hide disparities between classes, with a class being perfectly detected all the time, and another never. That is why the TPG update, in the classification case, changes to conserve at least one subgraph per class [22]. In this work, we use the Gegelati tool to implement TPG [22]. In the implementation used throughout our experiments, the natural selection process has been modified as follows. When selecting the n best roots that survive for the next generation of the training, $p\%$ of the roots are selected based on their averaged F1 score on all m classes, while the other $(100-p)\%$ are selected for their F1 score on a single class. In this work, $p = 10\%$ is used.

7.2.3 Timing and accuracy comparison in a favorable scenario

In this section, the performance of both TPG and CNN are compared. Both algorithms are trained on the CPU of a core Intel i7-8850H @2.60GHz with 6 cores and 12 threads and with SSE4.2 and AVX2 extensions. The CNN is also trained on a GPU NVIDIA Quadro P1000. The TPG is not implemented on the GPU as its non-symmetric structure is not suitable for such architecture. The WiSig database, introduced in Chapter 3, offers many degrees of freedom such as: the day of capture data, and receiver (positions and references). In this experiment, data are received on day 1 by the receiver Rx1 for both the training and test phases. It corresponds to a favorable scenario for training and identification because the receiver is the same for all signals and the relative position is different for all transmitters. Moreover, the testing scenario is the same as the learning one and the data are equalized.

Tables 7.1a and 7.1b give the confusion matrices obtained with the TPG [22] and the Sankhe_2020 CNN presented in Chapter 3, respectively. The rows of the confusion matrix are the true labels while the columns are the labels estimated by the network. The numbers represent the percentage obtained for each case. Those matrices show the capacity of TPG to learn radio labels correctly. Table 7.1a allows to validate the correct functioning of TPG. Both confusion matrices are very similar.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	96	1	0	1	0	2
Tx_2	0	93	0	7	0	0
Tx_3	0	3	95	0	0	2
Tx_4	1	3	0	96	0	0
Tx_5	0	0	1	0	99	0
Tx_6	0	0	0	0	0	100

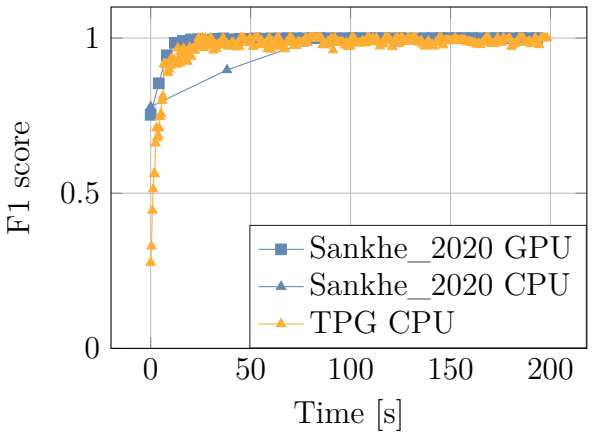
(a) TPG

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	100	0	0	0	0	0
Tx_2	0	87.5	0	12.5	0	0
Tx_3	0	30	70	0	0	0
Tx_4	0	0	0	100	0	0
Tx_5	12.5	0	0	0	87.5	0
Tx_6	0	0	0	0	0	100

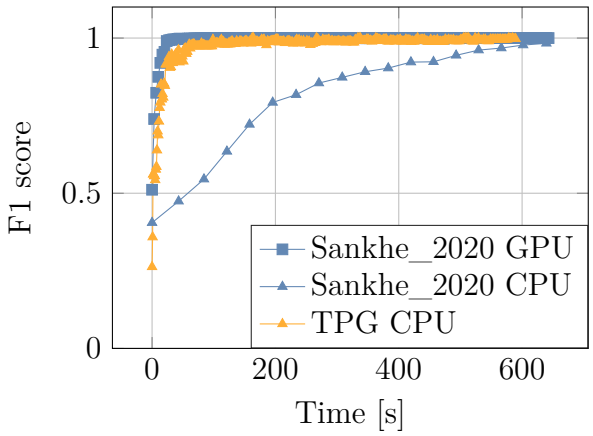
(b) CNN

Table 7.1 – Comparison of confusion Matrix obtained with CNN and TPG for training and test in same conditions with equalized data.

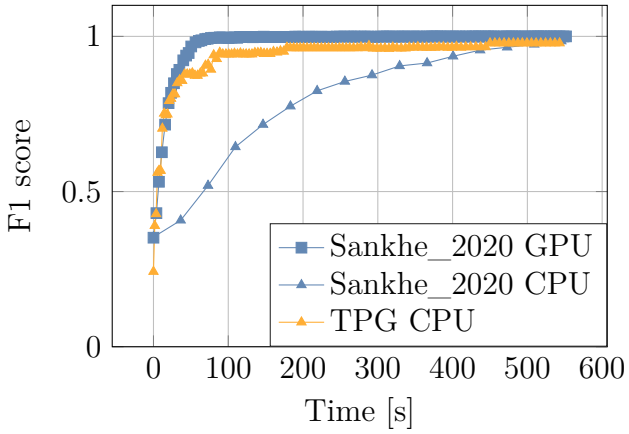
To compare those results in terms of timing, Figures 7.3 give the F1 scores during the training phase as a function of time for different batch sizes. The batch size corresponds to the number of signals used to take a decision and update the classifier: the network or TPG. The yellow triangles represent the F1 score of TPG during the training phase on the CPU, each triangle represents a generation with only one update. For a batch size of



(a) Batch size = 64.



(b) Batch size = 200.



(c) Batch size = 600.

Figure 7.3 – Time evolution of the F1 score of the different networks on different hardware, for different batch sizes.

64 signals which means 64 signals per generation for TPG, the convergence is not stable compared with 600 signals in a generation. The blue triangles represent the evolution of the F1 score for the CNN learning phase on the CPU while the blue squares correspond to the F1 score of the CNN using the GPU, each symbol corresponds to the F1 score value after an epoch or a generation, depending to the batch size the number of update by epoch change. When considering CPU, the TPG exhibits an important speed-up when compared to the CNN. Its speed is very close to a CNN training on a GPU with two advantages (i) the learning can be done on a platform without the GPU accelerator with similar speed (ii) the energy consumption is reduced as only the CPU is used for the TPG.

This analysis shows similar performance between TPG and DL. In the rest of the section, we analyze the behavior of TPGs with different propagation channels. A brief study of the impact of the receiver is proposed in Appendix C.

7.2.4 Environment impact on TPG classification

The impact of environmental change on classification accuracy is now evaluated on TPG. In this part, the only changing factor is the day of the emission. The locations of transceivers do not change, so the propagation channel should not change either. However, the transceivers are not in a controlled room, so 3 factors can affect the RFF:

— The environment channel: the transceivers are not in an anechoïc chamber and interference signals may alter the quality of the labeled database.

— The environmental conditions: the ambient factors such as humidity or temperature are not controlled in the room and can impact the performance of the components and change the distortions.

— The RFF modifications over time: the days of capture signals are distributed over one month so the component degradation could impact the RFF of the devices.

Table 7.2a gives the average accuracy for a test, realized on signals from days 2, 3 and 4, whereas the training is realized with signals from only day 1. The TPG obtains 56% of mean accuracy, with a perfect recongition of certain transmitters such as transmitters 1, 4 and 5.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	77.4	20.0	0.0	1.0	0.4	1.2
Tx_2	2.7	40.8	0.0	56.5	0.0	0.0
Tx_3	6.2	29.7	7.5	0.0	25.3	31.4
Tx_4	2.1	17.3	1.1	79.5	0.0	0.0
Tx_5	0.4	0.0	0.4	0.0	99.2	0.0
Tx_6	0.6	3.1	56.4	3.7	4.3	31.9

(a) Train with day 1 and test with different days.

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	100	0	0	0	0	0
Tx_2	1	60	21	18	0	0
Tx_3	3	1	17	1	7	71
Tx_4	0	0	1	99	0	0
Tx_5	0	0	1	0	99	0
Tx_6	0	11	0	0	0	89

(b) Augmented days training and test on day 1.

Table 7.2 – Confusion Matrix obtained with TPG.

The confusion matrix shows how difficult it is to generalize the training with other environmental conditions, as the conclusion done with neural network in Chapter 2. To

mitigate this problem a common process used with neural networks is data augmentation to present many different environmental conditions to the network [102]. With this augmentation, the network should learn the RFF without the implication of the environment. The augmentation can be realized physically or virtually. For physical augmentation, the number of experiments is increased to create more environmental conditions. Here, the database offers the possibility to physically augment the training dataset thanks to the different days captured. Data from days 2, 3, and 4 is used for training and the test is realized on day 1.

Table 7.2b gives the accuracy results achieved with data augmentation. It shows that TPG is able to generalize the identification and achieve 77% of mean accuracy for the new day. The comparison with Table 7.2a shows that physical data augmentation is interesting for RFF classification, especially with environments where variations may occur.

7.2.5 Study the behavior on WiSig

We propose to realize the same study as the one presented in Chapter 2 with the WiSig dataset. The TPG is trained with the dataset of a particular day and tested with each day dataset. Tables 7.3 show the mean error classification accuracy obtained for each situation, with no equalized data in Table 7.3a and equalized data in Table 7.3b. The rows represent the training day and the columns represent the testing day. These experiments aim to show the ability of the TPG to perform classification when the situation is not the same, and the error classification is given in the tables. The TPG cannot reach 98 or 99% accuracy in training with no equalized data and reached around 60% to 70%. With equalized data, the TPG reached 95% to 98% of accuracy in training. Compared with results obtained in Chapter 2 with the networks, the TPG obtained worse results, with both equalized and no equalized data. The mean error accuracy for no equalized data was 24% while TPG obtained 49%. For equalized data, Sankhe_2020 obtained 16% of mean error accuracy while the TPG obtained 29%. However, using mean accuracy to evaluate the TPG tends to mitigate the interest of TPG because as shown in Table 7.2a, some transmitters are correctly classified and some others are totally misclassified. This is probably due to the non-symmetric characteristic of the TPG. To conclude a 50% error in Table 7.3 can be obtained with 90% of classification for 3 transmitters and around 10% for 3 others.

Train \ Test	day_1	day_2	day_3	day_4
day_1	31	54	58	61
day_2	65	42	50	57
day_3	57	45	31	43
day_4	60	51	49	36

(a) TPG no equalized data.

Train \ Test	day_1	day_2	day_3	day_4
day_1	5	43	40	33
day_2	48	1	43	49
day_3	52	29	2	32
day_4	39	20	34	1

(b) TPG eqalized data.

Table 7.3 – Mean error accuracy in percentage obtained for different days in test and train with WiSig Database.

7.2.6 Conclusion

This section proposes to use a new machine learning technique called TPG to identify devices with RFF recognition. The results show a fast F1 score progression of TPG during the training phase on the CPU. The progression is very close to the F1 score progression of SoA CNN on the GPU. In the second part, TPGs are used to assess a deep analysis of the chosen database and interpretations of the impact of the propagation channel. The analysis concludes with the negative impact of changing captured conditions between the training and test phases for identification. We present the interest of physical data augmentation to be able to identify the transmitters in different situations. The augmentation has to be on different days and proposes different realistic configurations that we can have in the inference phase. Finally, a study is proposed to compare the network and TPG on the WiSig database and allows to conclude on the need to adjust some hyperparameters of TPG. Other studies are provided in Appendix C and has been published in [13].

7.3 Prunning Neural networks

This section presents the study realized during a master research around pruning. DL is often associated with high computational and memory requirements, which can be challenging for embedded systems. To obtain low complexity models, various compression methods have been explored, such as bit quantization [32], transfer learning [53], and network pruning [39, 41]. Over the past 5 years, the use of pruning to reduce complexity has increased significantly. Pruning was introduced in 1989 by LeCun et al. in [57] to

decrease the complexity of learning models by selectively eliminating network elements with the least influence on the model performance. The ratio of neutralized weights is called *sparsity*. There are several network pruning techniques, which are presented in the next subsection.

7.3.1 Prunning Definition and Methods

A. Definition

Pruning is based on the observation that a network naturally contains too many parameters and has many redundancies, wasting space and computation. Pruning takes its name from botany, which involves removing branches from a plant. Thus, pruning a neural network involves removing certain parts of the learning network, whether neurons or filters, to make our algorithm lighter and faster. This method also has the advantage of generalizing the problem and limiting the overfitting effect. The brain has a similar behavior: during the growth period, neurons multiply, and then with age, they diminish, leaving only the most efficient connections. The aim is, therefore, to deconstruct a network as much as possible, as shown in Figure 7.4, where on the right side some connections and neurons have disappeared, without losing the expected performance. This raises the question of whether it is preferable to design a lightweight network directly, or whether it is preferable to make a large network and reduce it using pruning methods.

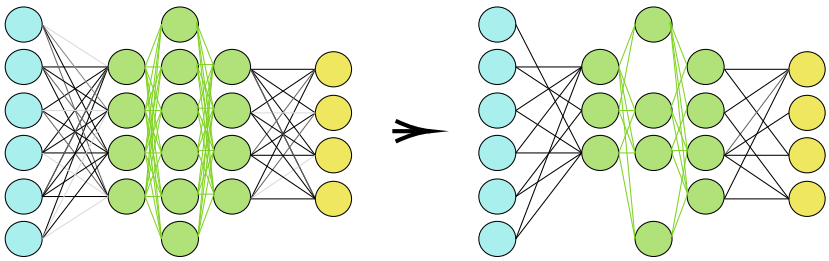


Figure 7.4 – Pruning example on FNN.

Let $\mathcal{N} = \{(W^k, B^k), k \in [\![1, K]\!]\}$ be a convolutional neural network consisting of K layers, with each layer k defined by a weight matrix W^k representing the weights and a bias vector B^k representing the biases. For simplicity, the individual weights are referred to as w_j. We define N as the total number of parameters present in network $\mathcal{N}$. During the pruning, the ratio of weights set to zero is called sparsity and the desired sparsity is

called r. A mask $\mathcal{M} = \{(M_{W^k}, k \in [\![1, K]\!]\}$ is considered for the network $\mathcal{N}$. The matrices M_{W^k} have the same dimensions as W^k.

m_{w_j} refers to the mask value for a given weight w_j, they take the value 1 if the weight of the network is to be kept and 0 if it is to be pruned. We define the pruned network as:

$$\mathcal{N}_r = \{(W^k \odot M_{W^k}), k \in [\![1, K]\!]\}, \tag{7.1}$$

with $\odot$ for element-wise multiplication. We then express the sparsity r as:

$$r = 1 - \frac{1}{N} \times \sum_{w_j \in \mathcal{N}} m_{w_j}. \tag{7.2}$$

The number of biases in a neural network is negligible compared to the number of weights. Therefore, they are ignored during pruning.

There are various pruning methods. It is possible to prune entire filters or neurons, or more finely, by removing certain parts of the filters or certain neuron weights. Moreover, three different pruning configurations exist:

1. **Prunning:** This simply involves removing the weakest weights from the network.
2. **Prunning and re-training:** The pruned network is recovered and then trained again.
3. **Iterative pruning:** This method, illustrated in Figure 7.5, consists of training a network, then pruning and re-training the network several times.

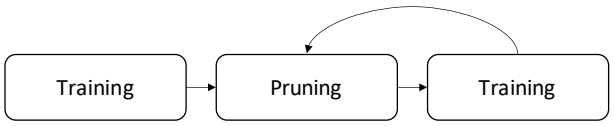


Figure 7.5 – Iterative pruning.

Training after pruning allows the network to reposition itself on the remaining weights while exploiting the strongest connections already formed during the initial training. Neurons whose connections have all been removed must also be deleted. These neurons are considered to be inactive or 'dead', so they will have no contributions to make to the rest of the network.

B. Unstructured pruning - Weight pruning

One of the most common pruning methods is unstructured pruning. This approach involves removing weights from individual items which generally have the lowest values. This can be done either on the whole network or layer by layer.

a. Global approach

Introduced by Han et al. [39] in 2015, this pruning method involves removing the least useful weights across the entire network, independently of layers. To apply this method in this work, we remove weights using the following method:

1. Selection of a threshold value.
2. Choose a norm (L_1 or L_2 norm) or a criterion, presented in the next subsection.
3. Delete all weights in the network whose norm is less than the threshold value.

b. Local approach

Another method of pruning called the 'layer approach' or 'local pruning', exists and involves pruning layer by layer rather than the whole network, as in the previous approach. Each layer is subject to an individually defined threshold. This approach has the advantage of maintaining a regular and balanced network structure, which preserves the quality of predictions. In addition, a balanced structure can facilitate parallelization of the calculation, which can speed up data processing.

c. Criteria for unstructured pruning

To be able to prune, it is necessary to define a criterion. This criterion consists of defining a score S or a rule to select the weights to be removed. Here we only presented the criteria used in this study which are data-independent criteria described in the literature:

Random This criterion involves randomly removing weights across the network.

L1 norm Also known as the magnitude norm, this norm focuses on removing weights with the smallest magnitude [39].

$$S(w_j) = |w_j|. \tag{7.3}$$

LAMP This criterion is a magnitude-based criterion that can be applied globally to the network [58]. It takes into account the relative magnitude of each weight within a given layer W^k. For $w_j \in W^k$:

$$S(w_j, W^k) = \frac{(w_j)^2}{\sum\limits_{w_i \geq w_j, w_i \in W^k} w_i^2}. \tag{7.4}$$

SynFlow This criterion [105] is sensitivity-based. It evaluates the weights based on their sensitivity when subjected to a loss function with input data consisting of ones.

$$S(w_j, g(w_j)) = |w_j \times g(w_j)|. \tag{7.5}$$

Here, $g(w_j)$ represents the gradient value obtained when a loss function is calculated as the sum of all the outputs, given input data consisting entirely of ones, for the network under study with all its weights considered in absolute value.

All criteria outlined here are meant to be applied with global pruning, implying that weight removal is determined by their scores across the entire network, regardless of their specific locations within the network. However, some norms can also be used for local pruning, which means that we prune each layer with the desired sparsity ratio separately. Local pruning means that all layers will undergo pruning with the same ratio.

C. Structured pruning - Pruning neurons and filters

It is also possible to perform structured pruning. This pruning method involves removing not weights, but neurons, filters or whole organized parts of filters (rows, columns, blocks) [89, 49]. While this method ensures that the structure of the network is preserved, it generally results in a greater deterioration in performance than unstructured pruning. For this reason, this method will not be studied in this document.

7.3.2 SoA Prunning applied to RF identification

Jian et al. [49] propose a CNN and a pruning method that uses the Alternating Direction Method of Multipliers to identify devices with their RF signature. They choose to prune only the convolutional layers, justifying this choice by the fact that these are the most computationally demanding parts. This article takes as an example a variant

of ResNet called ResNet50-1D. Wang et al. [113] propose a pruning strategy to perform local pruning as defined above, followed by 5 epochs of re-training. Using this method, the authors obtain a reduced model size of 93.5% of sparsity.

The common approach of the SoA involves retraining, demanding substantial data and extensive computational resources. To address this, we use *data-free* pruning, offering a solution to reduce the size of neural networks while preserving the network identification capabilities without the need for additional computational and memory resources for data processing. Moreover, the article [49] does not explore the behavior of the network in an ever-changing environment. Yet, the primary goal of RFF identification is to recognize devices in different environments, requiring the network to identify devices across various time periods and contexts. We name this capacity *resilience*.

7.3.3 Pruning-based identification system overview

We consider a scenario where multiple transmitters are interacting with a single receiver. The signals emitted by the different transmitters are received and collected to construct a database. This database is then used to train a classifier with the objective of identifying all transmitters independently. Once the network is trained, it is transferred to the receiver device. However, as introduced in Figure 7.6, we propose an additional processing step after the training phase, involving the pruning of the network. While various pruning techniques can be employed, we specifically examine data-free unstructured pruning. Sparse networks, together with the acceleration and memory optimization techniques [100], are promising. Once the network has undergone pruning, it is used in the receiver system, as depicted in Figure 7.6. Our goal is to attain the highest level of network sparsity, while preserving its accuracy performance, without utilizing data for the pruning operation and by preserving resilience.

A. Datasets

The datasets used in this study are Oracle and WiSig, introduced in Chapter 2. For each dataset, two distinct scenarios are defined, designated as Scenario 1 (S1) and Scenario 2 (S2). The datasets are described in Table 7.4. The signals from Scenario 1 are extracted, shuffled, and divided into training and testing datasets (90% assigned to training and 10% assigned to test). The signals from Scenario 2 are employed as a second test (test S2). We define *resilience* as the network ability to perform with data that is either recorded at a

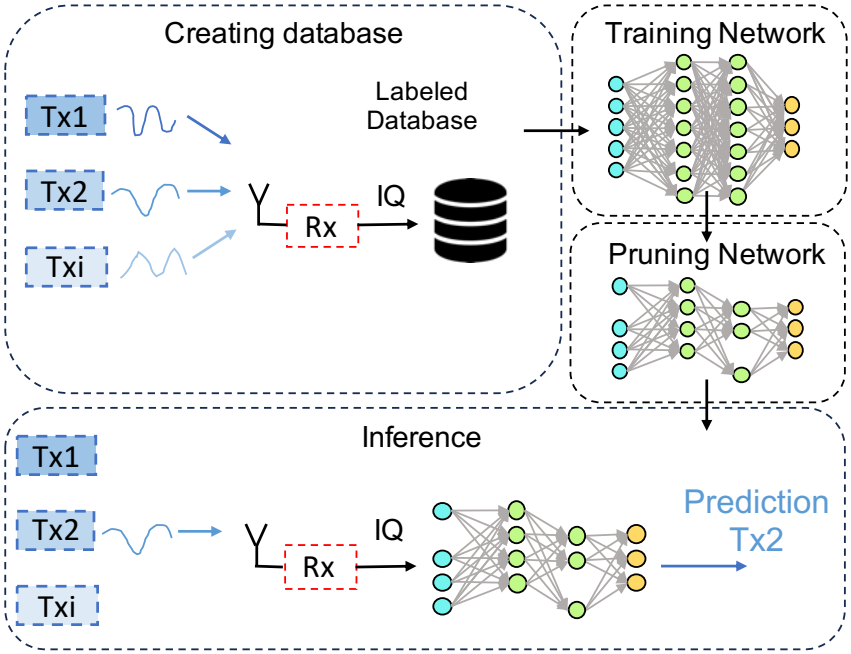


Figure 7.6 – Integrating pruning in RFF identification process.

different time or in a different context than the data it was trained on, here the resilience is measured with S2.

Database	**WiSig - ManySig [40]**	**ORACLE [92]**
Description	Recordings on 4 days over a month.	Signals recorded from 2ft to 62ft with two runs per location.
Transmitters	6 - Atheros AR5212/AR5213	16 - USRP X310
Signals	1000/day/Tx	4000/Tx/location/run
Scenario 1	Day 1 no equalized	Run 1 - 2ft
Scenario 2	Day 2-3-4 no equalized	Run 2 - 2ft

Table 7.4 – Summary of RFF identification datasets and scenarios chosen.

B. Convolutional Neural Networks

We chose to observe the performance of a reference architecture used in previous chapters, Sankhe_2020 [91] and a particularly lightweight network Hanna_2022 [40] presented with the WiSig dataset. It illustrates that networks of fundamentally different complexity can follow a similar methodology for complexity reduction. We give them the same raw

IQ signals in the time domain as input with dimensions of 256×2, representing 256 times samples across two channels (I and Q channels). With these inputs, Sankhe_2020 and Hanna_2022 respectively have 1 232 774 and 39 778 parameters for the classification of six transmitters.

C. Methodology

For reproducibility and representativeness, all experiments are performed on 5 fixed seeds. Each seed has an impact on both the train-test split and the weight initialization. All networks are trained on data from S1 (training dataset) and tested on S1 (testing dataset) and S2. Networks are evaluated on their macro F1 Score (expressed as a percentage).

D. Pruning algorithm

In this work, two different pruning are implemented and tested: local and global. In local pruning, a mask is first created with the same dimensions as the weights for each dense or convolutional layer, with all weights initially set to one. For each layer, scores are computed for each weight. To achieve the desired sparsity level, we calculate the number of weights that should be set to zero. This requires selecting an appropriate threshold and then setting the masks for weights with scores lower than the threshold to zero. After applying pruning to all layers, an element-wise multiplication of the weight matrices with their respective mask matrices is performed.

In global pruning, the methodology is similar. However, the weights are not removed on a per-layer basis, instead, the specified number of weights with the lowest scores are eliminated, regardless of their position in the network.

7.3.4 Experimental study

A. Training behavior

In this section, we examine the performance of the networks during training, using the Adam optimizer, a loss scheduler, diminishing of 10% the loss value every 10 epochs, and cross-entropy as the loss function. Networks are trained for 200 epochs with an early stopping if the loss does not decrease for 10 epochs. At the end of each training epoch, we evaluate the F1 Score of the networks on two different sets of datasets: Scenario 1 (S1) and Scenario 2 (S2).

The mean, minimum, and maximum F1 Scores of both CNNs for the different databases are shown in Table 7.5. The best results between Sankhe_2020 and Hanna_2022, for each scenario and dataset, are underlined.

Scenario	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**
	Sankhe_2020			Hanna_2022		
	WiSig [40]					
S1	<u>99.73</u>	<u>99.34</u>	<u>100.0</u>	99.57	99.02	99.85
S2	<u>58.55</u>	<u>51.47</u>	69.32	56.98	49.23	<u>72.75</u>
	ORACLE [91]					
S1	<u>99.92</u>	<u>99.67</u>	<u>100.0</u>	98.35	97.36	99.55
S2	<u>29.70</u>	<u>26.26</u>	<u>32.89</u>	26.40	25.10	27.67

Table 7.5 – Networks F1 Scores on the different datasets, without pruning.

a. Disparity between datasets and networks

While both networks perform similarly on S1, their performance on S2 varies significantly. The second scenario can be very different from the first, particularly for the Oracle database where the location of the transmitters is not fixed, which reduces the resilience of the network. More precisely, Sankhe_2020 outperforms Hanna_2022 across both databases due to its greater depth and parameter count, allowing it to handle more complex data. Despite Hanna_2022 having significantly fewer parameters, its classification performance remains strong.

b. Variability and Resilience

Hanna_2022 exhibits greater variability than Sankhe_2020 across different seed values, particularly in resilience testing with S2. While Hanna_2022 often shows lower minimum scores compared to Sankhe_2020, some Hanna_2022 realisations outperform any achieved by Sankhe_2020. This variability mainly originates from the initial random network weight values rather than the test/train data distribution, indicating that Hanna_2022 higher dependence on initialization is due to its fewer parameters. Consequently, some Hanna_2022 implementations may exhibit superior resilience compared to any Sankhe_2020 networks.

The choice of network architecture is crucial for RFF identification and should be made carefully. While larger networks may offer greater reliability and adaptability, smaller networks can enhance resilience at the expense of increased instability and slightly lower performance in the first scenario. In the next section, we will delve into pruning as a method to reduce network complexity while maintaining performance.

B. Pruning behavior

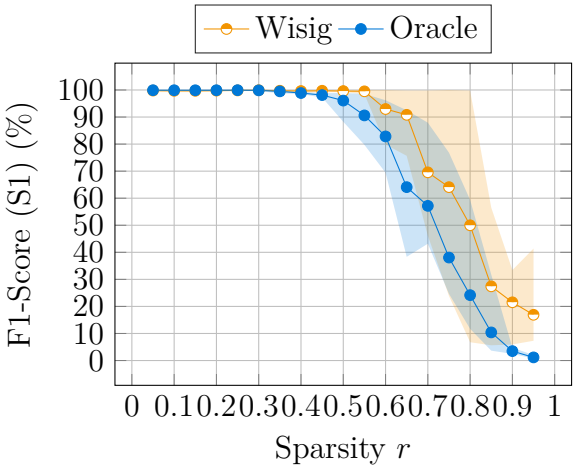
In this section, different pruning approaches are applied on both networks with different criteria and the performance of the network after pruning is evaluated thanks to the databases.

Performances and sparsity

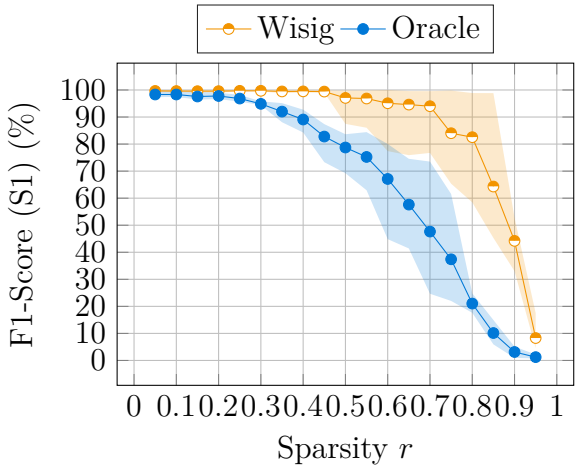
The previously trained networks (see Table 7.5) are pruned at different sparsity levels from 0.05 to 0.95 in 0.05 increments with the LAMP criterion. These pruned networks are then evaluated on both S1 and S2 over all 5 seeds.

The behavior of Sankhe_2020 and Hanna_2022, pruned at different sparsity with the LAMP criterion, are shown in Figure 7.7a and Figure 7.7b on S1 and Figure 7.7c and Figure 7.7d on S2, respectively. The average of the five seeds was plotted for all pruned networks, with the minimum and maximum F1 Scores achieved for each sparsity displayed in transparency. The observations for the LAMP criterion remain valid for the other criteria. A study which taking into account all the criteria mentioned above is presented in Appendix D.

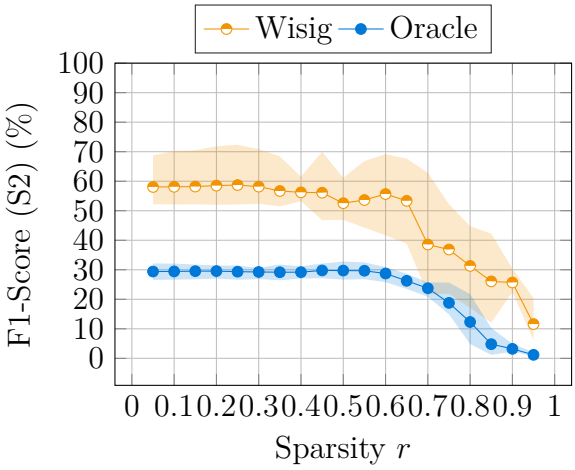
The two CNNs trained on the datasets exhibit different behaviors. Indeed, as the dataset becomes more complex, involving numerous transmitters across various locations and time points, the less the network is able to be sparse with unstructured pruning, as can be observed in the ORACLE datasets. Moreover, it can be observed in Figure 7.7a and Figure 7.7b that Sankhe_2020 is capable of handling a higher degree of pruning than Hanna_2022. This is due to the fact that Sankhe_2020 has more parameters and a greater depth. Finally, Figure 7.7c and Figure 7.7d demonstrate that unstructured pruning, when maintaining the F1 Score on S1, preserves both the original resilience and variability on S2, as was found in Table 7.5.



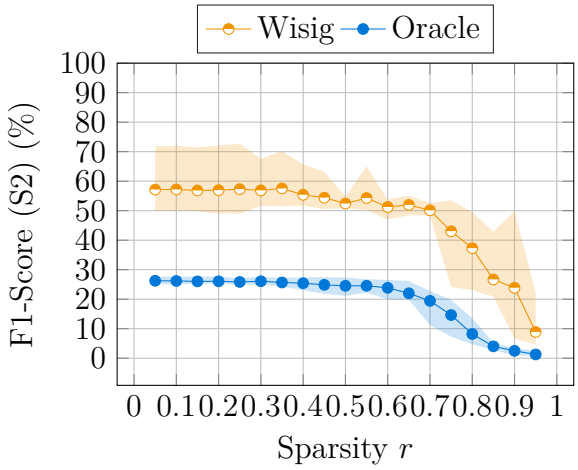
(a) Sankhe_2020 for S1.



(b) Hanna_2022 for S1.



(c) Sankhe_2020 for S2.



(d) Hanna_2022 for S2.

Figure 7.7 – LAMP pruning on different datasets and different networks.

7.3.5 Conclusion

This section proposes the application of unstructured pruning to compress CNNs without retraining, resulting in sparse networks for RFF identification. A comparison of two networks, Sankhe_2020 and the lightweight Hanna_2022, across two different databases reveals notable differences. Sankhe_2020, with a deeper structure, exhibits greater stability across various seeds, while Hanna_2022 instability leads to high-performing realizations only on some seeds.

The study demonstrates that unstructured pruning can achieve sparsity levels ranging from 0.35 to 0.60 without compromising the F1 Score or resilience. However, the choice of the initial network significantly impacts the pruned performance.

Further investigation into iterative pruning may prove beneficial in enhancing both F1 Score and sparsity levels. While this method requires training data, it mitigates the criticality of the initial network structure and enhances classification accuracy. This work will be published in set [7].

Chapter 8

Conclusions and Perspectives

8.1 Conclusions

RFF identification is an emerging physical layer authentication technique that can be used to detect spoofing and distributed denial of service attacks. This method uses the electromagnetic signature of the device in the form of imperfections in the transmitted signal to recognize the device. These identification solutions can be particularly used in the IoT context to reduce the risk of spoofing or to reduce the complexity of the identification process in the transmitter side. RFF identification can be used for different application contexts, attack, defense, or monitoring. There are two types of methods: parametric-based solutions and deep learning-based solutions. This thesis focuses on the DL solutions and the link between the network performances and the training dataset. In particular, the impact of the propagation channel and the bias in the database are considered and studied.

Firstly, this thesis introduces a virtual database generator RiFyFi_VDG based on wireless transmission and RFF models included in a flexible framework RiFyFi for RFF identification. An exploration of the database design for RFF identification with DL is proposed. This exploration considers the similarity between the RFF of transmitters, the transmission scenario, and the number of signals. Our analysis shows the impact of the similarity between RFF transmitters on the network convergence speed and the F1 score performance in a preamble context. A very large number of signals per transmitter is required when the RFF similarities between transmitters are strong or in a payload context, so having similar RFF devices can be a countermeasure to avoid RFF identification. The virtual database generator can help to pre-evaluate the required database design with a lot of flexibility, as shown by changing the OFDM modulation to a single carrier modulation. The RiFyFi is an open-source tool available at [11]. The RFF models allow to have a better understanding of RFF identification. However, it is also interesting to use real data to confirm our hypothesis and the findings obtained thanks to virtual data. Based on our

results obtained with virtual data, we propose six scenarios of increasing difficulty. The study shows that the use of payload data in training improves the classification accuracy and the resilience of the network for other scenarios, with the still open perspective on pre-processing for harsh transmission scenarios.

The second theme of this PhD is to reduce the complexity of identification solutions. We propose to use a new machine learning technique called TPG, which is a reinforcement learning based on genetic programming techniques, to identify devices thanks to their RFF. The TPG-based classification allows to achieve a lightweight and accurate identification. The results show a fast F1 score progression of TPG during the training phase on the CPU. The progression is very close to the F1 score progression of the SoA CNN on the GPU, which means that this machine-learning technique is promising for RFF identification. Finally, SoA pruning techniques have been applied two SoA networks, with different criterion to reduce the complexity of the networks by removing some weights. The study is conducted on two experimental SoA databases and shows that the complexity of the network can be divided by two depending on the initial size of the network.

8.2 Perspectives

During this thesis, several doors opened up thanks to discussions and seminars, some of which will be interesting to explore in future works. The identification of RFF poses several problems depending on the application context considered. As a reminder, this thesis is funded by the DGA, so the most interesting application contexts are defense. For example, one goal may be to identify a suspicious device among many others to detect a potential intruder. The RFF can be used to detect a class of similar devices, the RiFyFi_VDG is a great tool for experiments in this context without the need for many similar transmitters. However, our experiment shows that we need to improve the classification accuracy in noisy contexts, which is the first long-term contribution of this thesis that is considered. In the second time, it could be interesting to use transfer learning to improve classification accuracy. Then, the RiFyFi_VDG gives the opportunity to create a digital twin of a device to improve the reality of our synthetic data. Finally, from a practical point of view, the RFF identification system needs to be embedded in a laptop, for example.

8.2.1 How mitigate the propagation channel effect with signal pre-processing? (long-term)

Some recent SoA papers suggest pre-processing the data before using the neural network. This pre-processing can help to extract a particular impairment. For example, in Chapter 2, IQ imbalance extraction was mentioned thanks to DCTF [73] and PA extraction thanks to density trace figure [60]. Our experience with RFF identification has shown that it is very difficult to overcome a lack of power or a noisy scenario. In particular, the differences obtained between virtual and experimental data show the difficulty of detecting the RFF with the neural network. In this future work, different methods have to be explored, such as specific pre-processing to extract impairments and non-specific methods to improve the network classification even in a blind context. This work can be considered as long-term, as it requires the extension of the SoA to the signal pre-processing and can be a future PhD topic.

8.2.2 Using RiFyFi for Transfer learning (middle term)

The ideal scenario for DL involves having a large amount of labeled training data that matches the distribution of the test data. However, collecting enough training data is expensive, time-consuming, or even unrealistic. Transfer learning offers a promising solution by focusing on transferring knowledge across different domains [125]. This concept, which may have its roots in educational psychology, is supported by C. H. Judd generalization theory of transfer. According to Judd, transfer learning results from the generalization of experience. He suggests that as long as a person can generalize their experiences, it is possible to transfer knowledge from one situation to another. A key prerequisite for this transfer is the presence of a connection between the two learning activities. It could be interesting to perform transfer learning thanks to the digital twins RiFyFi_VDG and observe the performance and the learning capacity obtained on experimental data. For example, the network could be first trained using signals from several virtual devices and then retrained using signals from experimental devices.

8.2.3 Build a digitals twins of our experimental setup (middle term to long term)

The SoA of parametric-based identification gives us several ways to extract and estimate impairments. These techniques could be used to extract the different impairments of the transmitters used to create our real database, such as the CFO, the AM/AM of the PA, and the gain and phase for IQ imbalance. Then RiFyFi_VDG can be parametrized to create the digital twins of our emitters, especially in a context where the access to the device is reduced. It could be interesting to try an attack scenario by capturing the signal of the targeted devices. Then it possible to estimate the different impairments for each device and use RiFyFi_VDG to create a virtual database of emitters and train a network on this database. After that, the network can be used to identify the device in a real context. In the middle term, the inference will be performed in an ideal situation with a wired connection, and then in a real situation.

8.2.4 Lightweight opportunities (short term)

The lightweight solutions presented in this PhD are the main perspectives considered for future work in the short term. In fact, the PhD of Emma BOTHEREAU has begun at IRISA Laboratory in GRANIT team in october 2023 on frugal learning for RFF with lightweight aspect. The idea of this future work is to evaluate the capacity of numerous classes of light or heavy network to perform RFF identification, with resilience capacity when the scenario of the test is different from the training one. Then a perspective could be to implement a network or a light ML solution like TPG on the embedded system to perform the learning or only load the train network on an embedded system.

Finally, in future works, the energy consumption of different network architectures will be compared during the training phase and inference, on different hardware architectures, CPU, GPU or FPGA. This work could be interesting to evaluate the frugality of a network and to compare identification solutions in the context of embedded solutions.

8.2.5 System approach (long term)

Finally, the combined perspectives of this thesis point to a civil or military system with a compromise between security level and energy consumption. Because it is difficult to create a labeled database, it could be interesting to label signals thanks to data

decoding. In this case the devices have to transmit a known sequence, for example with preamble and MAC address. At the receiver side, the sequence is decoded to label the signal and an estimation of the impairments is done thanks to the preamble sequence. The impairments parameters can be used to create a virtual database thanks to RiFyFi_VDG, this database is used to train the network. Finally, depending on the application context and the security/energy trade-off, two identification solutions are: double authentication thanks to the MAC address and the RFF, or using only the RFF to reduce the information transmission cost for the transmitter and thus the energy consumption.

APPENDIX A

Deep Learning complexity

This short section section presents some elements to understand the complexity of the neural network. We propose to evaluate the number of parameters and the number of multiplications per layers.

Parameters: include all elements that are editable during the training step, such as the values of coefficients of the filters, the weights, and the bias.

Multiplication: is the most expensive operation and the most common used in classic and intensive DL. In a network, these multiplications are mainly found at the level of the filters to carry out the convolutions as well as between the inputs of the neurons and their associated weights.

Fully Connected Network

In fully connected layers the number of parameters and multiplication are calculated as follows

Parameters:

Each of the m outputs is a neuron connected to the n input elements. Thus, for a Dense layer, we have weights and bias for each combination between an input node and an output node:

$$nb_{parametres} = (n + 1) \times m, \tag{1}$$

with m the number of neurons and n the number of input elements.

Multiplications:

Each of the m output neurons has n connections to the inputs. We therefore have n multiplications by neurons:

$$nb_{multiplications} = n \times m. \tag{2}$$

Convolutionnal Neural Network

In convolutional layers, the number of parameters and multiplications are calculated as follows

Parameters:

For a convolutional layer, each filter has its own bias. So the number of parameters is

$$nb_{parameters} = (w_f \times h_f \times c_f + 1) \times N, \tag{3}$$

with w_f, h_f and c_f the size of the filter and N the number of filter.

Multiplications:

$$nb_{multiplications} = (w_f \times h_f \times c_f) \times (w_{out} \times h_{out}) \times N. \tag{4}$$

The max-pooling generates no multiplication, because this layer only compares the elements to conserve only one.

Softmax layer

The Softmax layer projects the data into the interval [0,1].

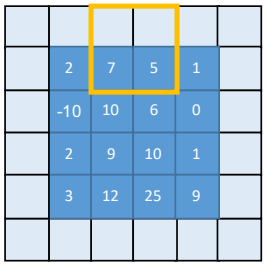
Multiplications :

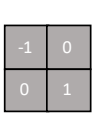
We have one multiplication by elements. So for a layer with N elements, we have

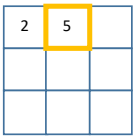
$$nb_{multiplications} = N \tag{5}$$

We define a step or stride s (generally worth 1) as well as a filling or padding p. The stride corresponds to the number of offset elements between 2 windows. Padding allows you to add 0s around the image to change the size of the output image. We generally use

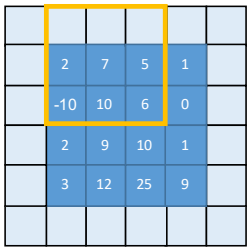
the SamePadding option so that the output image has the same dimensions as the input image.

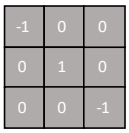


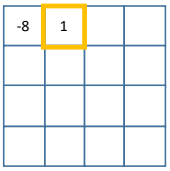




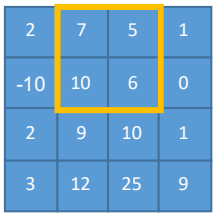
(a) s = 2 and p = 1.

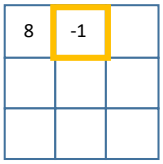






(b) s = 1 and p = SamePadding.





(c) s = 1 and p = 0.

Figure 1 – Convolution with different strides and padding.

Appendix B

RiFyFi parameter values

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

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

Table 1 – CFO values for different similarity scenarios.

	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°

Table 2 – Gain and phase impairments values for different IQ imbalances.

Tx1	Tx2	Tx3	Tx4
10^{-5}	10^{-4}	10^{-6}	10^{-7}

Table 3 – Phase Noise values.

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 4 – Values of impairments for different PA 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

Table 5 – Values of impairments for different all impairments.

Appendix C

Study the influence of the training conditions for TPG

In this section, the impact of both the propagation channel and the receiver is analyzed. The WiSig database is well documented and contains information on the kind of transceiver used, later denoted by reference. The key point is that all the receivers are the same reference (SDR N210) except for two receivers: receivers number 4 (Rx4) and 9 (Rx9), represented by orange circles in Figure 3.6. To stress the impact of the receivers, all training phases are realized with receiver Rx1.

Influence of the channel

The first analysis is done with signals from receiver Rx6. Receivers Rx1 and Rx6 are one meter distance and they are the same reference. Hence, the channel propagation has changed because of the distance between receivers and the RFF of the receiver has changed a little because of the singularity of the two systems. Table 6a shows that our emitters are, on average, correctly identified but the detection performance has been altered compared to the ideal case exposed in Section V-B. Two conclusions can be drawn: Two receivers from the same reference and with closed positions could be swapped during training and test phases with an accuracy penalty with respect to the ideal case. It also proves that different propagation channel between two devices from the same reference affects the results or, in other words, that the network learns a part of the propagation channel.

We now propose to realize the same analysis with the test done on signals from receiver Rx7. This receiver is the same reference as Rx1 and Rx6 but it is localized at the opposite of the room. Table 6b shows that three emitters are correctly identified and the average identification accuracy decreases in comparison with the results achieved by Rx6. The main difference between the two receivers is the location. So in a dynamic context, for which the channel propagation changes between training and test steps, the identification capacity decreases. To mitigate this phenomenon, we propose a channel augmentation using different receivers with the same reference. The confusion matrix 7 shows the result of a training realized on signals from Rx1, 2 and 3 and a test done on signals from Rx5, 6,

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	75.0	8.5	1.1	8.1	2.7	4.6
Tx_2	4.0	57.6	0.4	36.5	1.3	0.1
Tx_3	1.2	11.9	0.6	0.5	5.4	80.4
Tx_4	3.0	7.9	0.5	86.7	1.1	0.8
Tx_5	1.4	0.1	3.8	0.5	94.2	0.0
Tx_6	7.9	51.1	9.8	0.1	31.0	0.1

(a) RX 6

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	83.2	8.7	0.9	4.6	1.1	1.5
Tx_2	2.5	62.5	0.2	34.3	0.3	0.2
Tx_3	19.9	3.1	0.0	75.4	0.7	0.9
Tx_4	0.3	99.3	0.0	0.3	0.1	0.0
Tx_5	0.2	5.5	0.3	1.6	92.4	0.0
Tx_6	24.3	31.3	0.8	41.1	2.5	0.0

(b) RX 7

True \ Guess	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	15.4	12.2	24.4	19.6	16.0	12.5
Tx_2	11.5	7.6	33.1	23.3	16.4	8.0
Tx_3	4.4	4.8	69.9	11.9	7.9	1.1
Tx_4	7.9	10.6	46.0	15.5	13.3	6.6
Tx_5	6.0	7.1	41.2	22.1	19.3	4.2
Tx_6	1.5	2.9	64.5	14.0	11.4	5.7

(c) RX 9

Table 6 – Confusion matrix obtained with TPG for test done with different receivers.

7 and 8. The results are clearly better with the augmentation. The TPG is able to identify transmitters with other receivers in different locations but the same reference when the training phase is done with a diversity of receivers and locations.

This result would be even better by enhancing the augmentation with data from Rx1 moving at more locations.

Influence of the receiver RFF

Finally, we realize the test on the signals from receiver Rx9 again with a training phase on Rx1. Rx9 is close to Rx7 so we can expect a similar confusion matrix as in Table 6b". The main difference between Rx7 and Rx9 is the reference of radio. Rx9 is B210 when Rx7, 1 and 6 are N210. Table 6c shows the results obtained with this configuration and shows the incapacity to correctly identify the transmitter and in particular a strong performance penalty with respect to Table 6b.

Guess / True	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	76.6	4.7	7.0	9.3	0.0	2.4
Tx_2	7.9	65.3	10.6	14.2	0.3	1.7
Tx_3	18.5	2.0	15.1	34.9	11.8	17.7
Tx_4	5.6	69.9	5.6	17.7	0.2	1.0
Tx_5	0.8	0.0	4.3	1.6	87.8	5.5
Tx_6	7.7	0.3	11.3	1.2	2.6	77.0

Table 7 – Confusion Matrix obtained with TPG and augmented receivers training and test on receivers 5 to 8.

Some key assets can be drawn here: even a slight modification of the propagation channel or the environment propagation may lead to an important drop in detection accuracy. Physical data augmentation is thus required to keep good generalization properties. Besides, we prove here that the receiver RFF has a tremendous impact on the capacity to accurately classify a transmitter, even more than the propagation channel itself. It shows the necessity to propose a diverse and extensive dataset that can be physically augmented with a strong variety of propagation channels, environment characteristics, and strong diversity in both transmitter and receiver references.

Appendix D

Pruning performance across all criteria

This section, presents the performance across all criteria, presenting previously, considering three distinct experimentations:

i. Improving F1 Score on S1 through pruning,

ii. Achieving the highest pruning sparsity while maintaining over 99% F1 Score,

iii. Achieving the highest pruning sparsity while maintaining over 95% F1 Score.

All the criteria presented before are used in this section for global pruning. Exepted the L1 criterion is also intended to be used for local pruning, so we will examine the L1 criterion for both local and global pruning.

For each experiment, we calculated the average value across all seeds for all criteria to determine the highest level of sparsity possible. Table 8 presents the best criterion for each experimentation with the F1 Score obtained on S1 and S2 and the sparsity of the best configuration that answers to the experimentation objective.

The Local L1 criterion does not appear in the results table because the local criteria are less effective than global pruning because it uniformly removes weights across all layers. As the pruning ratio increases, layers with fewer parameters are quickly affected, leading to a shortage of connections. Meanwhile, other layers still have many weights that can be pruned without much impact on performance. The most interesting criterion seems to be LAMP in our experiment context. First, we propose to compare the criteria and then compare the networks.

LAMP vs SynFlow

Regarding global criteria, global L1 is less effective than both SynFlow and LAMP. SynFlow performs better on Hanna_2022 than Sankhe_2020, likely due to the smaller size of the network paired with the efficiency of gradient-based pruning, which seems to be particularly effective on small networks like Hanna_2022. But, overall, LAMP proves

to be the most effective criterion for both networks, allowing at least 35% pruning while preserving 99% of the F1 Score.

To enhance the classification process (Experimentation 1), SynFlow appears to be a better choice, as it enhances the F1 Score. Nevertheless, to reduce the number of active parameters, i.e., to achieve higher sparsity (Experimentation 2 and 3), LAMP appears to be the most suitable candidate for RFF identification.

Pruning improving F1 Score (S1)

On average, for Sankhe_2020, we have a F1 Score increase on S1 of 0.01% to 0.19%, while for Hanna_2022, the increase is between 0.01% to 0.15% (compared to unpruned networks, Table 7.5). Nevertheless, these gains are observed at low sparsity levels of only 0.05 to 0.10 for Hanna_2022 and 0.15 to 0.30 for Sankhe_2020.

Sankhe_2020 vs Hanna_2022

Despite Sankhe_2020 having more parameters than Hanna_2022, the sparsity that allows to validate the same experimentation is often smaller for Hanna_2022. Despite Sankhe_2020 being pruned up to 0.55 sparsity on the WiSig dataset in i experimentation, Sankhe_2020 still retains 14 times more active parameters than the original Hanna_2022.

Impact of initial design and pruning process

The initial design of the neural network is crucial for achieving the desired F1 Score and network size as all results are strongly correlated to the F1 Score of the original network for both S1 and S2.

Unstructured pruning methods, without retraining, are consistent across all tested datasets, showing that unstructured pruning is a viable method for reducing the number of active parameters of RFF identification neural networks.

Experimentation	**Criterion**	**F1 Score S1**	**F1 Score S2**	**Sparsity r**
Sankhe_2020				
i	LAMP	99.92	58.17	0.30
ii	LAMP	99.50	53.63	0.55
iii	LAMP	99.50	53.63	0.55
Hanna_2022				
i	Global L1	99.72	57.01	0.05
ii	LAMP	99.41	54.41	0.45
iii	LAMP	95.09	51.24	0.60

(a) WiSig

Experimentation	**Criterion**	**F1 Score S1**	**F1 Score S2**	**Sparsity r**
Sankhe_2020				
i	LAMP	99.93	29.35	0.25
ii	LAMP	99.49	29.17	0.35
iii	LAMP	96.08	29.73	0.50
Hanna_2022				
i	SynFlow	98.37	26.30	0.05
ii	-	-	-	-
iii	SynFlow	95.21	24.75	0.35

(b) ORACLE

Table 8 – Mean F1 Score and Sparsity for Different Models and Criteria.

BIBLIOGRAPHY

[1] *3GPP TS 36.104. Base Station (BS) radio transmission and reception. 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA).*

[2] Mahmoud Abdelaziz et al., « Digital predistortion for hybrid MIMO transmitters », *in*: *IEEE Journal of Selected Topics in Signal Processing* 12.*3* (2018), pp. 445–454.

[3] Fadele Ayotunde Alaba et al., « Internet of Things security: A survey », *in*: *Journal of Network and Computer Applications* (2017).

[4] Saeif Alhazbi et al., « Challenges of Radio Frequency Fingerprinting: From Data Collection to Deployment », *in*: *arXiv preprint arXiv:2310.16406* (2023).

[5] Gianmarco Baldini and Gary Steri, « A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components », *in*: *IEEE Communications Surveys & Tutorials* 19.*3* (2017), pp. 1761–1789.

[6] George Bebis and Michael Georgiopoulos, « Feed-forward neural networks », *in*: *Ieee Potentials* 13.*4* (1994), pp. 27–31.

[7] Emma Bothereau et al., « Investigating Sparse Neural Networks for Radio Frequency Fingerprint Identification », *in*: *Proc. IEEE Vehicular Technology Conference (VTC)*, 2024.

[8] Vladimir Brik et al., « Wireless device identification with radiometric signatures », *in*: *Proc. of the 14th ACM international conference on Mobile computing and networking - MobiCom 08*, San Francisco, California, USA: ACM Press, p. 116, ISBN: 978-1-60558-096-8, DOI: 10.1145/1409944.1409959.

[9] Lvdong Chen et al., « Radio Frequency Fingerprint Identification Based on Transfer Learning », *in*: *Proc. IEEE/CIC International Conference on Communications in China (ICCC)*, Xiamen, China: IEEE, July 2021, pp. 81–85, ISBN: 978-1-66544-385-2, DOI: 10.1109/ICCC52777.2021.9580203.

[10] Sheng Chen, Bernard Mulgrew, and Peter M Grant, « A clustering technique for digital communications channel equalization using radial basis function networks », *in*: *IEEE Transactions on neural networks* 4.*4* (1993), pp. 570–590.

[11] Alice Chillet, *RiFyFi*, URL: `https://github.com/JuliaTelecom/Rifyfi.jl`.

[12] Alice Chillet et al., « How to Design Channel-Resilient Database for Radio Frequency Fingerprint Identification? », *in*: *Proc. IEEE International Conference on Communications (ICC)*, 2024.

[13] Alice Chillet et al., « Tangled Program Graph for Radio-Frequency Fingerprint Identification », *in*: *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023.

[14] Alice Chillet et al., « Understanding Radio Frequency Fingerprint Identification With RiFyFi Virtual Databases », *in*: *IEEE Open Journal of the Communications Society* (2024).

[15] Howard C Choe et al., « Novel identification of intercepted signals from unknown radio transmitters », *in*: *Wavelet Applications II*, vol. 2491, SPIE, 1995, pp. 504–517.

[16] Corinna Cortes and Vladimir Vapnik, « Support-vector networks », *in*: *Machine learning* 20 (1995), pp. 273–297.

[17] Thomas Cover and Peter Hart, « Nearest neighbor pattern classification », *in*: *IEEE transactions on information theory* 13.*1* (1967), pp. 21–27.

[18] Boris Danev and Srdjan Capkun, « Transient-based identification of wireless sensor nodes », *in*: *Proc. International Conference on Information Processing in Sensor Networks*, IEEE, 2009, pp. 25–36.

[19] Boris Danev, Thomas S Heydt-Benjamin, Srdjan Capkun, et al., « Physical-layer Identification of RFID Devices. », *in*: *USENIX security symposium*, 2009, pp. 199–214.

[20] Mickaël Dardaillon et al., « Software defined radio architecture survey for cognitive testbeds », *in*: *Proc. 8th international wireless communications and mobile computing conference (IWCMC)*, IEEE, 2012, pp. 189–194.

[21] Murat Demirbas and Youngwhan Song, « An RSSI-based scheme for sybil attack detection in wireless sensor networks », *in*: *Proc. International symposium on a world of wireless, mobile and multimedia networks (WoWMoM'06)*, ieee, 2006, 5–pp.

[22] Karol Desnos et al., « GEGELATI: Lightweight Artificial Intelligence through Generic and Evolvable Tangled Program Graphs », *in*: *Workshop on Design and Architectures for Signal and Image Processing (DASIP)*, International Conference Proceedings Series (ICPS), Budapest, Hungary: ACM, 2021.

[23] Sepideh Dolatshahi, Adam Polak, and Dennis L Goeckel, « Identification of wireless users via power amplifier imperfections », *in*: *Proc. Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, IEEE, 2010, pp. 1553–1557.

[24] Abdurrahman Elmaghbub and Bechir Hamdaoui, « EPS: Distinguishable IQ Data Representation for Domain-Adaptation Learning of Device Fingerprints », *in*: *arXiv preprint arXiv:2308.04467* (2023).

[25] Abdurrahman Elmaghbub and Bechir Hamdaoui, « LoRa Device Fingerprinting in the Wild: Disclosing RF Data-Driven Fingerprint Sensitivity to Deployment Variability », *in*: *IEEE Access* 9 (2021), pp. 142893–142909.

[26] Abdurrahman Elmaghbub, Bechir Hamdaoui, and Weng-Keen Wong, « ADL-ID: Adversarial Disentanglement Learning for Wireless Device Fingerprinting Temporal Domain Adap », *in*: *arXiv preprint arXiv:2301.12360* (2023).

[27] Junhao Feng et al., « Lightweight CNN-Based RF Fingerprint Recognition Method », *in*: *Proc. 8th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, 2023, pp. 1031–1035.

[28] Hua Fu et al., « Deep learning based RF fingerprint identification with channel effects mitigation », *in*: *IEEE Open Journal of the Communications Society* (2023).

[29] Kaifeng Gao et al., « Julia language in machine learning: Algorithms, applications, and open issues », *in*: *Computer Science Review* 37 (2020), p. 100254.

[30] Robin Gerzaguet et al., « The 5G candidate waveform race: a comparison of complexity and performance », *in*: *EURASIP Journal on Wireless Communications and Networking* 2017 (2017), pp. 1–14.

[31] s Gerzaguet et al., « On the performance of digital adaptive spur cancellation for multi-standard radio frequency transceivers », *in*: *Digital Signal Processing* 33 (2014), pp. 83–97.

[32] Amir Gholami et al., « A Survey of Quantization Methods for Efficient Neural Network Inference », *in*: *ArXiv* abs/2103.13630 (2021).

[33] David Godard, « Channel equalization using a Kalman filter for fast data transmission », *in*: *IBM journal of Research and Development* 18.*3* (1974), pp. 267–273.

[34] Xinghao Guo, Zhen Zhang, and Jie Chang, « Survey of Mobile Device Authentication Methods Based on RF Fingerprint », *in*: *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France: IEEE, Apr. 2019, pp. 1–6, ISBN: 978-1-72811-878-9, DOI: 10.1109/INFOCOMWKSHPS47286.2019.9093755.

[35] Jose A Gutierrez del Arroyo, Brett J Borghetti, and Michael A Temple, « Considerations for radio frequency fingerprinting across multiple frequency channels », *in*: *Sensors* 22.*6* (2022), p. 2111.

[36] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis, « Detection of Transient in Radio Frequency Fingerprint using Signal Phase », *in*: *Wireless and optical communications* (2003), p. 6.

[37] Jeyanthi Hall, Michel Barbeau, Evangelos Kranakis, et al., « Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. », *in*: *Communications, internet, and information technology* 1 (2004).

[38] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis, « Radio frequency fingerprinting for intrusion detection in wireless networks », *in*: *IEEE Transactions on Defendable and Secure Computing* 12 (2005), pp. 1–35.

[39] Song Han et al., « Learning both Weights and Connections for Efficient Neural Network », *in*: *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.

[40] Samer Hanna, Samurdhi Karunaratne, and Danijela Cabric, « WiSig: A large-scale WiFi signal dataset for receiver and channel agnostic RF fingerprinting », *in*: *IEEE Access* 10 (2022), pp. 22808–22818.

[41] Yang He et al., « Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration », *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4335–4344.

[42] Weikun Hou, Xianbin Wang, and Jean-Yves Chouinard, « Physical layer authentication in OFDM systems based on hypothesis testing of CFO estimates », *in*: *Proc. IEEE International Conference on Communications (ICC)*, 2012, pp. 3559–3563.

[43] Weikun Hou et al., « Physical layer authentication for mobile systems with time-varying carrier frequency offsets », *in*: *IEEE Transactions on Communications* 62.*5* (2014), pp. 1658–1667.

[44] Jiasen Huang et al., « A novel joint estimation and compensation algorithm for non-idealities of analog front-end in DC-OFDM system », *in*: *Proc. IEEE 10th international conference on ASIC (ASICON)*, ISSN: 2162-7541, Oct. 2013, pp. 1–4, DOI: 10.1109/ASICON.2013.6811975.

[45] Anu Jagannath and Jithin Jagannath, « Embedding-Assisted Attentional Deep Learning for Real-World RF Fingerprinting of Bluetooth », *in*: *IEEE Transactions on Cognitive Communications and Networking* (Sept. 2022), DOI: 10.36227/techrxiv.20767315.v1.

[46] Anu Jagannath, Jithin Jagannath, and Prem Sagar Pattanshetty Vasanth Kumar, « A Comprehensive Survey on Radio Frequency (RF) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges », *in*: *arXiv:2201.00680 [cs]* (2022).

[47] Tong Jian et al., « Deep Learning for RF Fingerprinting: A Massive Experimental Study », *in*: *IEEE Internet of Things Magazine* 3.*1* (2020), pp. 50–57, ISSN: 2576-3199, DOI: 10.1109/IOTM.0001.1900065.

[48] Tong Jian et al., « MAC ID spoofing-resistant radio fingerprinting », *in*: *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[49] Tong Jian et al., « Radio Frequency Fingerprinting on the Edge », *in*: *IEEE Transactions on Mobile Computing* (2021), pp. 1–1, ISSN: 1536-1233, 1558-0660, 2161-9875, DOI: 10.1109/TMC.2021.3064466.

[50] Stephen Kelly, « Scaling Genetic Programming to Challenging Reinforcement Tasks through Emergent Modularity », *in*: (2018).

[51] Stephen Kelly and Malcolm I Heywood, « Emergent tangled graph representations for Atari game playing agents », *in*: *Proc. European Conference on Genetic Programming*, 2017, pp. 64–79.

[52] Stephen Kelly et al., « A modular memory framework for time series prediction », *in*: *Proc. of the 2020 Genetic and Evolutionary Computation Conference (GECCO)*, New York, NY, USA: Association for Computing Machinery, pp. 949–957, ISBN: 978-1-4503-7128-5, DOI: 10.1145/3377930.3390216.

[53] Seunghyeon Kim, Yung-Kyun Noh, and Frank Chongwoo Park, « Efficient neural network compression via transfer learning for machine vision inspection », *in*: *Neurocomputing* 413 (2020), pp. 294–304.

[54] Randall W Klein, Michael A Temple, and Michael J Mendenhall, « Application of wavelet-based RF fingerprinting to enhance wireless network security », *in*: *Journal of Communications and Networks* 11.*6* (2009), pp. 544–555.

[55] David A Knox and Thomas Kunz, « Wireless fingerprints inside a wireless sensor network », *in*: *ACM Transactions on Sensor Networks (TOSN)* 11.*2* (2015), pp. 1–30.

[56] Corentin Lavaud et al., « AbstractSDRs: Bring down the two-language barrier with Julia Language for efficient SDR prototyping », *in*: *IEEE Embedded Systems Letters* (2021), pp. 1–1, ISSN: 1943-0663, 1943-0671, DOI: 10.1109/LES.2021.3054174.

[57] Yann LeCun, John Denker, and Sara Solla, « Optimal Brain Damage », *in*: *Advances in Neural Information Processing Systems*, ed. by D. Touretzky, vol. 2, Morgan-Kaufmann, 1989.

[58] Jaeho Lee et al., « Layer-adaptive Sparsity for the Magnitude-based Pruning », *in*: *Proc. International Conference on Learning Representations (ICLR)*, 2020.

[59] Achankeng Leke and John M Cioffi, « A maximum rate loading algorithm for discrete multitone modulation systems », *in*: *IEEE Global Telecommunications Conference. Conference Record (GLOBECOM)*, vol. 3, 1997, pp. 1514–1518.

[60] Yuepei Li et al., « Power Amplifier enabled RF Fingerprint Identification », *in*: *Proc. IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, Waco, TX, USA: IEEE, May 2021, pp. 1–6, ISBN: 978-1-66540-309-2, DOI: 10.1109/WMCS52222.2021.9493272.

[61] Zewen Li et al., « A survey of convolutional neural networks: analysis, applications, and prospects », *in*: *IEEE transactions on neural networks and learning systems* 33.*12* (2021), pp. 6999–7019.

[62] Yun Lin et al., « Wireless Device Identification Based on Radio Frequency Fingerprint Features », *in*: *Proc. IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1–6, ISBN: 978-1-72815-089-5, DOI: 10.1109/ICC40277.2020.9149226.

[63] Kevin Merchant et al., « Deep learning for RF device fingerprinting in cognitive communication networks », *in*: *IEEE journal of selected topics in signal processing* 12.*1* (2018), pp. 160–167.

[64] Joseph Mitola, « Software radios: Survey, critical evaluation and future directions », *in*: *IEEE Aerospace and Electronic Systems Magazine* 8.*4* (1993), pp. 25–36.

[65] Cyrille Morin et al., « Deep Learning-based Transmitter identification on the physical layer », PhD thesis, 2020.

[66] Cyrille Morin et al., « Transmitter classification with supervised deep learning », *in*: *Proc. 14th EAI International Conference on Cognitive Radio-Oriented Wireless Networks (CrownCom)*, 2019, pp. 73–86.

[67] Nam Tuan Nguyen et al., « Device fingerprinting to enhance wireless security using nonparametric Bayesian method », *in*: *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM)*, 2011, pp. 1404–1412.

[68] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy, « Convolutional radio modulation recognition networks », *in*: *Proc. 17 International Conference Engineering Applications of Neural Networks (EANN)*, 2016, pp. 213–226.

[69] *ORBIT wireless research laboratory WINLAB, Rutgers University. http://www.orbit-lab.org/.*

[70] Hiren J Patel, Michael A Temple, and Rusty O Baldwin, « Improving ZigBee device network authentication using ensemble decision tree classifiers with radio frequency distinct native attribute fingerprinting », *in*: *IEEE transactions on reliability* 64.*1* (2014), pp. 221–233.

[71] Neal Patwari and Sneha K Kasera, « Robust location distinction using temporal link signatures », *in*: *Proc. of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 111–122.

[72] Linning Peng et al., « Deep Learning Based RF Fingerprint Identification Using Differential Constellation Trace Figure », *in*: *IEEE Transactions on Vehicular Technology* 69.*1* (Jan. 2020), pp. 1091–1095, ISSN: 0018-9545, 1939-9359, DOI: 10.1109/TVT.2019.2950670.

[73] Linning Peng et al., « Design of a Hybrid RF Fingerprint Extraction and Device Classification Scheme », *in*: *IEEE Internet of Things Journal* 6.*1* (Feb. 2019), pp. 349–360, ISSN: 2327-4662, 2372-2541, DOI: 10.1109/JIOT.2018.2838071.

[74] Jaakko Pihlajasalo et al., « Deep Learning OFDM Receivers for Improved Power Efficiency and Coverage », *in*: *IEEE Transactions on Wireless Communications* (2023).

[75] Adam C Polak, Sepideh Dolatshahi, and Dennis L Goeckel, « Identifying wireless users via transmitter imperfections », *in*: *IEEE Journal on selected areas in communications* 29.*7* (2011), pp. 1469–1479.

[76] Adam C Polak and Dennis L Goeckel, « Identification of wireless devices of users who actively fake their RF fingerprints with artificial data distortion », *in*: *IEEE Transactions on Wireless Communications* 14.*11* (2015), pp. 5889–5899.

[77] Adam C. Polak and Dennis L. Goeckel, « Wireless device identification based on RF oscillator imperfections », *in*: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy: IEEE, May 2014, pp. 2679–2683, ISBN: 978-1-4799-2893-4, DOI: 10.1109/ICASSP.2014.6854086.

[78] Thierry Pollet, Mark Van Bladel, and Marc Moeneclaey, « BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise », *in*: *IEEE Transactions on communications* 43.*2/3/4* (1995), pp. 191–193.

[79] Benjamin W Ramsey, Michael A Temple, and Barry E Mullins, « PHY foundation for multi-factor ZigBee node authentication », *in*: *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 795–800.

[80] Saeed Ur Rehman, Kevin Sowerby, and Colin Coghill, « Analysis of receiver front end on the performance of RF fingerprinting », *in*: *Proc. IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sydney, Australia, 2012, pp. 2494–2499, DOI: 10.1109/PIMRC.2012.6362777.

[81] KA Remley et al., « Electromagnetic signatures of WLAN cards and network security », *in*: *Proc. of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.* 2005, pp. 484–488.

[82] Francesco Restuccia et al., « DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms », *in*: *Proc. Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2019, pp. 51–60, DOI: 10.1145/3323679.3326503.

[83] Guillem Reus-Muns et al., « Trust in 5G open RANs through machine learning: RF fingerprinting on the POWDER PAWR platform », *in*: *Proc. IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2020, pp. 1–6.

[84] Shamnaz Riyaz et al., « Deep Learning Convolutional Neural Networks for Radio Identification », *in*: *IEEE Communications Magazine* 56.*9* (Sept. 2018), pp. 146–152, ISSN: 0163-6804, 1558-1896, DOI: 10.1109/MCOM.2018.1800153.

[85] Josh Robinson and Scott Kuzdeba, « RiftNet: Radio Frequency Classification for Large Populations », *in*: *Proc. IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2021, pp. 1–6, ISBN: 978-1-72819-794-4, DOI: 10.1109/CCNC49032.2021.9369455.

[86] Josh Robinson et al., « Dilated Causal Convolutional Model For RF Fingerprinting », *in*: *Proc. IEEE 10th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2020, pp. 0157–0162, ISBN: 978-1-72813-783-4, DOI: 10.1109/CCWC47524.2020.9031257.

[87] Debashri Roy et al., « Detection of rogue RF transmitters using generative adversarial nets », *in*: *Proc. IEEE wireless communications and networking conference (WCNC)*, 2019, pp. 1–7.

[88] Debashri Roy et al., « RFAL: Adversarial learning for RF transmitter identification and classification », *in*: *IEEE Transactions on Cognitive Communications and Networking* 6.*2* (2019), pp. 783–801.

[89] Muhammad Sabih, Frank Hannig, and Jürgen Teich, « Utilizing Explainable AI for Quantization and Pruning of Deep Neural Networks », *in*: *ACM Transactions on Embedded Computing Systems (TECS)* 19.*4* (2020), pp. 1–24, DOI: 10.1145/3370661.

[90] Hojjat Salehinejad et al., « Recent advances in recurrent neural networks », *in*: *arXiv preprint arXiv:1801.01078* (2017).

[91] Kunal Sankhe et al., « *No Radio Left Behind:* Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments », *in*: *IEEE Transactions on Cognitive Communications and Networking* 6.*1* (2020), pp. 165–178, ISSN: 2332-7731, 2372-2045, DOI: 10.1109/TCCN.2019.2949308.

[92] Kunal Sankhe et al., « ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks », *in*: *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Paris, France, 2019, pp. 370–378, ISBN: 978-1-72810-515-4, DOI: 10.1109/INFOCOM.2019.8737463.

[93] Amani Al-Shawabka et al., « DeepLoRa: Fingerprinting LoRa devices at scale through deep learning and data augmentation », *in*: *Proc. of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 251–260.

[94] Amani Al-Shawabka et al., « Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting », *in*: *Proc.IEEE Conference on Computer Communications (INFOCOM)*, Toronto, ON, Canada, 2020, pp. 646–655, ISBN: 978-1-72816-412-0, DOI: 10.1109/INFOCOM41043.2020.9155259.

[95] Guanxiong Shen et al., « Radio frequency fingerprint identification for LoRa using deep learning », *in*: *IEEE Journal on Selected Areas in Communications* 39.*8* (2021), pp. 2604–2616.

[96] Guanxiong Shen et al., « Radio Frequency Fingerprint Identification for Security in Low-Cost IoT Devices », *in*: *arXiv:2111.14275 [eess]* (2021).

[97] Guanxiong Shen et al., « Towards length-versatile and noise-robust radio frequency fingerprint identification », *in*: *IEEE Transactions on Information Forensics and Security* (2023).

[98] Guanxiong Shen et al., « Towards Scalable and Channel-Robust Radio Frequency Fingerprint Identification for LoRa », *in*: *arXiv:2107.02867 [eess]* (July 2021).

[99] Yong Sheng et al., « Detecting 802.11 MAC layer spoofing using received signal strength », *in*: *Prco. IEEE 27th Conference on Computer Communications (INFOCOM)*, 2008, pp. 1768–1776.

[100] Siddharth Singh and Abhinav Bhatele, « Exploiting Sparsity in Pruned Neural Networks to Optimize Large Model Training », *in*: *IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2023), pp. 245–255.

[101] Robert J Smith, Ryan Amaral, and Malcolm I Heywood, « Evolving simple solutions to the CIFAR-10 benchmark using tangled program graphs », *in*: *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2061–2068.

[102] Nasim Soltani et al., « More Is Better: Data Augmentation for Channel-Resilient RF Fingerprinting », *in*: *IEEE Communications Magazine* 58.*10* (2020), pp. 66–72, ISSN: 0163-6804, 1558-1896, DOI: 10.1109/MCOM.001.2000180.

[103] Naeimeh Soltanieh et al., « A Review of Radio Frequency Fingerprinting Techniques », *in*: *IEEE Journal of Radio Frequency Identification* 4.*3* (2020), pp. 222–233, ISSN: 2469-7281, 2469-729X, DOI: 10.1109/JRFID.2020.2968369.

[104] William C. Suski II et al., « Using Spectral Fingerprints to Improve Wireless Network Security », *in*: *Proc. Global Telecommunications Conference (GLOCOM)*, New Orleans, LA, USA, 2008, pp. 1–5, ISBN: 978-1-4244-2324-8, DOI: 10.1109/GLOCOM.2008.ECP.421.

[105] Hidenori Tanaka et al., « Pruning neural networks without any data by iteratively conserving synaptic flow », *in*: *ArXiv* abs/2006.05467 (2020).

[106] J Toonstra and Wintold Kinsner, « Transient analysis and genetic algorithms for classification », *in*: *Proc. IEEE Communications, Power, and Computing. Conference Proceedings (WESCANEX)*, vol. 2, 1995, pp. 432–437.

[107] J. Tubbax et al., « Compensation of IQ imbalance and phase noise in OFDM systems », *in*: *IEEE Transactions on Wireless Communications* 4.*3* (2005), pp. 872–877, ISSN: 1558-2248, DOI: 10.1109/TWC.2004.843057.

[108] Saeed Ur Rehman, Kevin Sowerby, and Colin Coghill, « RF fingerprint extraction from the energy envelope of an instantaneous transient signal », *in*: *Proc. IEEE Australian Communications Theory Workshop (AusCTW)*, Wellington, New Zealand, 2012, pp. 90–95, ISBN: 978-1-4577-1962-2 978-1-4577-1961-5 978-1-4577-1959-2 978-1-4577-1960-8, DOI: 10.1109/AusCTW.2012.6164912.

[109] Oktay Ureten and Nur Serinken, « Wireless security through RF fingerprinting », *in*: *Canadian Journal of Electrical and Computer Engineering* 32.*1* (2007), pp. 27–33.

[110] M. Valkama, M. Renfors, and V. Koivunen, « Blind source separation based I/Q imbalance compensation », *in*: *Proc. IEEE adaptive systems for signal processing, communications, and control symposium (AS-SPCC)*, 2000, pp. 310–314, DOI: 10.1109/ASSPCC.2000.882491.

[111] Mikko Valkama, Markku Renfors, and Visa Koivunen, « Blind I/Q signal separation-based solutions for receiver signal processing », *in*: *EURASIP Journal on Advances in Signal Processing* 2005.*16* (2005), pp. 1–11.

[112] Wenhao Wang et al., « Wireless Physical-Layer Identification: Modeling and Validation », *in*: *IEEE Transactions on Information Forensics and Security* 11.*9* (Sept. 2016), pp. 2091–2106, ISSN: 1556-6013, 1556-6021, DOI: 10.1109/TIFS.2016.2552146.

[113] Yu Wang et al., « LightAMC: Lightweight Automatic Modulation Classification via Deep Learning and Compressive Sensing », *in*: *IEEE Transactions on Vehicular Technology* 69.*3* (Mar. 2020), pp. 3491–3495, ISSN: 1939-9359, DOI: 10.1109/TVT.2020.2971001.

[114] Liang Xiao et al., « Channel-based detection of sybil attacks in wireless networks », *in*: *IEEE Transactions on Information Forensics and Security* 4.*3* (2009), pp. 492–503.

[115] Ning Xie, Zhuoyuan Li, and Haijun Tan, « A survey of physical-layer authentication in wireless communications », *in*: *IEEE Communications Surveys & Tutorials* 23.*1* (2020), pp. 282–310.

[116] Tian Yang et al., « Conventional Neural Network-Based Radio Frequency Fingerprint Identification Using Raw I/Q Data », *in*: *Wireless Communications and Mobile Computing* (2022), p. 8.

[117] Honglin Yuan et al., « Stable Nonlinear and IQ Imbalance RF Fingerprint for Wireless OFDM Devices », *in*: *arXiv preprint arXiv:2104.10397* (), p. 7.

[118] Yingjun Yuan et al., « Specific emitter identification based on Hilbert–Huang transform-based time–frequency–energy distribution features », *in*: *IET communications* 8.*13* (2014), pp. 2404–2412.

[119] Davide Zanetti, Boris Danev, and Srdjan Capkun, « Physical-layer identification of UHF RFID tags », *in*: *Proc. of the sixteenth annual international conference on Mobile computing and networking*, 2010, pp. 353–364.

[120] Shuiguang Zeng et al., « Physical layer authentication based on cfo and visibility graph », *in*: *Proc. IEEE International Conference on Networking and Network Applications (NaNA)*, 2018, pp. 147–152.

[121] Junqing Zhang et al., « Radio frequency fingerprint identification for narrowband systems, modelling and classification », *in*: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 3974–3987.

[122] Zhen Zhang et al., « An artificial radio frequency fingerprint embedding scheme for device identification », *in*: *IEEE Communications Letters* 26.*5* (2022), pp. 974–978.

[123] Sheng Zhong et al., « Privacy-preserving location-based services for mobile users in wireless networks », *in*: *Department of Computer Science, Yale University, Technical Report ALEU/DCS/TR-1297* 26 (2004).

[124] A. Zhu, J. C. Pedro, and T. J. Brazil, « Dynamic deviation reduction-based volterra behavioral modeling of RF power amplifiers », *in*: *IEEE Transactions on Microwave Theory and Techniques* 54.*12* (2006), pp. 4323–4332, ISSN: 0018-9480, DOI: 10.1109/TMTT.2006.883243.

[125] Fuzhen Zhuang et al., « A comprehensive survey on transfer learning », *in*: *Proceedings of the IEEE* 109.*1* (2020), pp. 43–76.

[126] Qiyue Zou, A. Tarighat, and A.H. Sayed, « Compensation of phase noise in OFDM wireless systems », *in*: *IEEE Transactions on Signal Processing* 55.*11* (Nov. 2007), pp. 5407–5424, ISSN: 1053-587X, DOI: 10.1109/TSP.2007.899583.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Identification de dispositif sensibles par apprentissage de l'empreinte Radio Fréquence

Mot clés : Apprentissage automatique supervisé, empreinte RF,

Résumé : L'identification de dispositifs dits sensibles est soumise à différentes contraintes de sécurité ou de consommation d'énergie, ce qui rend les méthodes d'identification classique peu adaptées. Pour répondre à ces contraintes, il est possible d'utiliser les défauts intrinsèques de la chaîne de transmission des dispositifs pour les identifier. Ces défauts altèrent le signal transmis et créent alors une signature unique appelée empreinte Radio Fréquence (RF). Pour identifier un dispositif grâce à son empreinte RF, il est possible d'utiliser des méthodes paramétriques pour extraire une signature qui peut être utilisée par un classifieur, ou bien d'utiliser des méthodes d'apprentissage telles que les réseaux de neurones. Toutefois, la capacité d'un réseau de neurones à reconnaître un dispositif dans un contexte particulier dépend fortement de la base de données d'entraînement. Dans cette thèse, nous proposons un générateur de bases de données virtuelles basé sur des modèles de transmission et d'imperfections RF, permettant d'étudier la robustesse d'un réseau en fonction des donnée d'apprentissage. Dans un second temps, nous proposons de réduire la complexité de l'identification via deux axes. Le premier consiste à utiliser des graphes programmables intriqués, qui sont des modèles d'apprentissage par renforcement, basés sur des techniques d'évolution génétique moins complexes que les réseaux de neurones. Le second axe propose l'utilisation de l'élagage sur des réseaux de neurones de la littérature pour réduire la complexité de ces derniers.

Title: Sensitive Devices Identification through Learning of Radio Frequency Fingerprint

Keywords: Deep Learning, Radio Frequency Fingerprint

Abstract: The identification of so-called sensitive devices is subject to various security or energy consumption constraints, making conventional identification methods unsuitable. To meet these constraints, it is possible to use intrinsic faults in the device's transmission chain to identify it. These faults alter the transmitted signal and create a unique signature called the Radio Frequency (RF) fingerprint. To identify a device using its RF fingerprint, it is possible to use parametric methods to extract a signature that can be used by a classifier, or to use learning methods such as neural networks. However, the ability of a neural network to recognise a device in a particular context is highly dependent on the training database. In this thesis, we propose a virtual database generator based on transmission models and RF imperfections, making it possible to study the robustness of a network as a function of the training data. Secondly, we propose to reduce the complexity of identification in two ways. The first involves the use of intricate programmable graphs, which are reinforcement learning models based on genetic evolution techniques that are less complex than neural networks. The second involves the use of pruning on neural networks from the literature to reduce their complexity.