Graphes programmables intriqués pour l'identification d'empreintes Radio-Fréquence

Alice Chillet, Baptiste Boyer, Robin Gerzaguet, Karol Desnos, Matthieu Gautier

Graphes programmables intriqués pour l'identification d'empreintes Radio-Fréquence

Alice CHILLET[1] Baptiste BOYER[1] Robin GERZAGUET[1] Karol DESNOS[2] Matthieu GAUTIER[1]

[1]Univ Rennes, CNRS, IRISA, France

[2]Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France

Résumé – Ce papier propose d'utiliser les graphes programmables intriqués ou Tangled Program Graph (TPG) dans le cadre de l'identification d'empreintes Radio Frequence (RF) ou Radio Frequency Fingerprint. Une empreinte RF est une signature unique créée par les distorsions électromagnétiques des différents composants matériels du dispositif radio. Cette signature est inscrite dans le signal émis et peut être utilisée comme un identifiant, par nature sécurisé car non reproductible. L'état de l'art présente des solutions d'identification par classification des empreintes grâce à des Réseaux de Neurones Convolutifs ou Convolutional Neural Network (CNN). Les TPG sont des modèles d'apprentissage par renforcement ou Renforcement Learning (RL), basés sur des techniques d'évolution génétique moins complexes que les réseaux de neurones. Ce papier propose d'utiliser la classification par TPG pour identifier à moindre coût et efficacement des empreintes RF. Les résultats montrent que la vitesse de convergence des TPG est comparable à celle obtenue par un réseau de neurones de l'état de l'art avec des performances en test comparables.

Abstract – This paper proposes to use Tangled Program Graph (TPG) for Radio Frequency Fingerprint (RFF) identification. RFF is a unique signature created by electromagnetic distortions of the different radio frequency hardware components in the device. This signature is contained in the transmitted signal and may be used as a secure identifier because it can not be easily spoofed. In recent years, RFF identification is mainly based on Deep Learning (DL). TPG is a new machine learning technique based on genetic evolution and is less complex than DL. In this paper, we propose to use TPG-based classification to achieve a lightweight and accurate RFF identification scheme. Results show that TPG achieve the same accuracy as a state-of-the-art convolutional neural network with a learning phase duration clearly reduced on CPU.

1 Introduction

L'identification par empreinte Radio Frequence (RF) ou Radio Frequency Fingerprint est une solution d'identification sécurisée, utilisant les singularités des composants analogiques des dispositifs communicants pour les identifier. Chaque composant analogique de la chaîne de transmission créé des distorsions électromagnétiques uniques dans le signal transmis appelées empreinte RF. Une radio logicielle ou Software Defined Radio (SDR) peut capter des signaux sur une large bande sous forme d'échantillons Phase-Quadrature (IQ) et utiliser cette empreinte pour en identifier l'émetteur. Celle-ci permet également de réaliser les différents traitements nécessaires à l'identification. En effet, pour identifier des dispositifs à partir de leurs empreintes, il existe deux familles de méthodes : les méthodes paramétriques et les méthodes par apprentissage. Les méthodes paramétriques suivent deux étapes : premièrement les caractéristiques de l'empreinte sont extraites du signal reçu puis un algorithme de classification est utilisé pour classer les signaux.

Les méthodes par apprentissage profond ou Deep Learning (DL) supervisé sont de plus en plus répandues et souvent réalisées par des Réseaux de Neurones ou Neural Network (NN) 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 [8, 9], Sankhe et al. explorent différentes architectures de CNN avec plus ou moins de paramètres. Un NN récurrent a également été exploré [12], tout comme les transformers, un autre type d'architecture de DL [10].

Les NN 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 reste fortement complexe. Dans le contexte de l'internet des objets (IoT), l'identification par RF peut être contrainte en termes de complexité, d'architecture matériel et de consommation d'énergie. Pour répondre à cette problématique, cet article propose d'utiliser les graphes programmables intriqués ou Tangled Program Graph (TPG) à la place des NN. Les TPG sont des algorithmes d'apprentissage par renforcement dont l'évolution est basée sur des techniques de programmation génétique. Les précédents travaux sur les TPG présentent des performances d'apprentissage comparables aux NN de l'état de l'art, tout en nécessitant 2 à 3 fois moins de calcul et 3 à 5 fois moins de mémoire [4].

Les principales contributions de ce papier sont les suivantes :

- Utilisation d'un nouveau mécanisme d'apprentissage automatique appelé TPG pour l'identification RF.
- Comparaison de la vitesse de convergence d'un TPG et d'un CNN de l'état de l'art.

Le papier suit le plan suivant : la section 2 présente la modélisation d'une transmission sans fils considérant les empreintes RF, et introduit la classification des empreintes. La section 3 introduit les TPG particulièrement pour la classification. La section 4, décrit la base de données utilisée pour l'analyse et présente les expérimentations et résultats. Enfin la section 5 propose une conclusion.

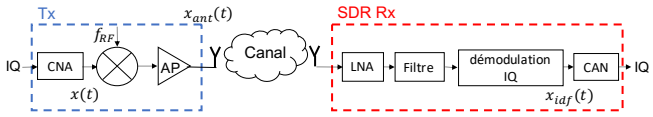

FIGURE 1 : Chaîne de transmission-réception.

2 Classification d'empreinte RF

2.1 Modélisation d'une chaîne de transmission et son empreinte RF

L'empreinte RF d'un transmetteur est une signature unique créée par les fonctions de distorsions des composants analogiques de la chaîne de transmission. Sur la partie gauche de la Figure 1, présentant la chaîne de transmission/réception, se trouve le transmetteur, il prend en entrée deux branches indépendantes et applique une modulation en phase (In phase) - en quadrature (Quadrature) (IQ). On retrouve ensuite le Convertisseur Numérique Analogique (CNA) permettant d'obtenir $x(t)$, puis l'oscillateur local (OSC) et enfin l'Amplificateur de Puissance (AP). L'ensemble des distorsions est appelé empreinte RF du transmetteur, notée $\mathcal{F}_{\mathrm{RFF_{Tx}}}$. Le signal émis $x_{\mathrm{ant}}(t)$ peut être modélisé par :

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

$$x_{\mathrm{ant}}(t) = \mathcal{F}_{\mathrm{AP}} \circ \mathcal{F}_{\mathrm{OSC}} \circ \mathcal{F}_{\mathrm{IQ_{mismatch}}} \circ \mathcal{F}_{\mathrm{CNA}}\left(x(t)\right), \quad (2)$$

où $\circ$ représente la composition de fonctions.

L'équation (2) met en évidence l'impact de chaque composant. $\mathcal{F}_{\mathrm{CNA}}$ représente les distorsions causées par le CNA. $\mathcal{F}_{\mathrm{IQ_{mismatch}}}$ modélise les imperfections induites par le modulateur IQ (appelées IQ imbalance) : les deux voix indépendantes du modulateur ne sont pas parfaitement orthogonales. $\mathcal{F}_{\mathrm{osc}}$ représente les deux imperfections de l'oscillateur, utilisé pour passer de la bande de base à la fréquence porteuse : i) un décalage en fréquence ii) un bruit de phase. Enfin, $\mathcal{F}_{\mathrm{AP}}$ correspond aux distorsions induites par les non-linéarités de l'AP, pouvant être modélisées par un modèle paramétrique comme un modèle de Rapp ou un modèle polynomial. Cette composition de fonction montre la difficulté d'extraction des caractéristiques des empreintes RF.

Sur la Figure 1, le bloc canal représente l'environnement de propagation sans fils qui peut impacter le signal notamment à cause des trajets multiples. Enfin, le récepteur est modélisé sur la droite de la figure, par un amplificateur faible bruit ou Low Noise Amplifier (LNA), un filtre, un bloc de démodulation IQ et un Convertisseur Analogique Numérique (CAN). Tout comme pour l'émetteur, l'ensemble des défauts des composants crée une fonction de distorsion, correspondant à l'empreinte RF du récepteur. Le signal IQ reçu et utilisé pour l'identification est noté $x_{\mathrm{idf}}(t)$ et peut être modélisé par :

$$x_{idf}(t) = \mathcal{F}_{\mathrm{RFF_{Rx}}} \circ \mathcal{F}_{\mathrm{canal}} \circ \mathcal{F}_{\mathrm{RFF_{Tx}}}\left(x(t)\right), \quad (3)$$

avec $\mathcal{F}_{\mathrm{canal}}$ représentant le canal de propagation et $\mathcal{F}_{\mathrm{RFF_{Rx}}}$ représentant la fonction de distorsion causée par le récepteur. L'impact du récepteur et du canal sur le signal complique l'identification de l'émetteur.

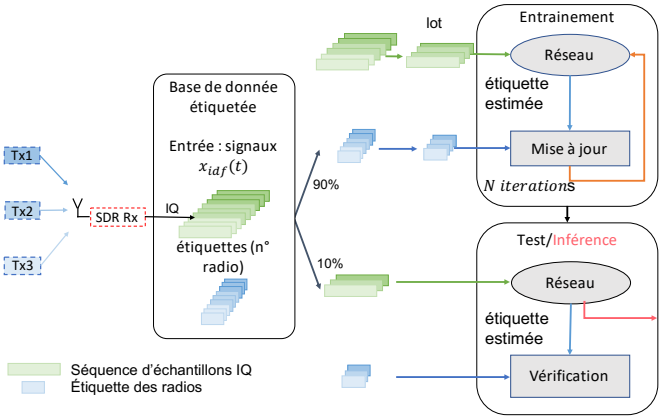

FIGURE 2 : Chaine de traitement des données pour des techniques d'apprentissage automatique.

2.2 Réseaux de neurones pour la classification

Le DL est de plus en plus utilisé pour répondre à des problématiques de classification et le domaine de l'identification par empreinte RF ne fait pas exception à la règle, [12, 8, 9]. La Figure 2 décrit la chaîne de traitement des données pour une classification par NN. La radio logicielle reçoit les séquences d'échantillons IQ et leurs attribue une étiquette en fonction de l'émetteur de la séquence (Tx1, Tx2 ou Tx3). La base de données est ensuite divisée en deux, 90% pour l'entraînement et 10% pour le test. En entraînement, le NN prend en entrée les séquences groupées par lot ou *batch* et estime l'étiquette correspondant à chaque séquence. Une fois toutes les séquences du lot vues, une mise à jour du réseau est faite par un mécanisme de descente de gradient et de comparaison des étiquettes estimées à la vérité terrain. Au cours de l'apprentissage, l'ensemble du jeu de données d'entraînement est présenté au réseau plusieurs fois, ce qui correspond au nombre d'epoch. Le mécanisme de mise à jour est répété pour chaque lot et chaque époch. En test, le réseau prend en entrée les séquences du jeu de données de test. Il estime l'étiquette associée à chaque séquence, qu'il compare à l'étiquette attendue pour obtenir un score, permettant d'évaluer sa capacité de classification, ici le score F1 [1].

Dans cet article, les séquences sont composées de 256 échantillons IQ, et un lot représente 600 séquences de différents émetteurs. Nous avons repris un réseau de l'état de l'art inspiré d'AlexNet et utilisé par Sankhe et al. [9] que nous nomerons Sankhe. Il est constitué de 4 couches convolutives de 128 filtres 7×1 et 5×1 et d'un étage de maxpooling. Enfin, 3 couches denses avec 256, 128 et 6 noeuds (le dernier nombre correspondant au nombre de classes) permettent la classification. Cette architecture est composée de 1 232 774 paramètres et nous servira de réseau de comparaison.

3 Tangled Program Graph

3.1 Introduction des TPG

Introduit en 2017, les TPG sont des modèles d'apprentissage par renforcement ou Renforcement Learning (RL) construit grâce à des techniques de programmation génétique. Contrairement aux NN dont la topologie est choisie par un expert en science de la donnée, les TPG sont construits au fil des

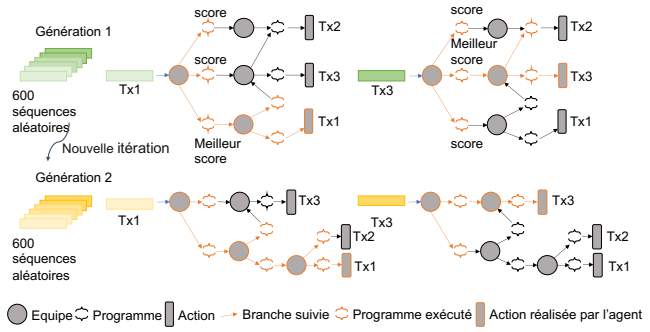

FIGURE 3 : Exemple et sémantique des TPG.

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 NN dans l'état de l'art, permettant une réduction de la complexité et de la quantité de mémoire nécessaire, en entraînement et en inférence [4].

Un TPG est structuré comme un graphe orienté dont les sommets et les arrêtes sont appelés respectivement *Equipes* et *Programmes*, ils spécifient un flux de contrôle d'un agent RL, et non un flux de données comme dans les NN. Un exemple de TPG est donné sur la Figure 3. Le flux de contrôle du TPG débute en racines à chaque nouvel environnement d'apprentissage (ici une nouvelle séquence d'échantillons IQ). Tous les *programmes* associés à la sortie d'une *équipe* sont exécutés. Un *programme* correspond à une séquence d'instruction simple telle que soustraction, multiplication, [...] prenant en entrée différentes variables de l'environnement et retournant une unique valeur par *programme*, cette valeur est le score. Quand tous les *programmes* associés à la sortie de l'*équipe* sont exécutés, l'exécution du TPG se poursuit suivant la branche ayant obtenue le meilleur score. Cette branche peut mener à une *action* de l'agent RL sur l'environnement. Les TPG sont construits par apprentissage de l'environnement, la complexité est ajoutée au fil des génération si cela amène un meilleur score, ici le score F1 [5].

Le processus d'entraînement du TPG est basé sur un algorithme génétique, qui est un algorithme d'optimisation bio-inspiré. Un graphe initial est créé de manière aléatoire, avec différentes racines, branches et *programmes* aléatoire également. Après l'évaluation des objectifs, chaque racine obtient un score pour les séquences vues et le graphe évolue en conservant les racines avec le meilleur score et en supprimant les autres. À chaque génération, de nouvelles racines sont créées, aléatoirement ou par copie ou mutation de celles ayant survécues. Les *programmes* sont également soumit à l'évolution génétique.

3.2 TPG pour la classification RF

Les TPG ont été introduits pour le RL, mais ils sont également utilisés en classification, et présentent des résultats intéressants sur la base de données CIFAR-10 [11].

La classification par TPG s'intègre facilement dans la chaîne de traitement des données présentée Figure 2, à la place du réseau. La phase de mise à jour correspond à l'évolution génétique du graphe, et chaque itération correspond à une nouvelle génération.

Au cours d'une génération, le TPG classe 600 séquences d'échantillons IQ qui ont été choisies aléatoirement. Ces séquences sont les données d'entrées de chaque racine du TPG pour cette génération. La topologie du TPG reste la même, et une *Action* correspond donc à la classification d'une séquence dans une classe

L'évolution génétique peut créer un graphe déséquilibré, pour lequel le score F1 des sous graphes est très différent, une classe pouvant être parfaitement détectée tout le temps et une autre jamais [11]. C'est pourquoi la mise à jour du TPG, en cas de classification, change pour conserver au moins un sous-graphe par classe [2]. Le processus de sélection naturelle a donc été modifié : lors de la sélection des n meilleures racines qui survivent pour la prochaine génération, $p\%$ des racines survivantes sont sélectionnées en fonction de leur score F1 moyen sur toutes les classes m, tandis que les autres $(100 - p)\%$ sont sélectionnées pour leur score F1 sur une seule classe. Dans la section 4, $p = 10\%$ est utilisé.

4 Expérimentations et résultats

4.1 Base de données

Il existe différentes base de données pour l'identification par empreinte RF comme Oracle [8], celle proposée dans [6] et DARPA [12, 7], Toutefois cette dernière n'est pas publique. Construire une base de données de signaux IQ est une tâche difficile car les conditions d'enregistrement, telle que la dynamique du canal, peuvent affecter la fiabilité de la détection.

Dans cet article, nous avons utilisé une base de données récente nommée WiSig [3]. Elle contient un grand nombre de signaux et d'information sur l'enregistrement des signaux (positionnement des émetteurs et des récepteurs et références des radios utilisées). WiSig est ainsi constituée de données WiFi capturées par 41 Périphérique universel de radio logicielle avec une bande passante de 20 MHz. Les signaux proviennent de 174 émetteurs WiFi sur quatre captures différentes effectuées sur un mois. À partir de ces radios les auteurs ont créé différentes bases de données [3]. Pour nos expériences, nous avons choisi le jeu de données *ManySig* avec 6 émetteurs et 12 récepteurs, et 1000 signaux pour chaque transmission.

4.2 Comparaison de performance

Cette section présente une comparaison des performances des TPG et celles du CNN présenté dans la section 2.2. Les deux algorithmes sont entrainés sur un processeur Intel i7-8850H à 2,60 GHz avec 6 cœurs et 12 threads et avec les extensions SSE4.2 et AVX2, noté CPU. Le CNN est également entrainé sur un GPU NVIDIA Quadro P1000. Le TPG n'est pas implémenté sur le GPU car sa structure non symétrique n'est pas adaptée à une telle architecture. La base de données WiSig offre de nombreux degrés de libertés (jour de capture des données, récepteurs) pour créer des sous-ensembles de données ou *dataset* représentant différents scénarios. Le *dataset* utilisé ici est divisé en 90%-10% pour l'entraînement et le test et comporte les signaux reçus par la radio 1, le jour 1. A noter qu'il s'agit d'un scénario favorable pour l'apprentissage et l'identification car il n'y a pas de diversité de canal de propagation ni de radio de réception.

TABLE 1 : Matrice de confusion obtenue pour le TPG en phase de test, scenario jour 1, Rx1

Réel \ Estimé	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	96.0	1.0	0.0	1.0	0.0	2.0
Tx_2	0.0	93.0	0.0	7.0	0.0	0.0
Tx_3	0.0	3.0	95.0	0.0	0.0	2.0
Tx_4	1.0	3.0	0.0	96.0	0.0	0.0
Tx_5	0.0	0.0	1.0	0.0	99.0	0.0
Tx_6	0.0	0.0	0.0	0.0	0.0	100.0

TABLE 2 : Matrice de confusion obtenue pour le CNN en phase de test, scenario jour 1, Rx1

Réel \ Estimé	Tx_1	Tx_2	Tx_3	Tx_4	Tx_5	Tx_6
Tx_1	100.0	0.0	0.0	0.0	0.0	0.0
Tx_2	0.0	87.5	0.0	12.5	0.0	0.0
Tx_3	0.0	30.0	70.0	0.0	0.0	0.0
Tx_4	0.0	0.0	0.0	100.0	0.0	0.0
Tx_5	12.5	0.0	0.0	0.0	87.5	0.0
Tx_6	0.0	0.0	0.0	0.0	0.0	100.0

Les Tables 1 et 2 donnent respectivement les matrices de confusion obtenues en phase de test avec le réseau TPG et le CNN Sankhe. Les lignes de la matrice de confusion correspondent aux vraies étiquettes tandis que les colonnes sont les étiquettes estimées par le réseau. Les nombres représentent le pourcentage obtenu pour chaque cas. Ces matrices montrent la capacité des TPG à étiqueter correctement les radios après une phase d'entraînement.

Enfin, pour évaluer les performances et valider l'intérêt d'utiliser les TPG, nous comparons la vitesse de convergence de la phase d'entraînement des deux réseaux. La Figure 4 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.

5 Conclusion

Cet article propose d'utiliser une nouvelle technique d'apprentissage automatique appelée TPG pour identifier les dispositifs radio à partir de leurs empreintes RF. Les résultats mettent en évidence la capacité des TPG à classifier les radios avec des performances similaires au CNN, tout en obtenant une progression rapide du score F1 du TPG pendant la phase d'entraînement sur CPU. Cette progression est très proche de la progression du score F1 du CNN de l'état de l'art sur GPU. Les TPG offrent la possibilité d'implémenter des algorithmes d'apprentissage automatisé pour l'identification d'empreinte RFF sur un matériel plus léger ne disposant pas de GPU.

Remerciements

Ce travail a été financé par l'intermédiaire d'une bourse CREACH Lab et par les projets ANR-22-CE25-0007-01 (ANR RedInBlack) et ANR-22-CE25-0005-01 (FOUTICS).

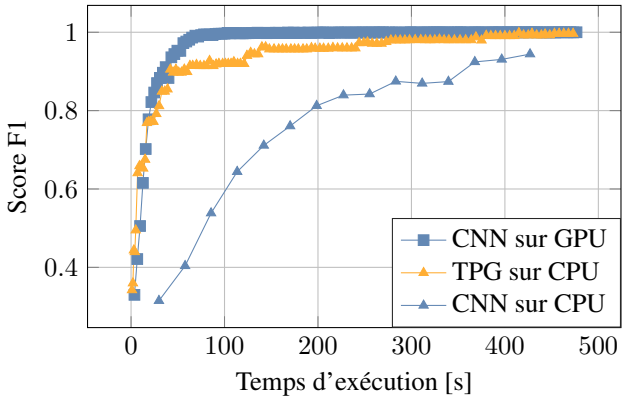

FIGURE 4 : Evolution de la vitesse de convergence des différents réseaux sur différents materiels, par observation du score F1.

Références

[1] D. CHICCO, G. JURMAN et AL. : The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics 21*, 2020.

[2] K. DESNOS, N. SOURBIER et AL. : Gegelati : Lightweight artificial intelligence through generic and evolvable tangled program graphs. *In DASIP Workshop*, International Conference Proceedings Series (ICPS), 2021.

[3] S. HANNA, S. KARUNARATNE et AL. : WiSig : A Large-Scale WiFi Signal Dataset for Receiver and Channel Agnostic RF Fingerprinting. Rapport technique, 2022.

[4] S. KELLY : Scaling Genetic Programming to Challenging Reinforcement Tasks through Emergent Modularity. 2018.

[5] S. KELLY, J. NEWSTED et AL. : A modular memory framework for time series prediction. *In Procs of the Genetic and Evolutionary Computation Conference (GECCO)*, 2020.

[6] C. MORIN et L. CARDOSO : Transmitter Classification With Supervised Deep Learning. *arXiv :1905.07923 [cs, eess]*, 2019.

[7] J. ROBINSON, S. KUZDEBA et AL. : Dilated Causal Convolutional Model For RF Fingerprinting. *In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) IEEE*, pages 0157–0162.

[8] K. SANKHE, M. BELGIOVINE et AL. : ORACLE : Optimized Radio clAssification through Convolutional neuraL nEtworks. *In 2019 INFOCOM IEEE Conference on Computer Communications*, pages 370–378.

[9] K. SANKHE, M. BELGIOVINE et AL. : *No Radio Left Behind :* Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments. *in 2020 IEEE Transactions on Cognitive Communications and Networking*, 6(1):165–178.

[10] G. SHEN, J. ZHANG et AL. : Radio Frequency Fingerprint Identification for Security in Low-Cost IoT Devices. 2021.

[11] R. SMITH, R. AMARAL et AL. : Evolving simple solutions to the cifar-10 benchmark using tangled program graphs. *In 2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 2061–2068.

[12] N. SOLTANI, K. SANKHE et AL. : More Is Better : Data Augmentation for Channel-Resilient RF Fingerprinting. *in 2020 IEEE Communications Magazine*, 58(10):66–72.