Siamese Network on I/Q Signal for RF Fingerprinting

Louis Morge-Rollet, Frédéric Le Roy, Denis Le Jeune, Roland Gautier

Siamese Network on I/Q Signal for RF Fingerprinting*

Louis Morge-Rollet[1], Frédéric Le Roy[1], Denis Le Jeune[1], and Roland Gautier[2][0000−0003−3570−1061]

[1] ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285, F-29200, Brest, France
{firstname.last.name}@ensta-bretagne.org
[2] Univ Brest, Lab-STICC, CNRS, UMR 6285, F-29200, Brest, France
{firstname.last.name}@univ-brest.fr

Abstract. RF Fingerprinting techniques aim to authenticate a wireless emitter by the imperfections due to these components. It can be useful for authentication and network management for the future IoT networks. Various methods has been proposed using hand-crafted features and classic machine learning but nowadays many researchers try to apply deep learning architectures for RF Fingeprinting. Our contribution is based on Siamese Network, a deep learning architecture widely used by the face recognition community. We use the deep learning architectures proposed by the RF Fingeprinring community which processes the I/Q (In-phase and Quadrature) signal and the siamese network learning paradigms developed for the facial recognition to propose siamese architectures for RF Fingerprinting. One of the main advantage of the siamese network is the possibility to use one-shot learning and its ability to require a few data for the final implementation of the network. In this paper, we explain our implementation, our results and discuss about the potential benefits of our approach for final implementation in a wireless network.

Keywords: RF Fingerprinting, Siamese Network, Deep Learning

1 Introduction and state of the art

Cybersecurity is a major concern of our epoch. Devices become more and more connected and cyberattacks are increasingly frequent and massive. The wireless technologies, such as WiFi, Bluetooth and Mobile networks are massively used. With the incomming of new technologies such as autonomous vehicules, smart grid, smart cities among others, the demand for connectivity will explose and require the use of new protocols such as 5G and IoT Networks (Zigbee, LoRa, ...). Many IoT protocols are based on low energy constraints but these technologies need to be secured. However the security measures such as cryptography

* This work was funded by ENSTA Bretagne of Brest and also supported by the IBNM (Brest Institute of Computer Science and Mathematics) CyberIoT Chair of Excellence of the University of Brest.

are difficult to implement for IoT protocols due to the complexity of keys management and energy consumption of standard cryptography [1].
The RF Fingerprinting is a part of *physical layer securities* [2] aiming to protect communications based on physical-layer properties. This technique consist in authenticate a wireless emitter using the specific impairments of these components. The manufacturing process has some uncertainties and two devices that seems to be similar have their own physical impairments. The impairments such as I/Q offset, I/Q imbalance, clock offset among others can be used to authenticate an emitter. The RF Fingerprinting is considered as a *Non-Cryptographic authentication* technique [3], however there is a debate in the community to know if it can replace the cryptographic authentication protocol (RSA, ...) or be used as a second factor for authentication [4]. RF Fingerprinting can also be used for intrusion detection [5] or to secure network layer against attacks [6]. Our approach based on deep learning architecture and siamese network will focus on authentication but it is possible to generalize it to other applications.

1.1 Deep Learning architecture

The machine learning (ML) is a part of artificial intelligence, based on algorithms (SVM, neural network, ...) able to learn how to solve a problem from data. Neural networks are bio-inspired mathematical models, they are composed of stacked layers (i.e parallel set) of basic unit called *neuron*, generally many layers (called *hidden layers*) are stacked to mimic the way that brain processes informations. The deep learning (DL) generally refers to neural network with two or more hidden layers. Many architectures, inspired from brain specific parts, have appeared over time like Convolutive Neural Networks (CNN) or Recurrent Neural Networks (RNN) to solve specific problem like, respecively, image recognition or time-series prediction.
Many methods has been described in the literature for RF Fingerprinting. Some methods focus on the transcient aspect of the signal [7], others on the steady-state aspects (also called *Modulation-based*) like [8] or the both aspects [9]. With the increasing popularity of deep learning, RF Fingerprinting community begins to use deep learning architecture on raw I/Q signals, specifically the CNN [1], [10], [11] and [12]. Futhermore, the DARPA has lauched in 2017 the program RFMLS[3] (Radio Frequency Machine Learning Systems) which aims to develop the use of machine learning for radio frequency. One argument of the RFMLS project is to develop the use of deep learning architecture to replace classic machine learning techniques based on expert hand-crafted features which are dependent on a priori asumptions [13].

1.2 Siamese network

A siamese network consists of two neural networks which have identical weights and their inputs are projected on a latent space where similarity measure are

[3] https://www.darpa.mil/attachments/RFMLSIndustryDaypublicreleaseapproved.pdf

applied (L_2 distance, ...) to know the similarity between the two inputs. The first application of siamese networks was for signature recognition [14] but that type of architecture is widely used by the facial recognition community [15], [16] and [17]. Other applications using siamese networks have also appeared like dimensionality reduction [18] or voice casting [19] among others. The siamese networks can be really useful in several cases: when a few data are available for the final implementation, when there is a lot of classes and for detection intrusion. G. Koch et al. [20] show the possibility of using siamese network for *one-shot learning* (i.e when there is only one learning example per class) for hand drawn characters recognition. Futhermore, Langford et al. [21] show the possibility of using siamese network on compressed spectrogram for specific emitter identification (a task similar to RF Fingerprinting), the authors also show the performance gains of siamese network compared to classic CNN for low SNR.
Several learning paradigms has been proposed to train a siamese network. The first approach, developed by LeCun et al. in [14], was based on cosine similarity. The most popular approach is based on *contrastive loss* [15], [18] and [19], which uses a specific loss which constraints the latent representation to respect some properties (see further explanation in section 2.2). G. Koch et al. proposed in [20] a siamese network learning paradigm as a *logistic regression* problem using *weighted* L_1 *norm* (see further explanation in section 2.2) which seems yielding better results than previous methods [21]. The previous approaches were considered as *end-to-end problems* but other approaches differ from it like DeepFace [16] which consists to use *transfer learning* (i.e transfer some knowledge learned from similar task to a new one) or the *triplet loss* [17] which consists to a specific *end-to-end problem* using three inputs instead of two.

1.3 Proposed approach

Our approach consist in using the deep learning architecture coming from the RF Fingerprinting community [1], [10] and [11] directly on I/Q signals collected over real-world measurements and the siamese network paradigm for RF Fingerprinting. This paper is composed as follow: Proposed method (section II), Experimental data analysis and results (section III), Potential benefits and further work (section IV) and Conclusion (section V).

2 Proposed Method

2.1 Dataset

Original dataset: The dataset [4] on which this study is based come from real-world measurements and was used by [11] and [1] to explore deep learning architectures for RF Fingerprinting. It was composed of 2 types of datasets: over-the-air and over-the-cable configurations. These datasets are composed of 16 identical USRP X310 SDR (Software-Defined Radio) platforms. Each emitter is

[4] http://www.genesys-lab.org/oracle (last visit the 26/08/2020)

recorded twice (*run 1* and *run 2*) for a duration of 4 seconds with a sample rate of 5 Ms/s which corresponds to 20 millions I/Q samples. The process is repeated for different distances in the range 2ft to 62ft with an interval of 6ft. The SDR receiving platform is always the same for each experiment: USRP B210. For our experimentation, the over-the-air configuration is chosen, which is considered more realistic. We use the 2ft recordings, which can be considered as a LOS path and with a really high SNR ($>$ 45 dB). Only the first 10.24 ms (4000*128 samples) of *run 2* for each emitter was used to create the database, which is considered to be of better quality than the *run 1*. Contrarily to [10] and [11] we use non-overlapping windows to extract the examples from the recording, which allows a better independance (in term of sampling) between all the examples. Our dataset, called *baseline dataset*, is composed of 16 classes of emitters with 4000 examples per class, which is considered enough for CNN classification.

Siamese dataset preparation: From the *baseline dataset*, a second dataset has been created to train the siamese network: the *siamese dataset*. The strategy used to create this dataset is inspired with the previous works on siamese network [14], [15], [16], [18], [19] and [20]. It is composed of a equal number of positive pairs (i.e two inputs from the same emitter) and negative pairs (i.e two inputs from different emitters). The process to create the dataset is the following; for a specific input of the dataset we choose N (here 5) inputs with the same class (without the corresponding input) using a sampling without replacement to create the positive pairs and we choose N inputs with different class using a sampling without replacement to create negative pairs. This process is repeated for each input of the dataset to create the *siamese dataset*. Concerning the train/test split of the dataset, the scikit-learn train_test_split function is used on *baseline dataset* and the process described above is applied separately on the training set and testing set.

2.2 Architecture and learning paradigms

The architecture used for this work (see table 1(a)) is inspired by an architecture from [11]. The network processes the I/Q signal as an 2x128 image (i.e 2 for I and Q and 128 for sample number) with one channel, corresponding to an input size of 2x128x1. In our experiment, we compare several learning paradigms. The first learning paradigm comes from [20] and consists as a *logistic regression* problem (i.e the output predict the propability that two inputs are similar) using a *weighted L_1 norm*. Indeed, an element-wise absolute difference is applied to the latent representations followed by a logistic regression (using binary cross-entropy loss): $\hat{y}(x_1, x_2) = \sigma(\sum_i \alpha_i |G_W(x_1)[i] - G_W(x_2)[i]| + \alpha_0)$ where $G_W(x_i)$ represent the latent representation of the input x_i.
The second learning paradigm called the *contrastive loss*, is based on the work of LeCun et al. [18] and the third called *contrastive transfer*, is based on the work of [16] using *transfer learning*, but instead of using a *weighted L_1 norm* like

[20], we used a *contrastive loss*. The *contrastive loss* proposed in [18], to train a *mapping function* (G_W) for a dimensionnality reduction purpose, consists to constrain the latent representations to respect some properties. Especially, as mentioned by [15] and [19], similar points (x_1 and x_2) need to be near from each others and the distance between disimilar points (x_1 and $x_1^{'}$) need to be greater than a specific constant called margin m (here 1). This constraint can be express like: $E(x_1, x_2) + m < E(x_1, x_1^{'})$ where $E(x_i, x_j) = \|G_W(x_i) - G_W(x_j)\|_2$ is the distance (using L_2 norm) between the projection of (x_i, x_j) in the latent space. The associated loss function is the following:

$$L(I_1, I_2, Y) = Y * \|G_W(I_1) - G_W(I_2)\|_2^2 + (1 - Y) * max(0, m - \|G_W(I_1) - G_W(I_2)\|_2)^2 \quad (1)$$

Where:

- I_i is an input and $G_W(I_i)$ his corresponding latent representation
- Y indicated if the pair are similar ($Y = 1$) or dissimilar ($Y = 0$)

The first two learning paradigms are *end-to-end problems* unlike the third one which is based on *transfer learning*. For this approach we have proceded as following: we train the network proposed in [11] for a K-class classification problem (considered as our *high-level caracteristics* extractor), we remove the two last layers (i.e the softmax and the last dense layer), add an other dense layer of 128 neurons and train only the last layer (the parameters of the other layers are fixed) using the previously introduced *contrastive loss*.

We used Adam optimizer on 32 epochs with batch size of 128. We used regularization to avoid over-fitting like l_2 regularization on each layer and a dropout of 50% at the first dense layer. The hyperparameters (see table 1(b)) have been found using grid search and hold-out validation for the learning rate μ, the l_2 regularization parameter l and the number of neurons D of the last dense layer.

(a) Neural network architecture

Layers	Characteristics
Input	(2, 128, 1)
Conv2D	50 filters (1x7) + ReLu
Conv2D	50 filters (2x7) + ReLu
Flatten	
Dense	256 neurons + ReLu
Dense	D neurons + ReLu

(b) Hyperparameters

	μ	l	D
Logistic regression	0.0001	0.00001	128
Contrastive loss	0.001	0.0001	128
Contrastive transfer	0.001	0.0001	128

Table 1: Neural network parameters

3 Experimental data analysis and results

We train our model using Keras framework on the school cluster (Intel Skylable Gold 6132). The metric used to evaluate the first learning paradigm is the accuracy. For the two others, we define a specific metric, which consists to compare the distance of the latent reprensentations to the half of the margin. If the distance is lower than the half of the margin the inputs are considered as a similar pair (Y=1), otherwise the inputs are considered as dissimilar pair (Y=0).

3.1 Experiments

The performances of the several learning paradigms are shown in the table 2. The *logistic regression* have better performances than the others learning paradigms, which confirms the conclusion of [21]. One can say that is unsual that training loss is greater than testing loss. But this phenomena, discuted by A. Géron in this post[5], can be explained in our case by the regularization applied during the training (i.e dropout and l_2 regularization).

Learning paradigm	Train accuracy	Test accuracy
Logistic regression	0.9909	0.9952
Contrastive loss	0.9669	0.9744
Contrastive transfer	0.9195	0.933

Table 2: Performances of learning paradigms

3.2 The dataset problem

The performances obtained are good although slightly below than [21] (reaching 99.79%). There may be several explanations to this lack of performances. First of all, the dataset used is not really suitable for a siamese network problem (contrary to Omniglot, [20]). Indeed, a classic siamese dataset consists of many classes with few examples per class which possed high inter-class variability. On the contrary, the *baseline dataset* is composed of few classes and many examples per class which is usually more adequate for K-class classification problems. Maybe the variability of dissimilar pairs are not large enough to train a good network. On the second hand, the impairments present in our *siamese dataset* is less controled than in the dataset used by [21] which seems coming from simulation, including 4 emitters and having a single impairment (frequency offset). Conversely, our dataset is based on real-world measurements on 16 differents emitters (USRP X310) with various impairments. Futhermore, our approach does not require pre-processing like time-frequency transform (used in [21]) and directly work on I/Q signals.

4 Potential benefits and further work

The main problem of deep learning architectures and more generally machine learning algorithms is what we defined as *scalability*, i.e a model needs to be retrained for a new group of unknow emitters which is not really scalable for IoT devices with computational and energy constraints. The majority of the work on RF Fingerprinting considered the problem as K-class problem with an relatively large amount of data to train the algorithm (approximatively a thousand exemples per class). These works are interresting because they proposed new architectures/algorithms for RF Fingerprinting but the authors rarely take into account the scalability problems introduce by theirs approaches. An other problem is that K-class classification is also not really performant when the number of emitters of the network is too large and changing over time: some emitters

[5] https://twitter.com/aureliengeron/status/1110839223878184960

can leave the network and new ones can join it. The last point is concerning the spoofing attack: if an architecture/algorithm is trained to recognize K emitters (legitime and known) how will it behave when an illegitimate emitter try to communicate on the network.
The main interest of a siamese network algorihtm is that it doesn't learn a classifier but an "advanced" similarity metric. This allows to train the siamese network on a big database which generalize well the variability of the emitter and use it as similarity metric with a K-Nearest Neighbors (KNN) algorithm for final implementation with unknow emitters. This approach has several advantages:

- The final implementation need at least one example per emitter: one-shot learning
- The architecture doesn't need to be retrained for the final implementation
- Outlier detection can be used to detect illegitimate emitters

This type of approach is widely use for facial recognition, where an input image is compared with a list of images, to identify the corresponding person (match) or an intruder (unmatch). It consists: to store the latent representations of known emitters (one per emitter), to compute the latent representation of the new input and to compute distance on the latent space to determine if the emitter belongs to the network (using a pre-determined threshold) and if that is the case to which emitter it belong. It is quite similar to 1NN (K=1) approach but with the concept of similarity metric replacing classical metric such as L_2 distance.
To our knowledge, only Ioannidis et al. [1] has proposed a method for one-shot learning for deep learning architecture based on an other type of approach. Our future work will explore the performance of deeper architectures for siamese network, complex-valued neural networks and others learning paradigms such as triplet loss. It will also be interesting to use 1NN algorithm (or more generally KNN) to test the performance of this approach from a *one-shot learning* point of view. Furthermore, we need to explore the performances under a range of SNR and multi-path environments.

5 Conclusion

The purpose of this article was to proposed a siamese approach for RF Fingerprinting based on the raw I/Q signal. We present the architecture of the network, the learning paradigms choosen and present the results on a real-world measurement dataset. We also introduced the potential benefits of this architecture for final implementation in a IoT network and some potential research works and improvements.
One of the main advantage of this approach compared to others (such as [21]) for RF Fingerprinting is that the network doesn't require preprocessing like time-frequency transform and directly works on I/Q signals. An other advantage of siamese network is their ability to perform *one-shot learning*. Use of deeper architectures and/or complex-valued neural networks (exploiting the complex nature of the signals) can further increase the obtained performances. This type of approach can be useful for final implementation, on IoT networks or more generally radio networks, to perform authentication and to allow a better and more flexible network management.

References

1. K. Sankhe , M. Belgiovine, F. Zhou , L. Angioloni, F. Restuccia, S. D'Oro , T. Melodia, S. Ioannidis and K. Chowdhury: No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments, IEEE Trans. on Cognitive Communications and Networking, Vol. 6, NO. 1, 2020.
2. Y. Zou, J. Zhu, X. Wang and L. Hanzo: A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends, Proce. of the IEEE, Vol. 104, 2016.
3. K. Zeng, K. Govindan, and P. Mohatara,: Non-cryptographic Authentication and Identification in Wireless Networks, IEEE Wireless Communications, Vol. 17, 2010.
4. Pieter Robyns, Eduard Marin, Wim Lamotte, Peter Quax, Dave Singelée, Bart Preneel: Physical-Layer Fingerprinting of LoRa devices using Supervised and Zero-Shot Learning, Wisec'17, 2017.
5. Nam Tuan Nguyen, Guanbo Zheng, Zhu Han and Rong Zheng: Device Fingerprinting to Enhance Wireless Security using Nonparametric Bayesian Method, INFOCOM, 2011.
6. S. Capkun K.B. Rasmussen. Implications of radio fingerprinting on the security of sensor networks. SecureComm, 2007.
7. O. Ureten and N. Serinken: Wireless security through RF fingerprinting, Canadian Journal of Electrical and Computer Engineering, vol. 32, no. 1, pp. 27–33, 2007.
8. V. Brik, S. Banerjee, M. Gruteser and S. Oh: Wireless Device Identification with Radiometric Signatures, MobiCom'08, 2008.
9. H. Yuan, Z. Bao, A. Hu: Power Ramped-up Preamble RF Fingerprints of Wireless Transmitters, Radioengineering, Vol. 20, n. 3, SEPT. 2011, 703–709.
10. S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury: Deep Learning Convolutional Neural Networks for Radio Identification, IEEE Communications Magazine, 2018.
11. K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis and K. Chowdhury: ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks, INFOCOMM, 2019.
12. E. Mattei, C. Dalton, A. Draganov, B. Marin, M. Tinston, G. Harrison, B. Smarrelli and M. Harlacher: Feature Learning for Enhanced Security in the Internet of Things, GlobalSIP, 2019.
13. A. Ghasemi, C. Parekh and P. Guinand: Spectrum Awareness Under Co-Channel Usage Via Deep Temporal Convolutional Networks, WinCOMM, 2019.
14. J. Bromley, I. Guyon, Y. LeCun, E. Sickinger and R. Shah: Signature Verification using a Siamese Time Delay Neural Network, NIPS-94, 1994.
15. S. Chopra, R. Hadsell and Y. LeCun: Learning a Similarity Metric Discriminatively, with Application to Face Verification, CVPR, 2005.
16. Y. Taigman, M. Yang, M. Ranzato and L. Wolf: DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR, 2014.
17. F. Schroff, D. Kalenichenko, J. Philbin: FaceNet: A Unified Embedding for Face Recognition and Clustering, CVPR, 2015.
18. R. Hadsell ,S. Chopra and Y. LeCun: Dimension Reduction by Learning an Invariant Mapping, CVPR, 2006.
19. A. Greese, M. Quillot, R. Dufour, V. Labatut and J. Bonastre: Similarity Metric Based on Siamese Neural Networks for Voice Casting, ICASSP, 2019.
20. G. Koch,R. Zemel and R. Salakhutdinov: Siamese Neural Networks for One-shot Image Recognition, 2015.
21. Z. Langford, L. Eisenbeiser and M. Vondal: Robust Signal Classification Using Siamese Networks, WiseML, 2019.