Learning to fingerprint: physical layer identification

Cyrille Morin[1], Thibaut Vial, Jakob Hoydis[2], Leonardo S. Cardoso[1], Jean Marie Gorce[1]

[1] Citi-lab/Inria
[2] Nokia Bell Labs

We investigate the use of a neural network to identify the transmitter of a random modulated message, without any preamble or identifying code. The Cognitive Radio Testbed [1] (FIT/CorteXlab) is used to collect experimental datasets in a reproducible way to train the neural network.

Keywords: Deep Learning, Transmiter Identification, GNU Radio, FIT/CorteXlab

1 Introduction

In all digital communication protocols, packets are always sent along with some sort of identification code. That code adds data to send over valuable radio space. In Internet of Things (IOT) protocols such as SigFox, the issue is not throughput but efficiency while sending small amount of data, So packets are getting smaller and smaller but the header stays the same, to the point up to 50% of sent data is from the header and not user data.

Another aspect of this is that, on the physical layers, it is currently extremely easy to impersonate a given device simply by copying its identification code. An identification scheme working on RF characteristics of received signal[2] would allow for the diminution of packet sizes and so either the diminution of the cost to send a packet or an increase in the payload capacity. And it would also allow for an increase in security since a potential attacker would need to reproduce hardware characteristics of the attacked radio chip.

2 Experimental setup

Neural network

We demonstrate the use of a neural network (NN) to classify the emitter of received packets. In our case, the neural network is tasked of classifying which emitter amongst 21 possible has sent a given packet. This NN is set to use raw complex samples gathered directly after a software defined radio (SDR) device (in our case a USRP) without any signal processing. The NN is a 4 layers network with two convolutionnal and two fully connected layers. Just before processing by the neural network, the input samples from a packet are normalized with a L_2 norm.

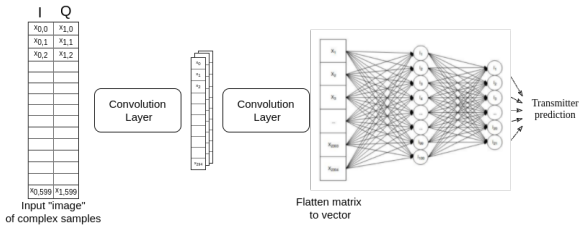

Figure 1: Architecture of the Neural Network

Datasets

We use the FIT/CorteXlab experimental platform to generate reproducible datasets. This platform consist of 38 nodes inside an isolated and anechoic chamber, each node being a computer associated with a SDR device (USRP or PicoSDR). This means that we can experiment on any frequency up to 6GHz without causing or suffering from any interference from the outside world.

The datasets are generated by sending random packets modulated in QPSK from the 21 different transmitters and recording them on a unique receiver. GNU Radio is used to build simple flowgraphs to emit and record packets for offline processing by the neural network. We plan to publicly release the datasets and the script used to generate them so that anyone can freely reproduce this experiment and compare their algorithms.

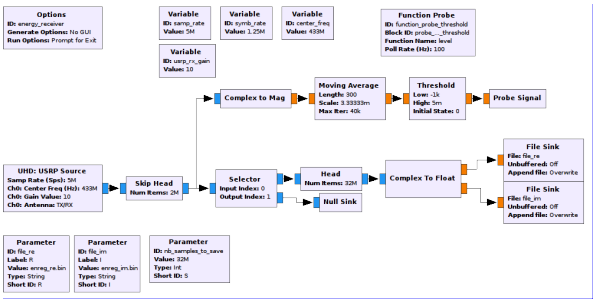

Figure 2: Recording GNU Radio flowgraph

In the dataset generation process, each transmitter sends 50000 random packets one after another and the emission power is the same for every

transmitter. The various transmitters are located at different points in the FIT/CorteXlab room so the distance to the receiver, and thus the attenuation of the signal, is different for each emitter. To reduce the effects of this, we also generate datasets where the emission power is set to vary.

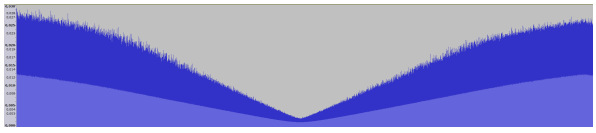

Figure 3: Variation of the received power over the 50000 received packets from one transmitter

Training

Generated datasets are split between training, valitation and testing data with 70% of the examples for training, 10% for validation and 20% for testing. The training is done by batches of 100 examples at a time with 40k iterations of the training process taking about half an hour on a i7-7820HQ CPU.

3 Results and ongoing works

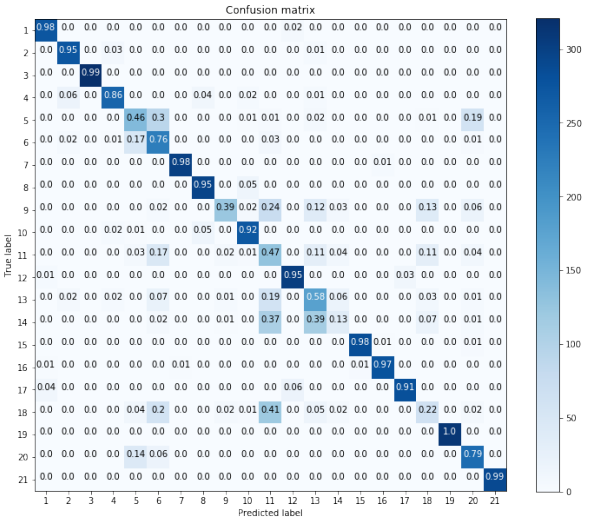

Figure 4: Confusion matrix of a network trained after 200k iteration from a dataset with varying emission power

Two metrics are important in this case: the ability to learn from a dataset, and the performance over datasets generated with different parameters. When data is not normalised and when the emission power is static, the network is able to learn with up to 99% accuracy in 40k iterations of the training process but is not able to perform on datasets with different power settings. We achieve an accuracy of 80% with a network trained on 200k iterations with varying emission power and normalisation with only a 10 point decrease in accuracy when tested on datasets with different power settings.

But if there is some kind of change in the experimentation room, the networks are completely lost. This means that it's heavily reliant on the caracteristices of the channel to distinguish different emitters.

To reduce the dependency of the classification over the channel characteristics, we will introduce a robot (a Turtlebot) in the FIT/CorteXlab room and have it wander randomly in the room with metal reflectors.

4 Conclusion

We have demonstrated the use of a simple neural network to identify the sender of random packets from raw samples without any identification code. This shows interesting prospects in reducing physical layer overhead payload and increasing security in settings where mobility is not present.

References

[1] A. Massouri, L. Cardoso, B. Guillon, F. Hutu, G. Villemaud, T. Risset, and J.-M. Gorce, "Cortexlab: An open fpga-based facility for testing sdr & cognitive radio networks in a reproducible environment," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014, pp. 103–104.

[2] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Using the Physical Layer for Wireless Authentication in Time-Variant Channels," jul 2009. [Online]. Available: http://arxiv.org/abs/0907.4919http://dx.doi.org/10.1109/TWC.2008.070194