HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Department: Information and communication technologies
Faculty: Information technology and communication systems
Orientation: Software engineering

# Bachelor Thesis

## RF fingerprinting on NFC devices

## Intermediary report

**Not confidential**

| | |
|---|---|
| **Student:** | Luc Wachter |
| **Project proposed by:** | Joël Conus<br>Kudelski Group SA<br>22-24, Route de Genève<br>1033 Cheseaux-sur-Lausanne |
| **Teacher in charge:** | Alberto Dassatti |
| **Academic year:** | 2019-2020 |

Yverdon-les-Bains, 19th June 2020

# 1 Initial project description

Radio Frequency (RF) fingerprinting is a technique that allows the identification of radio transmitters (such as Internet of Things (IoT) devices) by analysing the spectrum of their transmissions. Indeed a device's spectrum is unique because of tiny imperfections in the manufacturing process of its analog components. This analysis can typically be performed using machine learning algorithms.

Near-Field Communication (NFC) technology is often used in access control and payment applications but many implementations are vulnerable to relay attacks. This type of attacks allows an attacker to relay messages between a reader and an NFC device without the knowledge of the device's owner, effectively convincing the reader it is communicating with the legitimate device. Research and tools that facilitate such attacks are publicly available.

The goal of this project is to determine if RF fingerprinting could be used as an authentication technique against relay attacks.

The main steps of this project are the following:

- Build a simple lab setup with Software-Defined Radio (SDR) equipment to acquire signals between an NFC device and its reader
- Acquire RF spectrum data of various NFC devices
- Analyse the signals
- Classify the signals of the devices by using supervised machine learning classification techniques in order to differentiate trusted devices from attacker / relay devices
- Determine if this identification technique could be used as an authentication feature against relay attacks

As the receiving equipment (SDR) has an influence on the recorded signals, for this project we consider a single receiver to record the RF samples. Similarly, the lab setup should be built to provide an ideal low-noise & low-interference environment to simplify the analysis phase.

The expected deliverables are the following:

- A tool able to identify NFC devices by analysing the RF spectrum of their signals, at least in an ideal environment and with a small number of devices
- A detailed account of the steps taken and the setup used (as part of the report)
- An analysis of the results (as part of the report)

Collaboration with other researchers in this field is wished (EPFL, ElectroSense).

# Table of contents

## List of Figures

## List of Tables

# 2 Introduction

## 2.1 Project description

Radio Frequency (RF) fingerprinting is a technique that allows the identification of radio transmitters by extracting small imperfections in their spectrum. These imperfections are caused by tiny manufacturing differences in the devices' analog components. Using Software-Defined Radio (SDR) equipment, we can analyse this spectrum in order to extract the aforementioned differences and identify a device.

Such techniques can be used on any type of radio transmission: Bluetooth, BLE, WiFi, LTE, etc. This project aims to use RF fingerprinting on NFC devices. Indeed, NFC is often used in access control and payment applications but many implementations are vulnerable to relay attacks. Spoofing the imperfections in an emitter's radio spectrum is close to impossible at the present time, since it is essentially a hardware signature. This is why a technique like the one described here would be a valuable additional security layer.

The goal of this project is to determine whether applying machine learning techniques to the problem of RF fingerprinting NFC devices could be used as an authentication technique, in order to prevent relay attacks. If a dataset of sufficient quality and variety can be produced, it could be another outcome of the project. Indeed, while some exist for 802.11 communications, no dataset seems to be available for raw recordings of NFC transactions.

## 2.2 Context

This project is conducted in the context of my bachelor thesis at HEIG-VD.

- Department: Information and communication technologies
- Faculty: Information technology and communication systems
- Orientation: Software engineering

It was proposed by Mr Joël Conus of Kudelski Group SA.

## 2.3 Document description

This intermediary report marks the middle of the project. Because of this, it is firmly anchored in the analysis and conception phases, which means much of what is presented is subject to change in the second half of the project.

Nevertheless, this document describes the research done while studying the state of the art. It then presents the acquisition setup and the results it brought, before showing the steps undertaken to validate the captured signals through decoding. Finally, it showcases the first conception ideas and decisions made for the learning model, in light of our study of the state of the art.

# 3 Necessary concepts

Before analysing the problem and the existing works on the matter and before describing the hardware used and the data captured, it is worth explaining some of the concepts behind SDR and NFC. We will only discuss the elements needed to understand this document, and will refer the reader to other works for details. Readers that are acquainted to software-defined radio and NFC can skip this section.

## 3.1 Software-defined radio

First, we will note that radio waves are electromagnetic radiations operating in the Radio Frequency (RF) range. RF is a portion of the electromagnetic spectrum generally comprised between ∼3kHz and 300GHz [1]. It is used for radio communications of all sorts.

The main principle behind software-defined radio, as its name suggests, is to make as many of the elements of a traditional radio's pipeline digital. Doing this makes it possible to use a computer's processor to do the signal processing tasks that once required specialized hardware. Of course, it is not functional to simply strap an antenna to an Analog-to-Digital Converter (ADC) and do everything else in software. We still need some analog components in front of the ADC to preprocess the signal and ensure a consistent sampling. [2, 3, 4]

These analog components are combined in SDR hardware available to buy. Popular examples include the cheap RTL-SDR dongle[1] (receiver) and the HackRF[2] (transceiver). Once an SDR is plugged into a computer, the proper drivers and software are installed and an antenna is connected to the device, anybody is able to receive and process a wide range of frequencies. This range is limited by the components in the SDR hardware and by the antenna.

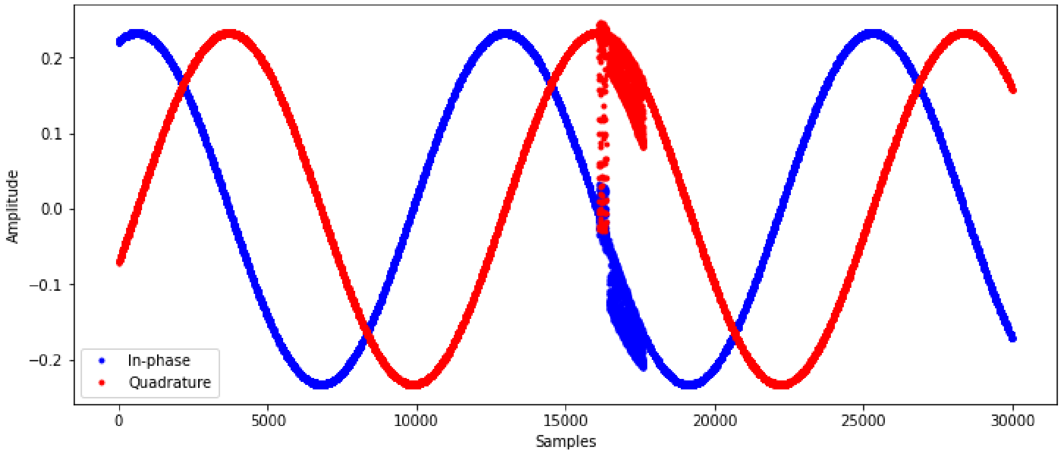

Figure 1: In-phase and quadrature signals

To finish this introduction to SDR, we will describe the representation of the data. Figure 1 shows a graph of the samples returned by the SDR pipeline as dots. As can be seen, the signal is represented using two components. The ”quadrature” component is shifted by 90 degrees in relation to the ”in-phase” component. Every sample can be represented by a complex number where the real part adds to the in-phase component and the imaginary part adds to the

[1] https://www.rtl-sdr.com/about-rtl-sdr

[2] https://greatscottgadgets.com/hackrf/one

quadrature component. Such a representation allows us to get a lot more information from the signal. [5, 6]

Here, the X axis represents the sample number (starting at 0) rather than a time value. The link between the sample number and the time elapsed is the sampling rate ($F_s$), expressed in number of samples per second (S/s). For example, considering a sampling rate of 3MS/s, figure 1 shows a signal over 10ms: $\frac{30000[S]}{3000000[S/s]} = 0.01[s]$.

In this document, we will often use the magnitudes representation of a signal. Said representation is built by taking the magnitude (or module) of each complex sample and plotting it. Figure 2 shows the magnitudes of the samples in figure 1.

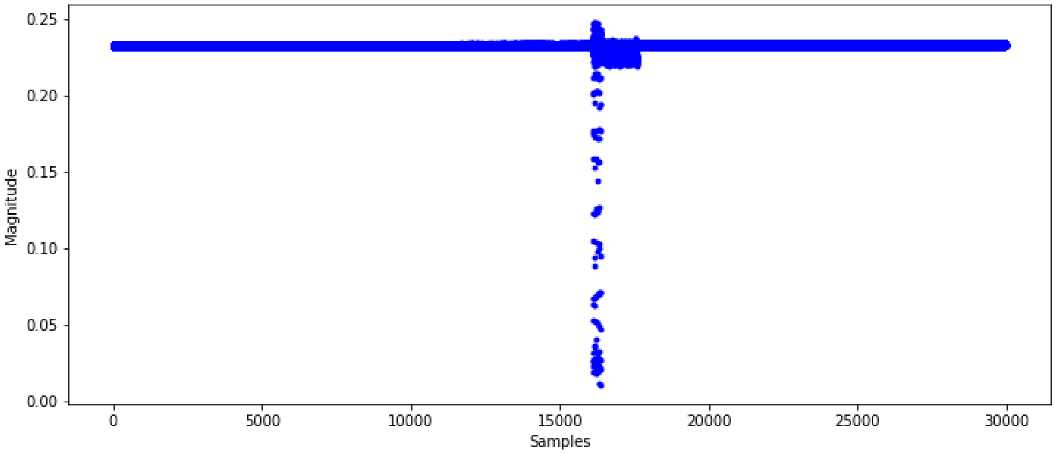

Figure 2: Signal represented as magnitudes

## 3.2 Near-field communication

In order for our radio system to be effective, it needs to be adapted to the protocols used. This is why we need to understand how NFC operates.

NFC is the name for a group of communication protocols designed for small distance (max. 10cm) transactions. The idea is that a reader can supply power to a passive tag and get the information stored in it. The reader devices are sometimes called initiators or PCD (for Proximity Coupling Device) and the tags are sometimes called targets or PICC (for Proximity Inductive Coupling Card).

Operating at a frequency of 13.56MHz, NFC is well in the High Frequency (HF) range of 3 to 30MHz. This is substantially lower than most communication protocols like WiFi (2.4GHz or 5GHz), Bluetooth (2.4GHz) or ZigBee (868MHz, 915MHz or 2.4GHz).

Also in contrast to these other protocols, NFC's modulation scheme is On-Off Keying (OOK) which is a form of Amplitude Shift Keying (ASK). (Except for NFC type B, which uses BPSK in target to initiator mode.)

Because it is designed to work only in close proximity, NFC uses inductive coupling between devices to transmit information. In simple terms, this means the reader can only send a signal as far as its generated magnetic field goes. The passive tag responds by modulating the same magnetic field. It also means that interferences from radio devices operating in the same frequency range practically don't affect an NFC communication. [7]

# 4 State of the art

## 4.1 Taxonomy

It is certainly useful to start with a review of the different ways to categorize the features and algorithms used by researchers in the field of radio frequency fingerprinting. More specifically, we'll focus on Radio Frequency Machine Learning (RFML) research, though other fingerprinting techniques exist that don't involve machine learning. The goal is to define our needs precisely and select the important things to consider.

### 4.1.1 Taxonomy for features

The features we select must allow us to identify a precise device among potentially very similar devices. We need what Delgado et al. [8] describe as a Physical Unclonable Function (PUF). PUFs are physical distortions that are unique to a specific system. They are another way of talking about fingerprints.

Xu et al. [9] propose three ways to categorize radio signal features:

- based on the specificity of the feature (from vendor specific to device specific),
- based on the layers (PHY, MAC, Network and higher),
- and based on the acquisition method (passive or active).

Features from the MAC and higher layers typically require in depth knowledge of the protocols in play. Not only that, but they also tend to be less specific than we would like (either vendor specific or depending on the type of device). This indicates we should probably focus on the physical (PHY) layer features, which rely on imperfections in the manufacturing process of the devices.

### 4.1.2 Taxonomy for fingerprinting algorithms

Riyaz et al. [10] provide a visual categorization of fingerprinting approaches, which we adapt in figure 3. We take a look at these approaches in the following paragraphs.

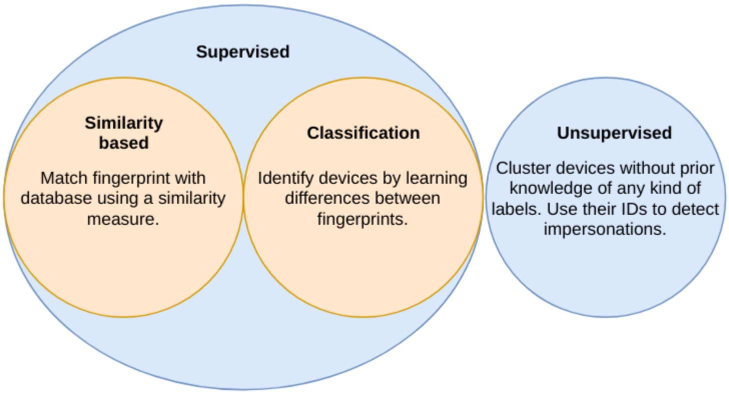
Supervised
Similarity based
Match fingerprint with database using a similarity measure.
Classification
Identify devices by learning differences between fingerprints.
Unsupervised
Cluster devices without prior knowledge of any kind of labels. Use their IDs to detect impersonations.

Figure 3: Fingerprinting algorithms taxonomy

**Supervised approaches:** Supervised approaches use features from labelled data to generate a function capable of separating the different classes. These approaches can be further categorized in similarity based and classification techniques.

Similarity based techniques are white-list algorithm that use a database of fingerprints and a similarity measure to determine whether a device is legitimate. Developing a technique like this usually requires prior knowledge of vendor specific device features [10].

Classification systems can also require deep knowledge of the features and protocols used, in the case of ”traditional”, manually tuned classifiers. Those are built to extract predetermined features and work similarly to other white-list algorithms afterwards.

In this age of deep learning though, research seems to be more interested in classification techniques that are able to extract the features they need by themselves. This can be done through deep Multi-Layer Perceptrons (MLP) [8, 11] or through more advanced techniques like Convolutional Neural Networks (CNN) [10, 12, 13, 14, 15]. The latter have proved very powerful in domains like computer vision, natural language processing and recommendation systems. This success is one of the reasons experimentations on CNNs are common in RFML research.

**Unsupervised systems:** Unsupervised approaches cannot by themselves discriminate a legitimate device from an illegitimate one. They don't have that information, since they work with unlabeled data. In order to detect attempts of impersonation, such a system has to keep a record of fingerprints and linked identifiers (MAC addresses, serial numbers...). It can then throw an alert and update a black-list when multiple fingerprints are linked to the same ID or when multiple IDs are linked to the same fingerprint [9, 16].

Xu et al. [9] specify that unsupervised approaches are appropriate when the fingerprints of legitimate devices are not available.

## 4.2 Acquisition

The vast majority of the research considered for this project uses USRP systems to record transmissions as raw I/Q signals. The number of devices can be anywhere from 5 to 500 (but most often less than 20). They all use data acquired from WiFi (802.11) or Zigbee (802.15.4) devices. [10, 12, 13, 14, 15, 16]

Some, like Sankhe et al. [15], also describe how they add artificially induced impairments to simulated signals with MatLab. This is something we could also do, but it is secondary. The focus is on the creation of a robust dataset of real world transmissions between an NFC reader and tags.

## 4.3 Features

### 4.3.1 Features selection

Even if we don't plan to manually select and extract the features that will form the fingerprints of our devices, it is useful to learn about them. It will allow us to make sure they are present for the algorithm to extract and also allow us to design preprocessing methods that magnify the features.

Whether we end up with a system that is able to identify many devices uniquely, or one that only tries to separate a specific device from the others, we will need device specific accuracy. We don't want to make relay attacks impossible only if the attacker doesn't use a device from the same vendor as the victim's device. This is why in section 4.1.1, we concluded that the features we are most interested in are from the physical layer.

Because of their nature, these features should be appropriate no matter the protocol used. The following list is composed of features described by Riyaz et al. [10] and also used in other works.

- I/Q imbalance: The amplitude and the phase are not exactly the same on the in-phase and the quadrature signals, because of the imperfections in the quadrature mixers.
- Phase noise: When the baseband signal is up-converted to the carrier frequency, it is sensible to phase noise which creates rotational vibrations.
- Carrier frequency offset: The difference between the carrier frequency of the transmitter and the carrier frequency of the receiver.
- Harmonic distortions: The Digital-to-Analog Converters (DAC) cause harmonic distortions because of imperfections.

### 4.3.2 Preprocessing

It is clear that preprocessing the data appropriately can greatly increase the accuracy of a model and reduce its complexity.

The first question to ask is how should the data be partitioned, in order to be fed to the learning algorithm. This question will be discussed in section 4.5.1 since it is identical to choosing the input of our model.

Several works mention wavelet transforms (either discrete or continuous) as effective means to amplify the characteristic features of a wireless device [9, 12, 13]. They report increased accuracy and scalability, and reduced complexity. It is certainly worth it to explore this preprocessing method.

## 4.4 Machine learning applied to radio frequency

### 4.4.1 Challenges

Riyaz et al. [10] highlight some of the challenges faced when working on RFML problems. They are reformulated in the three first items of the list below. The next items are additions.

1. Finding the optimal partition length to feed the learning algorithm.
2. Finding the optimal network architecture for the problem (in the case of neural networks).
3. The absence of standard datasets to train and evaluate a model.
4. Finding a cheap preprocessing method that improves performance and reduces complexity.
5. Achieving a demonstrably scalable system.

Great insight into item 1 is given by Youssef et al. [13]. Indeed, they compare the performance of models trained with different input segment sizes. We delve deeper into this matter as well as item 2 in section 4.5.

Item 3 is the core matter of the first part of our project. Indeed a considerable portion of our work consists in the elaboration of a dataset of NFC transmissions for different PICC devices.

### 4.4.2 Supervised or unsupervised

The aim of this project is to explore supervised learning techniques. This is why most of the literature considered here studies supervised systems. Also, in general, it seems works that use supervised methods are a lot more abundant than their unsupervised counterparts. This could be because of the popularity of models such as convolutional neural networks, and their apparent adaptability to the problem of RF fingerprinting.

Despite this, we can see that unsupervised learning techniques are also showing promising results. An example of this is the work of Nguyen et al. [16]. The paragraph about unsupervised systems in section 4.1.2 describes the basics of their system pretty well. They use a Nonparametric Bayesian model to detect the number of devices and then cluster them based on their fingerprints. This technique allows them to discriminate between an unknown number of devices that the model never encountered before. They show good results with four devices using two features strictly from the PHY layer.

While this work is interesting, it uses pre-engineered features. We were unable to find research on deep unsupervised learning techniques (such as deep belief networks) for the problem. Such a solution could be an interesting topic for another project.

### 4.4.3 Comparing supervised approaches

Several articles have compared the performance of different machine learning approaches. In the next paragraphs we discuss the findings of two of them.

Riyaz et al. [10] compared their convolutional neural network with the techniques of Support Vector Machine (SVM) and logistic regression. They report substantially higher performance using their CNN (up to 60-70% better accuracy). Their results, although limited in the number of devices (they only classify up to five), also seem to show that CNNs are more scalable than the other methods. Indeed, increasing the number of classes caused a significant drop in the performance of the other algorithms, but not for the CNN.

Youssef et al. [13] compare the performance of four supervised algorithms: SVM, Deep Neural Networks, CNN and Accelerated Levenberg-Marquardt Multi-Stage Training (A-LM MST). The latter is an advanced classification technique that makes training deep neural networks less resource heavy by using a hierarchy of smaller Multi-Layer Perceptrons (MLP) rather than one big MLP. They use data collected from 12 WiFi transmitters.

Their results show impressive performance for their MST, especially when provided with very little data. Their CNN model is close second, although it performed significantly worse with very little data. Then comes the DNN with pretty good results and then only the SVM.

These results can guide us in our choices of experimentation. They show that SVM algorithms, while capable to some extent, are not appropriate for this specific problem. They also show CNNs are very promising, more scalable and more adaptable than "simple" DNNs. The MST solution is very interesting, but reducing the computational complexity is not our primary concern and the improvement doesn't seem that substantial.

## 4.5 Neural network architectures

Studying the various architectures employed in previous works can help us define what works and what direction to take for our own experiences. This is what we will do in this section.

### 4.5.1 Input format

In section 4.4.1, we noted that one of the challenges we face is to find the optimal partition length for the input of our model. To give an element of answer, Youssef et al. [13] don't only compare the performance of 4 different models, they also do it with five different partition sizes (32, 64, 128, 256 and 512). They find their DNN architecture handles short segments better than long ones, while their CNN gets progressively more performant as the segments get larger.

Input format...

### 4.5.2 Hidden layers

Once the input is decided, the rest of the architecture is still in question...

Complex networks

output: noise, legitimate, other

incremental learning

CNNs - good shift invariance thanks to pooling layer?

# 5 Dataset creation

The first step of any classification project, especially if it uses machine learning, is to acquire and refine data. As the project was proposed, it was already clear that no existing dataset would be available. This proved to be true as discussed in section 4, where we see that existing research focuses on WiFi and Zigbee technologies.

## 5.1 Radio setup

This section's goal is to describe the material used to capture the communications between NFC readers and tags. The final setup is illustrated in figure 4.

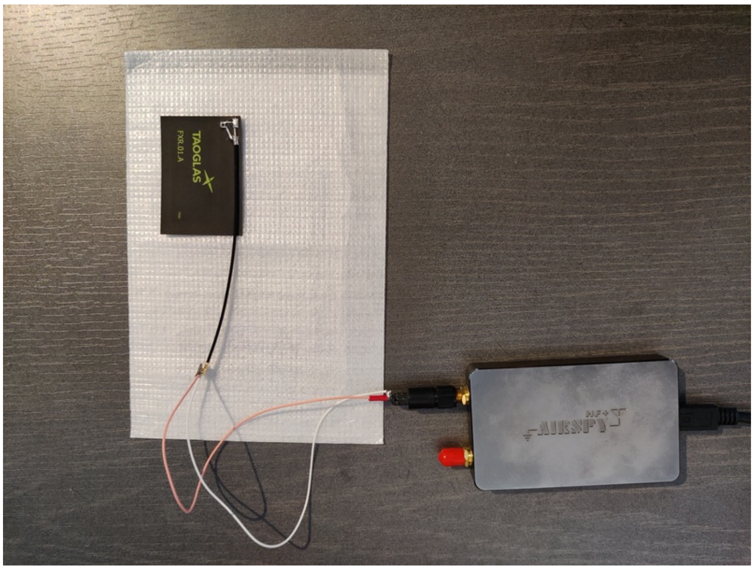

Figure 4: SDR and antenna setup

### 5.1.1 The SDR

LimeSDR and its shortcomings (heat, instability, freq range, sampling rate)...

Airspy HF+: made for HF, a lot more stable, noise level a lot lower, configurable...

768 kHz, 384 kHz, 192 kHz, 96 kHz and 48 kHz

### 5.1.2 The "antenna"

As described in section 3.2, NFC uses inductive coupling rather than the more common far field electromagnetic radiations. Because of this, our system needs a near-field antenna, which in this case is really just an inductor. A simple loop of copper wire qualifies as such, but in order for our antenna to be perfectly tuned to 13.56MHz, we used an industrial antenna: the Taoglas FXR.01.A[3].

The antenna should be placed at least 15mm away from metallic objects, for they interfere with the magnetic field used for the communication.

[3] https://www.taoglas.com/product/fxr01-nfc-flex-reader-antenna

As can be seen in figure 4, the adapter between the SDR and the antenna is homemade using spare connectors and copper wires. The risk of interferences because of this rather unsophisticated adapter is noted, but doesn't seem to be significant later in the work.

## 5.2 Inventory of devices

Here, we list the devices used to create the dataset. These include the PCDs (readers) and the PICCs (tags) whose communications were captured.

In terms of readers, table 1 lists the few devices used. We weren't able to find a specific NFC chip in either smartphone's characteristics. `reader2` was mostly used during the analysis phase, with the LimeSDR Mini system. The final dataset will be created with the help of `reader1`.

| Name | Type | Model |
|---|---|---|
| **reader1** | Smartphone | OnePlus 8 |
| **reader2** | Smartphone | Nokia 7+ |

Table 1: Inventory of PCD devices

On the other hand, the list of tags and their technical details can be found in table 2. A picture of tags 1 to 7 is also provided in figure 5. As the table shows, tags 1 to 5 use the exact same chip model. Tags 1 to 8 are all NFC type A compliant, while tag 9 uses the FeliCa standard from Sony. It will be interesting to contrast the classification performance between tags of the same type and between tags of different types.

On the PICCs that are marked writable, the content is harmonized to ensure the algorithm won't use the content as a feature to identify devices.

| Name | NFC type | Standard | Chip | Writable | ATQA | SAK |
|---|---|---|---|---|---|---|
| **tag1** | NFC-A | ISO 14443-3A | NTAG213 | Yes | 0x0044 | 0x00 |
| **tag2** | NFC-A | ISO 14443-3A | NTAG213 | Yes | 0x0044 | 0x00 |
| **tag3** | NFC-A | ISO 14443-3A | NTAG213 | Yes | 0x0044 | 0x00 |
| **tag4** | NFC-A | ISO 14443-3A | NTAG213 | Yes | 0x0044 | 0x00 |
| **tag5** | NFC-A | ISO 14443-3A | NTAG213 | Yes | 0x0044 | 0x00 |
| **tag6** | NFC-A | ISO 14443-3A | Mifare Classic 1k | Yes | 0x0004 | 0x08 |
| **tag7** | NFC-A | ISO 14443-3A | Mifare Classic 1k | Yes | 0x0004 | 0x08 |
| **tag8** | NFC-A | ISO 14443-4 | Mifare Classic 4k | No | 0x0002 | 0x38 |
| **tag9** | FeliCa | JIS 6319-4 | RC-S967 | No | - | - |

Table 2: Inventory of PICC devices

## 5.3 Dataset description

### 5.3.1 Requirements

In order for the dataset to be robust, it needs to answer some basic criteria. The following list assumes the dataset will be used to train a machine learning model to discriminate between passive tags.

- It should contain recordings of both very similar and very different devices.

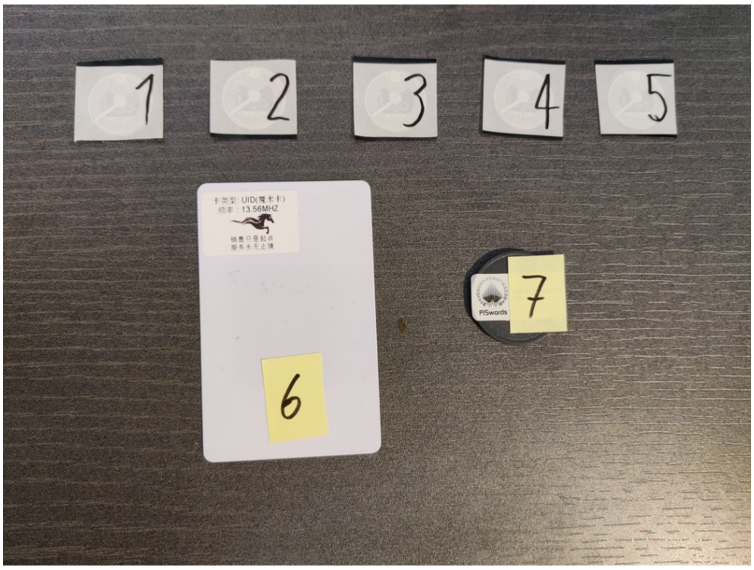

Figure 5: NFC tags 1-7

- The data transmitted should not be a discriminating factor.
- The number of devices should be high enough to analyse scalability and generalization.
- Only one reader should be used to capture the data.
- Only one SDR should be used to capture the data.
- The capture's parameters should be constant across captures.

If some of those items are not met, the model trained using this data could use the content of the tags or the capture's parameters to categorize the data. It may also be unable to generalize.

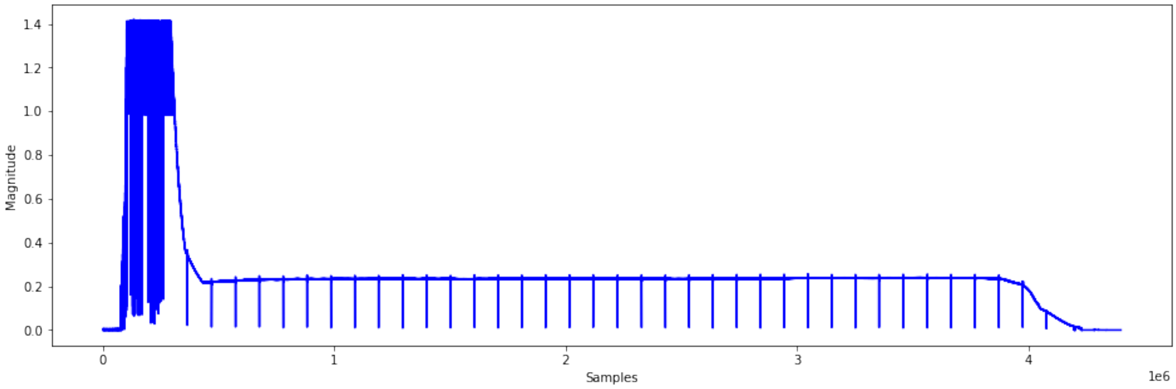

Figure 6: **Two seconds?** of NFC communication represented as magnitudes

## 5.4 Validating the dataset

## 5.5 Preprocessing

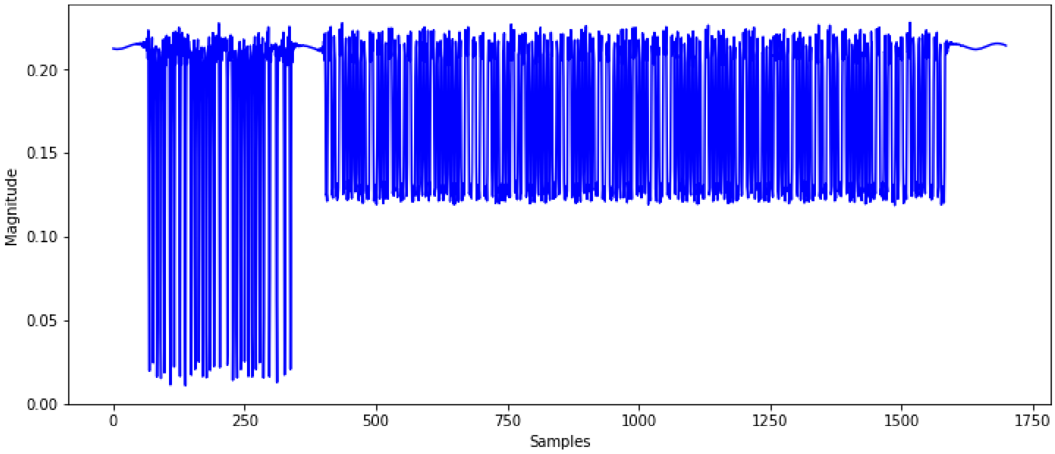

Figure 7: NFC single request/response represented as magnitudes

# 6 Model conception

## 6.1 Model architecture

## 6.2 Experiences

# 7 Conclusion

## Bibliography

1. PRITCHARD, Mike. *elttam - Intro to SDR and RF Signal Analysis* [online] [visited on 2020-04-12]. Available from: https://www.elttam.com/blog/intro-sdr-and-rf-analysis/.
2. Software-defined radio. In: *Wikipedia* [online]. 2020 [visited on 2020-06-16]. Available from: https://en.wikipedia.org/w/index.php?title=Software-defined_radio&oldid=959182555.
3. WILLIAMS, Al. *Your First GNU Radio Receiver With SDRPlay* [Hackaday] [online]. 2015 [visited on 2020-04-12]. Available from: https://hackaday.com/2015/11/12/your-first-gnu-radio-receiver-with-sdrplay/.
4. SPIESS, Andreas. *#286 How does Software Defined Radio (SDR) work under the Hood? SDR Tutorial* [online]. 2019 [visited on 2020-02-24]. Available from: https://www.youtube.com/watch?v=xQVm-YTKR9s&list=PLRlusPRPpIXTMhPZPnOITOWx_UTDW6iwi&index=12&t=273s.
5. KUISMA, Mikael. *I/Q Data for Dummies* [online] [visited on 2020-04-10]. Available from: http://whiteboard.ping.se/SDR/IQ.
6. OSSMANN, Michael. *Software Defined Radio with HackRF - Great Scott Gadgets* [online] [visited on 2020-02-24]. Available from: https://greatscottgadgets.com/sdr/.
7. Near-field communication. In: *Wikipedia* [online]. 2020 [visited on 2020-02-21]. Available from: https://en.wikipedia.org/w/index.php?title=Near-field_communication&oldid=940679317.
8. DELGADO, Oscar; KECHTBAN, Louai; LUGAN, Sebastien, et al. Passive and active wireless device secure identification. *IEEE Access* [online]. 2020, pp. 1–1 [visited on 2020-05-08]. ISSN 2169-3536. Available from: https://ieeexplore.ieee.org/document/9082628/.
9. XU, Qiang; ZHENG, Rong; SAAD, Walid, et al. Device Fingerprinting in Wireless Networks: Challenges and Opportunities. *arXiv:1501.01367 [cs]* [online]. 2015 [visited on 2020-04-24]. Available from arXiv: 1501.01367.
10. RIYAZ, Shamnaz; SANKHE, Kunal; IOANNIDIS, Stratis, et al. Deep Learning Convolutional Neural Networks for Radio Identification. *IEEE Communications Magazine* [online]. 2018, vol. 56, no. 9, pp. 146–152 [visited on 2020-04-24]. ISSN 0163-6804 1558-1896. Available from: https://ieeexplore.ieee.org/document/8466371.
11. STANKOWICZ, James; ROBINSON, Josh; CARMACK, Joseph M., et al. Complex Neural Networks for Radio Frequency Fingerprinting. In: *2019 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)* [online]. Rochester, NY, USA: IEEE, 2019, pp. 1–5 [visited on 2020-02-24]. ISBN 978-1-72814-352-1. Available from: https://ieeexplore.ieee.org/document/8923089.
12. OYEDARE, Taiwo; PARK, Jung-Min Jerry. Estimating the Required Training Dataset Size for Transmitter Classification Using Deep Learning. In: *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)* [online]. Newark, NJ, USA: IEEE, 2019, pp. 1–10 [visited on 2020-03-13]. ISBN 978-1-72812-376-9. Available from: https://ieeexplore.ieee.org/document/8935823.

13. YOUSSEF, Khalid; BOUCHARD, Louis; HAIGH, Karen, et al. Machine Learning Approach to RF Transmitter Identification. 2017. Available also from: https://www.researchgate.net/publication/320890695_Machine_Learning_Approach_to_RF_Transmitter_Identification.

14. MORIN, Cyrille; CARDOSO, Leonardo; HOYDIS, Jakob, et al. Transmitter Classification With Supervised Deep Learning. *arXiv:1905.07923 [cs, eess]* [online]. 2019 [visited on 2020-03-13]. Available from arXiv: 1905.07923.

15. SANKHE, Kunal; BELGIOVINE, Mauro; ZHOU, Fan, et al. No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments. *IEEE Transactions on Cognitive Communications and Networking* [online]. 2019, pp. 1–1 [visited on 2020-02-24]. ISSN 2332-7731, 2372-2045. ISSN 2332-7731, 2372-2045. Available from: https://ieeexplore.ieee.org/document/8882379.

16. NGUYEN, Nam Tuan; ZHENG, Guanbo; HAN, Zhu, et al. Device fingerprinting to enhance wireless security using nonparametric Bayesian method. In: *2011 Proceedings IEEE INFOCOM* [online]. Shanghai, China: IEEE, 2011, pp. 1404–1412 [visited on 2020-06-12]. ISBN 978-1-4244-9919-9. Available from: http://ieeexplore.ieee.org/document/5934926/.