Synchronization and Deep Learning: Experiences Learned from Dataset Creation

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

[1]CITI Lab Inria

[2]NVIDIA Research

March 30, 2023

Summary

Context

PhD work of Cyrille Morin

- Main topic: on Deep Learning for radio
- Specific work: transmitter identification through radio fingerprinting
- Two objectives:
  - Construct a "good" dataset with Cognitive Radio Testbed (FIT/CorteXlab)
  - Test Deep Learning Convolutional Neural Network (CNN) models to fingerprint

Disclaimer!!

- Deep Learning for Radio $\Rightarrow$ Not the holy grail!
- Choose your problems:
  - Many problems in communications have well behaved closed form solutions
    $\Rightarrow$ no need for DL
  - Some problems are too difficult to model, or models are too complex or current tools are not adaptive
    $\Rightarrow$ DL may help

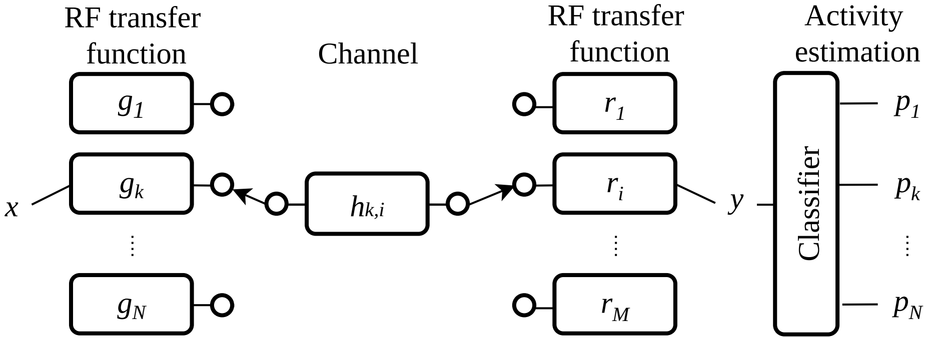
Transmitter Fingerprinting with Deep Learning

What is radio fingerprinting?

- Using the radio unique characteristics of the transmitters to tell them apart
- Radio's version of recognising the "voice" of familiar people on the phone
- Achieve radio identification in spite of the ID fields of packets!

Why do radio fingerprinting?

- IoT/small packet context
  - IoT based sensor networks: payload is of the same order of size as headers
  - High cost of header w.r.t. payload (energy and medium access)
  - Potential gains with TX identification based on radio fingerprinting
- Security/privacy context
  - Make ID spoofing harder :)
  - Enable ID-field validation :)
  - Track individual radios even if address is randomised :(


RF transfer function
Channel
RF transfer function
Activity estimation
g_1
g_k
g_N
x
$h_{k,i}$
r_1
r_i
r_M
y
Classifier
p_1
p_k
p_N

(Identifiable) Radio Characteristics

Base band

- Implementations of digital filters
- processing delays, jitter
- ...

(Identifiable) Radio Characteristics

Base band

- Implementations of digital filters
- processing delays, jitter
- ...

Intermediate Frequency

- Different sampling frequencies, digital-to-analog converters (linearity, resolution)
- Filters, amplifiers
- I/Q imbalance, DC offsets
- ...

(Identifiable) Radio Characteristics

Base band

- Implementations of digital filters
- processing delays, jitter
- ...

Intermediate Frequency

- Different sampling frequencies, digital-to-analog converters (linearity, resolution)
- Filters, amplifiers
- I/Q imbalance, DC offsets
- ...

Radio Frequency

- RF filter characteristics, RF amplifiers
- Local oscillator frequency offset, jitter, noise
- ...

(Identifiable) Radio Characteristics

From Prior art

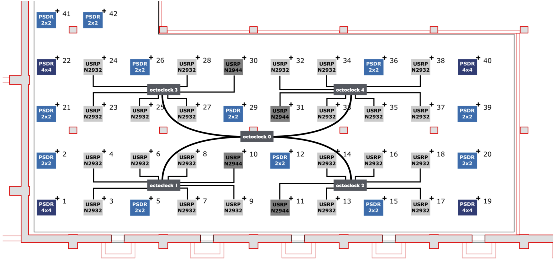
- Channel effects **[Xiao2009]** **[Xiao2009a]**
  - Channel effects dominate over RF signature
  - Identification is restricted to propagation features: not robust
- Power amplifier characteristics **[Sankhe2018]** and **[Wong2018]** **[Hanna2018]**
  - Artificially manipulate the TX signals to exploit different characteristics
  - Easy to impersonate
- Local oscillator imperfections **[Hanna2018]**

Objectives

The task at hand

- Identify 21 transmitters using an IoT like signal (USRP N2932 - *equiv. N210 with SBX*)
- Use only raw IQ samples
- No channel equalization
- Produce a correctly labelled dataset devoid of channel bias (as much as we can)
- Test learning and generalisation ability

FIT/CorteXlab

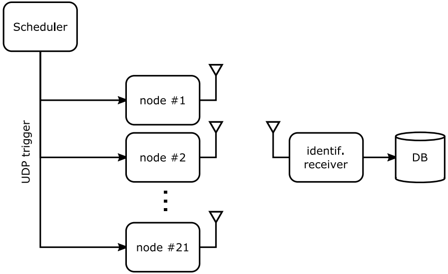


Characteristics

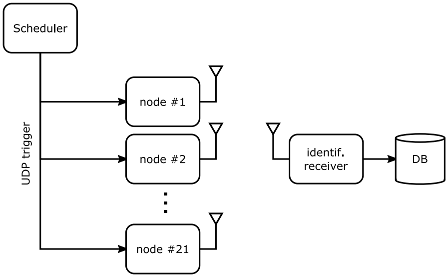
- Total of 42 radio nodes among USRPs N2932 and N2944R, PicoSDRs 2x2 and 4x4, Octoclocks for synchronization (USRPs only)
- Fully eletromagnetically isolated and semi-anechoised experimentation room (at least 60 dB of isolation)
- Robots for radio mobility
- Remote access operation, 100% automated experiment deployment

Dataset Generation

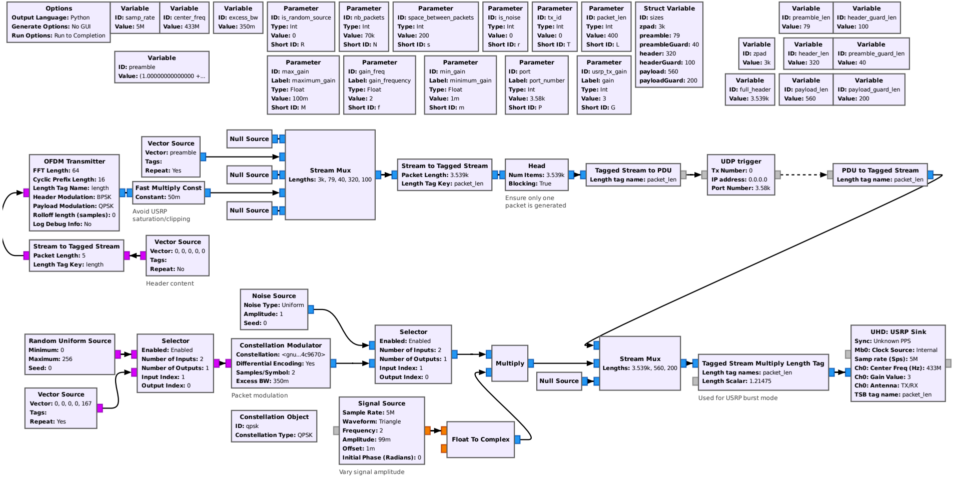
Overall structure



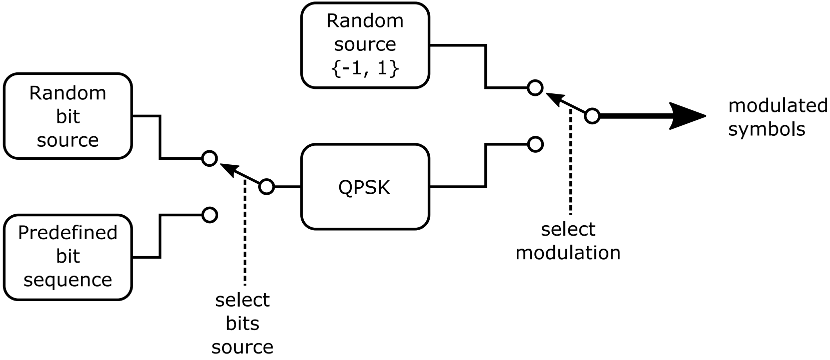
Overall structure



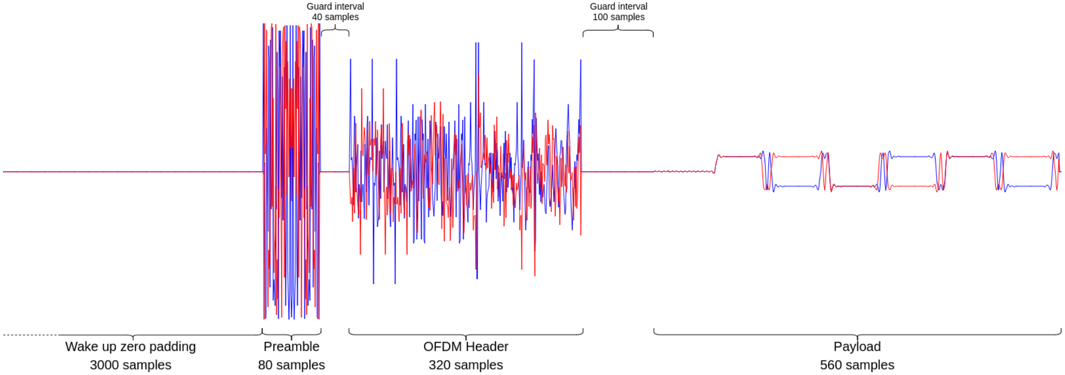
Transmission



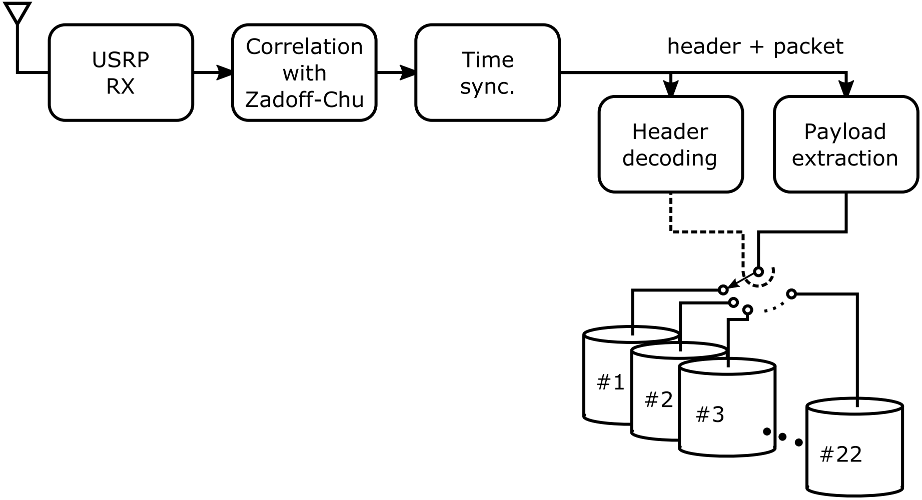
Payload selection



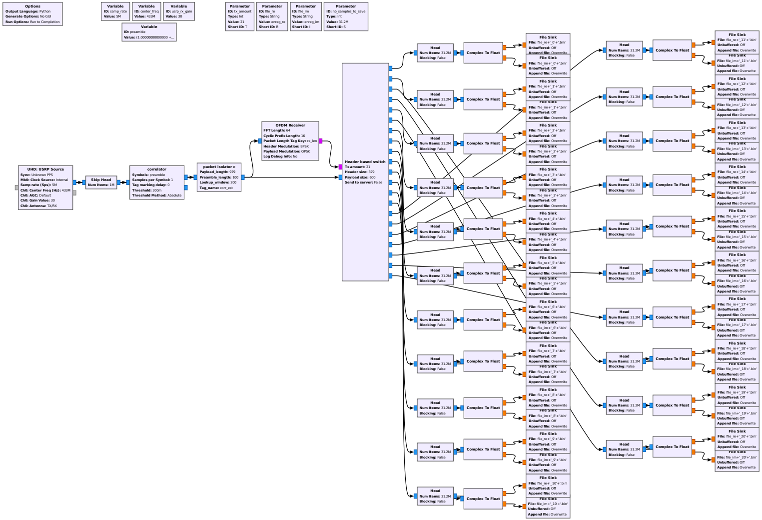
Frame


Guard interval
40 samples
Guard interval
100 samples
Wake up zero padding
3000 samples
Preamble
80 samples
OFDM Header
320 samples
Payload
560 samples

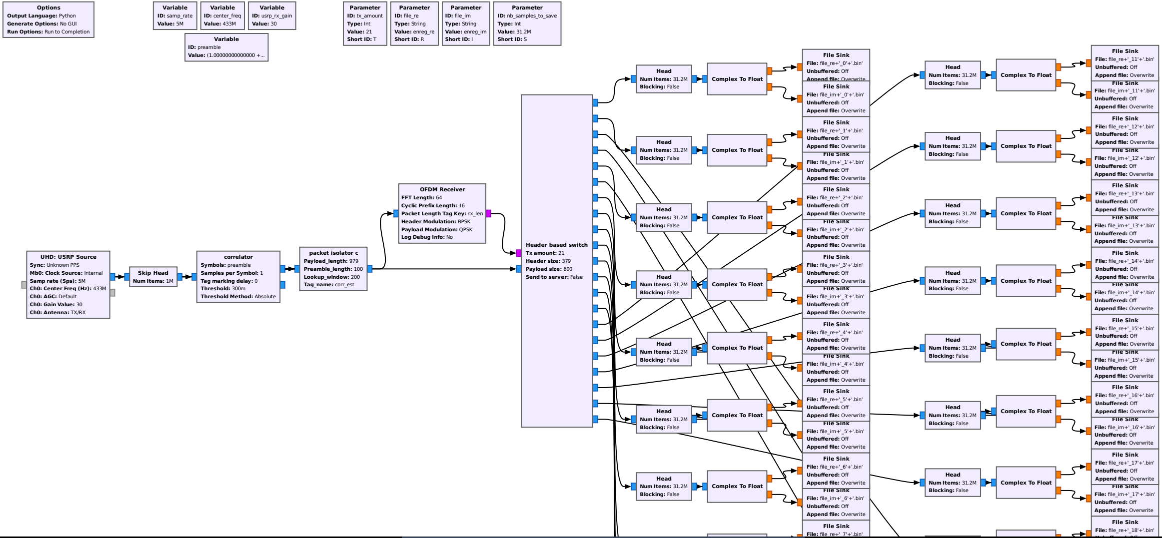
Reception and Labeling (Simplified)



Reception and Labeling



Reception and Labeling (zoom)



Learning and Classifying

Classification

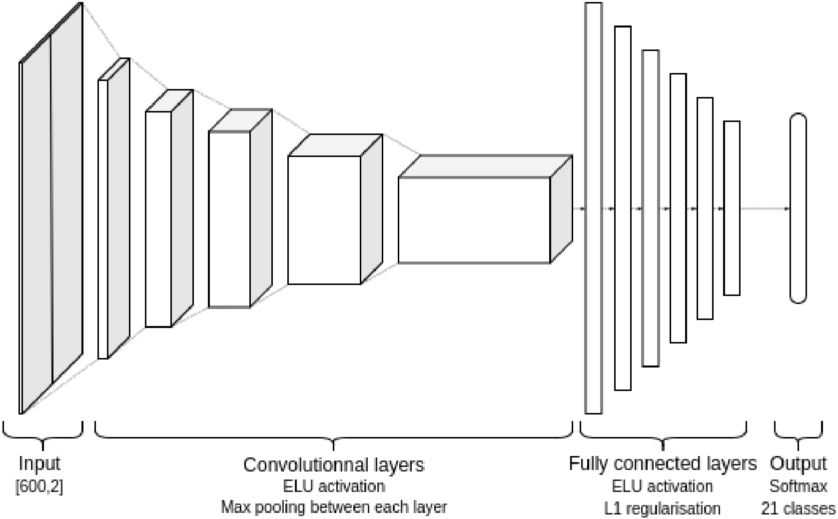
Classifier and data

- Gold standard classifiers for fingerprinting: Convolutional Neural Networks (CNN)
- Used mainly for image recognition, due to its feature extraction capabilities
- Raw packet (600 samples) is used to construct an "image"
- Complex samples $\Rightarrow$
  - Real part is encoded in first column
  - Imag part is encoded in second column
  - Rows are each sample of the packet
- $2 \times N_{\text{samples}}$ "image"

Data handling

- $50000 \times 21 = 1050000$ examples in one dataset
- Dataset examples split in 70% training, 10% validation and 20% test sets

Neural network architecture

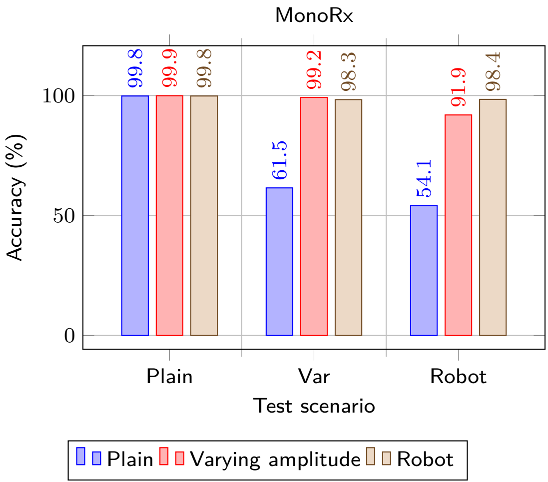


Training and testing process

Training process

- Offline - on a GPU server connected to FIT/CorteXlab
- 128 examples per batch
- More than 30 training epochs

Generalisation capabilities



Moving to multi-RX: three months of misery...

Move to MultiRX

What we have done until now...

- Up to this point we were using node 6 as the RX
- High percentage accuracy on tests over the same training dataset (good)

Move to MultiRX

What we have done until now...

- Up to this point we were using node 6 as the RX
- High percentage accuracy on tests over the same training dataset (good)

Need to add channel and receiver characteristics variations $\Rightarrow$ MultiRX

- Create datasets with other receivers
- Combine all datasets into a big mixed dataset to mix all channels
- Provide generalisation with respect to receiver RF signature and channels

Move to MultiRX

What we have done until now...

- Up to this point we were using node 6 as the RX
- High percentage accuracy on tests over the same training dataset (good)

Need to add channel and receiver characteristics variations ⇒ MultiRX

- Create datasets with other receivers
- Combine all datasets into a big mixed dataset to mix all channels
- Provide generalisation with respect to receiver RF signature and channels

Problem

- Multi-RX datasets gave very poor performance!

Further investigation

SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs $\Rightarrow$ good

Further investigation

SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs $\Rightarrow$ good
- **Cross:** Training on dataset of one RX and testing on one of another RX yields poor results $\Rightarrow$ expected

Further investigation

SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs $\Rightarrow$ good
- **Cross:** Training on dataset of one RX and testing on one of another RX yields poor results $\Rightarrow$ expected
- **After:** Training and testing on a dataset for the same RX collected at different moments:

Further investigation

SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs $\Rightarrow$ good
- **Cross:** Training on dataset of one RX and testing on one of another RX yields poor results $\Rightarrow$ expected
- **After:** Training and testing on a dataset for the same RX collected at different moments:
  - Poor results for all RXs $\Rightarrow$ weird...

Further investigation

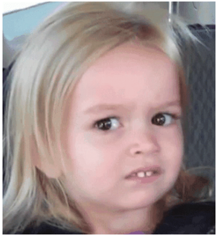
SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs ⇒ good
- **Cross:** Training on dataset of one RX and testing on one of another RX yields poor results ⇒ expected
- **After:** Training and testing on a dataset for the same RX collected at different moments:
  - Poor results for all RXs ⇒ weird...
  - Except node 6 ⇒ WTH!??!!??

Further investigation

SingleRX tests, but this time done over more RXs

- **Same:** Training and testing on the same dataset works perfectly for the selected set of RXs $\Rightarrow$ good
- **Cross:** Training on dataset of one RX and testing on one of another RX yields poor results $\Rightarrow$ expected
- **After:** Training and testing on a dataset for the same RX collected at different moments:
  - Poor results for all RXs $\Rightarrow$ weird...
  - Except node 6 $\Rightarrow$ WTH!??!!??



Debugging

Verification of RXs

- Exchange USRP devices that do not generalise ⇒ some replacements work, some others don't
- Hours and hours of analysis of hundreds of base band packets by eye ⇒ no visible differences

Debugging

Verification of RXs

- Exchange USRP devices that do not generalise $\Rightarrow$ some replacements work, some others don't
- Hours and hours of analysis of hundreds of base band packets by eye $\Rightarrow$ no visible differences

Hypothesis

Debugging

Verification of RXs

- Exchange USRP devices that do not generalise $\Rightarrow$ some replacements work, some others don't
- Hours and hours of analysis of hundreds of base band packets by eye $\Rightarrow$ no visible differences

Hypothesis

Could it be a sampling/synchronization problem?

Debugging

Verification of RXs

- Exchange USRP devices that do not generalise ⇒ some replacements work, some others don't
- Hours and hours of analysis of hundreds of base band packets by eye ⇒ no visible differences

Hypothesis

Could it be a sampling/synchronization problem?

Facts

- Before starting the MultiRX tests the octoclocks were installed
- We turned on "external reference clock" on the basis of "it can not hurt to be synchronized"

Testing the hypothesis

Disable "external reference clock" and repeat the generalisation tests $\Rightarrow$

Testing the hypothesis

Disable "external reference clock" and repeat the generalisation tests $\Rightarrow$ IT WORKS!!!

Testing the hypothesis

Disable "external reference clock" and repeat the generalisation tests ⇒ IT WORKS!!!

So what happened to node 6?

Testing the hypothesis

Disable "external reference clock" and repeat the generalisation tests $\Rightarrow$ IT WORKS!!!

So what happened to node 6?

We verified the cable between the USRP and its octoclock $\Rightarrow$ continuity issues

Testing the hypothesis

Disable "external reference clock" and repeat the generalisation tests $\Rightarrow$ IT WORKS!!!

So what happened to node 6?

We verified the cable between the USRP and its octoclock $\Rightarrow$ continuity issues

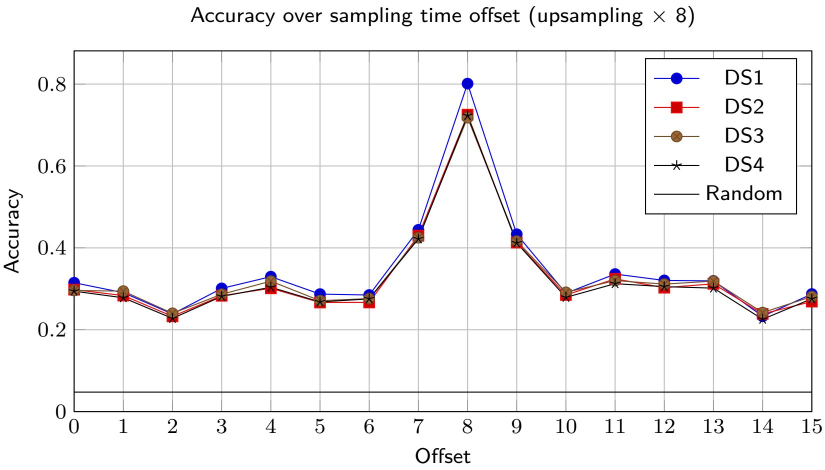
Final explanation

- The signal used to train is always sampled at the same rate (2 samples/syms) at the RX, but with:
  - A random initial sampling time for each transmission for unsynchronised nodes
  - A fixed and *constant* initial sampling time for each transmission for synchronized nodes

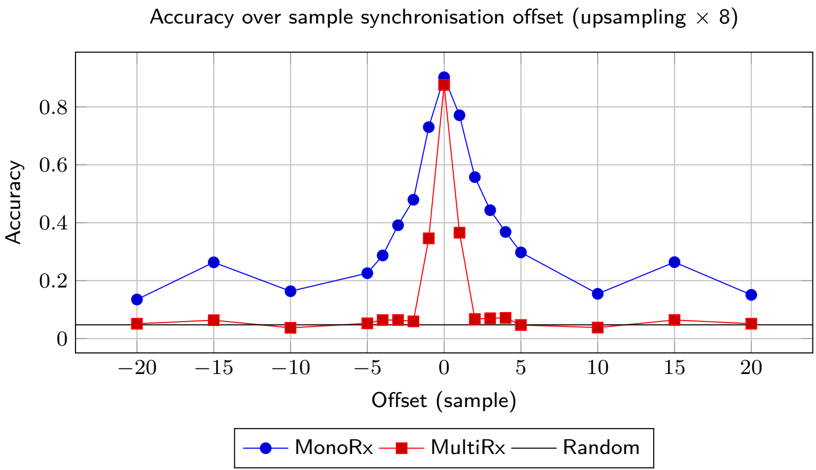
- Which essentially means:
  - A random initial sampling time for each transmission
    $\Rightarrow$ CNN learns to "ignore" sampling differences
  - A fixed and constant initial sampling time for each transmission
    $\Rightarrow$ CNN learns one sampling and can not generalise to another sampling time

Let's dig deeper...

Sample time synchronisation



Sample time synchronisation



Synced radio	Tx	Both	Rx	None
Tx	81.2%	34.5%	19.2%	38.3%
Both	54.4%	45.6%	38.2%	38.2%
Rx	21.1%	28.5%	80.2%	22.3%
None	41.3%	43.8%	34.6%	81.1%

Table: Accuracy of networks trained on one synchronisation possibility and tested on the others

Conclusion

Takeaways

- Creating datasets for training DL for radio (or anything else) can be hard
- Seemingly innocuous parameters and settings can completely derail your DL system
- Datasets with raw I-Q samples: always check your clock source on UHD
  - Using PPS to synchronise time is probably OK
  - Using the 10 MHz REF signal to synchronise clock references should be carefully studied in your specific case
- Keep a logbook of important changes to your platforms if you're changing them while experimenting

If you want to explore further

- Full paper is online at: https://hal.inria.fr/hal-03117090/
- Datasets online at: https://wiki.cortexlab.fr/doku.php?id=tx-id
- FIT/CorteXlab wiki page: https://wiki.cortexlab.fr
- Next publication on this is coming soon...