# MPA-MLF - Final project - Czech/Slovak/Erasmus

## Classification 5G base stations

**Date:** 7.4.2025, UPDATED

## 1 Introduction

We are all using our cell phones on a daily basis. These are our connections with family and friends, help us to search for information, etc. The cell phone needs always be connected to the network, through a so-called base station (as telecommunication engineers, we call them eNodeB's or gNodeB's).

But, there can be attackers taking advantage of security vulnerabilities in mobile networks. With the use of specialized hardware and software, they can steal some informations, send malicious messages, or track users. One of the known tools is a so-called False Base Station (FBS), sometimes called Rogue Base Station (RBS), or International Mobile Subscriber Identifier (IMSI) Catcher. This is a device pretending as the legitimate base station and trying the mobile terminals to connect.

The false base station thus needs to behave as the legitimate one, i.e., it needs to broadcast the same information to the radio channel.

## 2 Task description

The frame structure of 4G/LTE signal sent from the base station is shown on left part of Fig. 1. A crucial part of transmitted information are so-called synchronization sequences. There are two of them - a primary synchronization sequence (PSS) and a secondary synchronization sequence (SSS). During our measurement campaign, we captured huge number of such sequences and used them to calculate the channel frequency responses. From these channel frequency responses, we created samples that you will process. An example of one sample visualized in 2D is shown in the right part of Fig 1. Note it consists of 72 rows (corresponding to 72 subcarriers allocated to PSS/SSS) and 48 columns (corresponding to 48 repetitions of channel frequency responses).

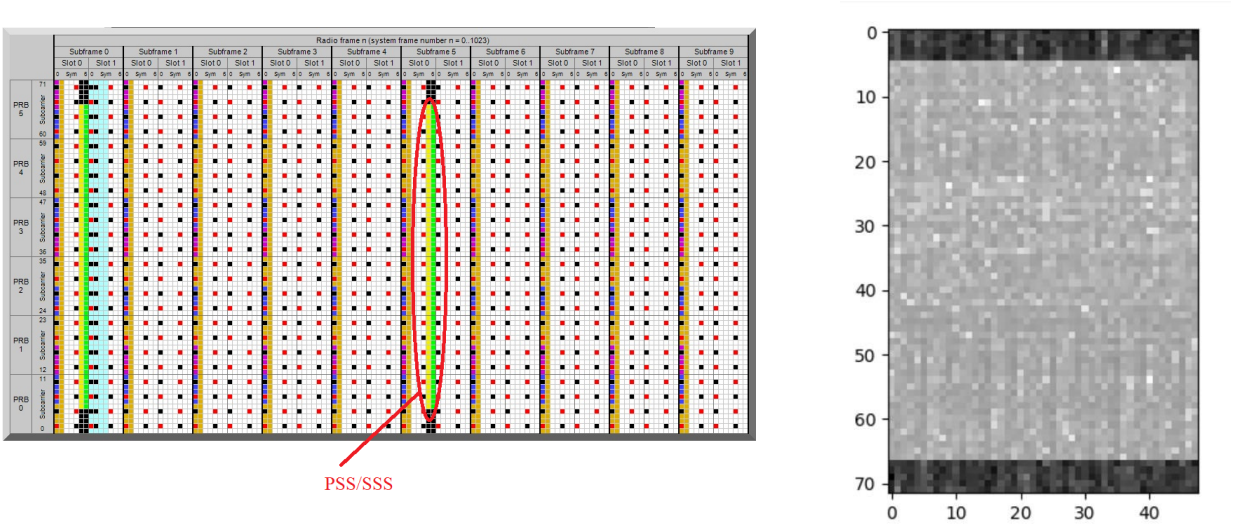


Figure 1: LTE/4G frame structure (left), example of training sample in 2D (right)

The goal of your project is to classify whether there is only a legitimate base station (gNodeB) of T-mobile operator (class 0) transmitting from a neighboring building or whether an attacker brought his transmitter into the building and is trying to steal the information from users (class 1 and class 2). The false/attacker's base station could be located at one of two positions - class 1 corresponds to the first one, and class 2 corresponds to the second one.

All dataset files can be found here: Link to the Datasets . The link contains a .zip archive with the following files:

1. **Train** - folder with training files in *.npy* format.
2. **Test** - folder with test files in *.npy* format.
3. **label_train.csv** - *.csv* file with labels for your training data.
4. **test_format.csv** - example of the data format that is for submission.

Please also note that you are not forced to process the samples as 2D images, you are also free to convert the data to 1D vectors etc.

## 3 Steps

Your task is to make a classification module that will work correctly to solve the task described above. We do not prescribe the type of ML model. You can use any architecture we discussed during the semester (or any other that you are familiar with). Try to achieve the highest test accuracy possible. Tune your model's performance using any techniques we discussed, like Data augmentation, Regularization, Batch normalization, etc. On top of that, use a hyperparameter tuning algorithm to find the best hyperparameters and the model structure. Describe your approach properly in the report.

## 4 Submission and grading

You are required to work in pairs. Please make your pairs on your own and write down the teams in the following document: Link to the document . You are required to create your teams until **6.4.2025**.
Your solutions will be submitted in the following different ways:

- **Report, e-learning**. You will upload your report into the e-learning. Upload your report in .pdf format including the final results achieved on Kaggle.
- **Model predictions, Kaggle** You are required to test your results in the Kaggle competition, link: Link to the competition .
- **Code, GitHub, e-learning**. Please **Link with GitHub the folder with the final code to your report**.

The deadline for submitting the report is **4.5.2025**. The deadline for submitting the results to Kaggle is **4.5.2025**

## 5 General comments

- The report should have all the necessary formalities that a report of a similar type has (introduction, problem description, the main body of your work, conclusions, all figures and tables should have labels and numbering and should be referenced in the text, etc.). You should be familiar with this (at least) from your Bachelor's thesis.
- Do not include the screenshots of your code in the report. If you wish to describe some of your algorithms, use pseudo-code/flow charts. Do not describe well-known algorithms.
- Do not present your work in the report as plain text. Use graphs, figures, and tables to show and present the relevant information.

- You are required to do all of your coding in Python. You don't have to do all coding from scratch, but you are all allowed to use Python's libraries and frameworks (for instance, *Keras*, *PyTorch*, *Scikit-learn.*.).
- The use of Google Collab is strongly recommended but not required.
- Ensure that you properly describe your work in your report; we are more interested in the process of your work than in a correct result.
- Reports that fail to follow the instructions given in this section will be rejected.