CRYING REMOVING

# Learning features on crying sections

Before removing the ‘Crying segments’ (CSs), the first step involves differentiating them from the ‘Non-Crying segments’ (NCSs). This is done by supervised learning, using the 37 labeled signals from the current database. The learning phase will enable to determine one or more characteristics differentiating the CSss and NCS, so that an automatically CS detection can be computed. It will be useful when other recordings will be done, enlarging the database.

The labeling of the raw signals, the determination of the characteristics to be used and the K-fold Cross Validation are the three necessary phases before the CSs can be removed.

## Labelling the crying sections thanks to annotators

The first step was to label the signals with CSs and NCSs. This was done by 3 annotators on the entire signal basis. Independently, Lindsay, Arrabella and I listened to the different samples and annotated them on Audacity. An example of annotated signal is in Figure 1.

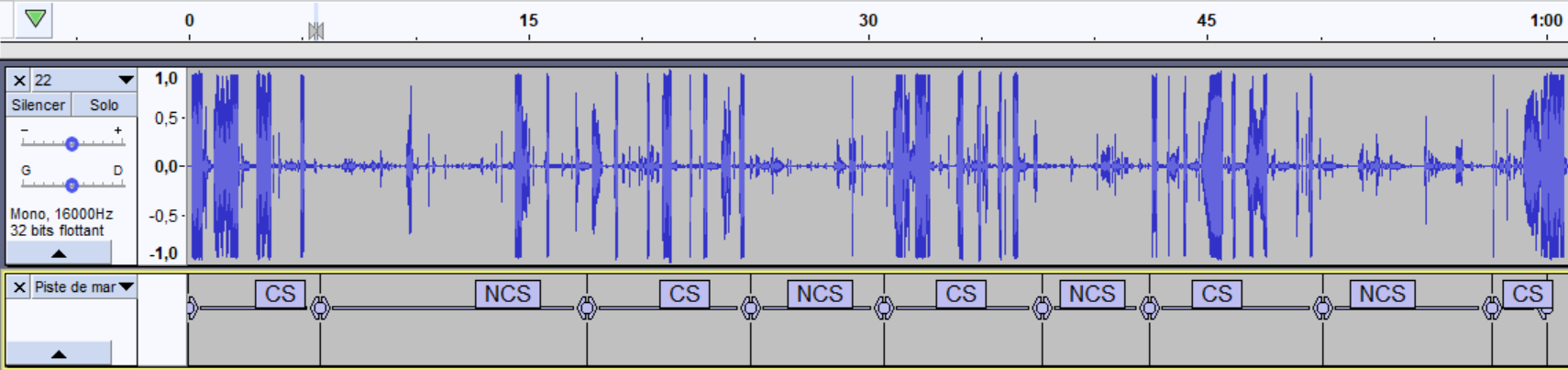


Figure 1: Julie's annotations of signal 22 on Audacity

The labels were extracted as text files. Each text file follows a strict name structure: ObersatorID\_SampleID.txt. Every line corresponds to a CS or NCS, with the beginning and end time of the section (cf figure 2).

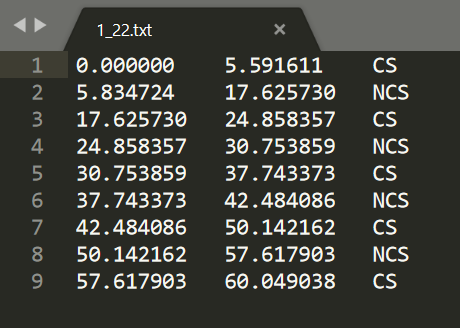


Figure 2: Text file generated by Audacity after Julie's annotations on signal 22

These files are read one by one in MATLAB, with the aim of creating a vector of 0 and 1, respectively corresponding to the NCSs and CSs labels. A moving average with a 1-second window and a 25% overlap was applied. The window was chosen based on the duration of CS, which is often 1 second or more.

The level of agreement between the annotators was then measured on each signal using Fleiss’ KAPPA. It is a statistical measure which assesses the [reliability of agreement](https://en.wikipedia.org/wiki/Inter-rater_reliability) between a fixed number of raters when assigning [categorical ratings](https://en.wikipedia.org/wiki/Categorical_rating) to a number of items[[1]](#endnote-1). In the project, three raters (Arabella, Lindsay and I), two categorical ratings (CS/NCS) and sixty items (60 sections of 1 second) were used to find a Fleiss’ KAPPA coefficient for each signal. A KAPPA coefficient equal to 0 means no agreement, while 1 means perfect agreement. The detailed calculation of this coefficient can be seen in annex … . It was implemented on Matlab using a function in the Matlab File Exchange [[2]](#endnote-2) . GIVE THE VALUES OF KAPPA COEFFICIENTS

Finally, each CS with a 2/3 or 3/3 agreement have been retained. The figure 3 illustrates the final annotated labels of signal 22.

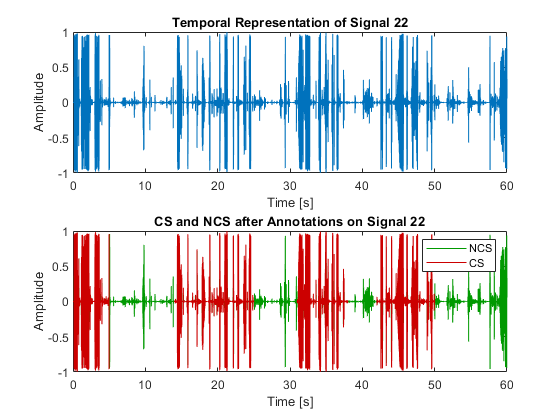
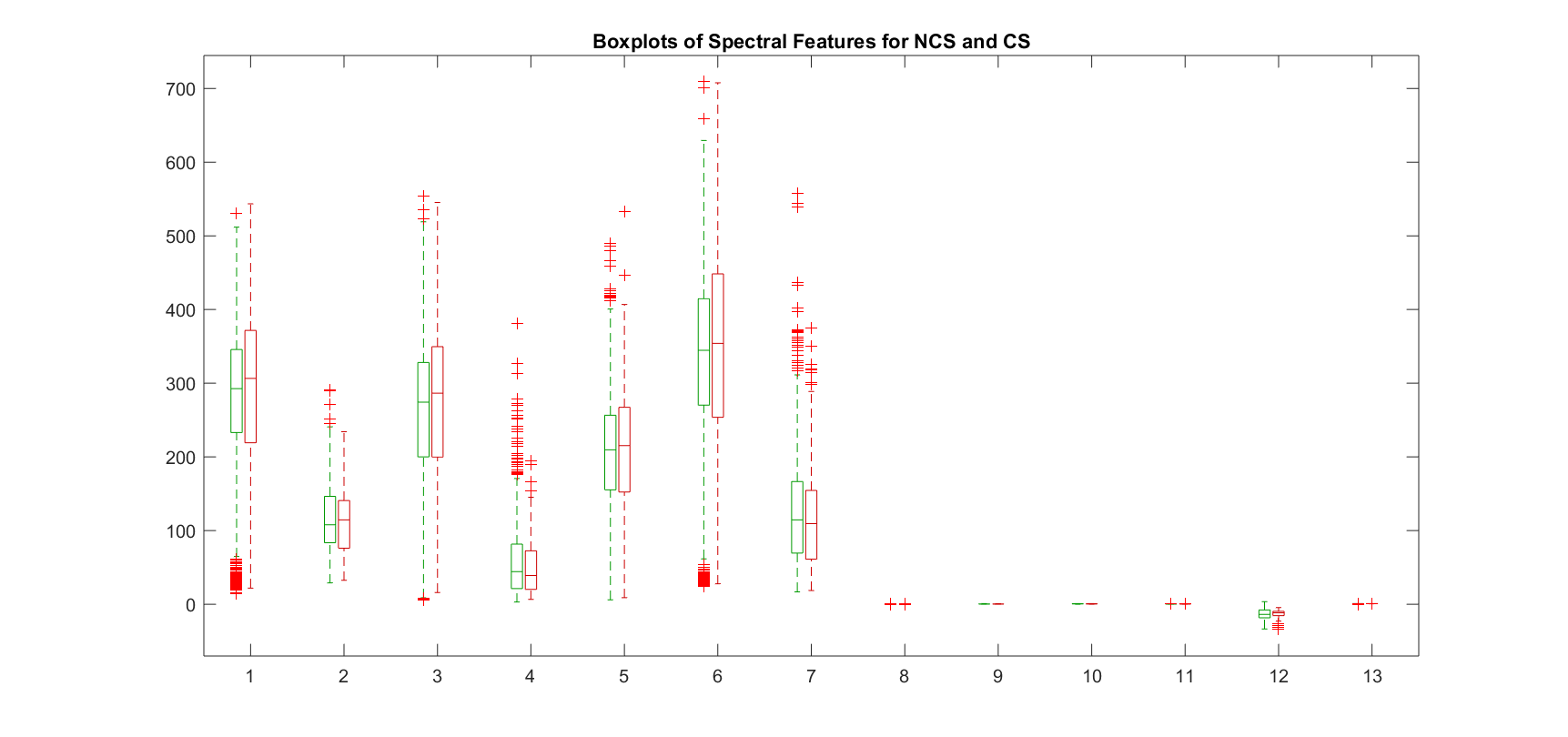


Figure 3: Annotated CS and NCS of Signal 22

## Analysis of differences between CS and NCS

Once the theoretical labeling of the signals has been done, it is necessary to learn how to detect CSs. In order to know which characteristics will be most appropriate for the detection of CSs, a first study has been done. It is based on box plots illustrating the differences between NCSs and CSs on common signal processing characteristics. Power spectrum features as well as MFCC coefficients and LPC coefficients are analyzed (figures …)



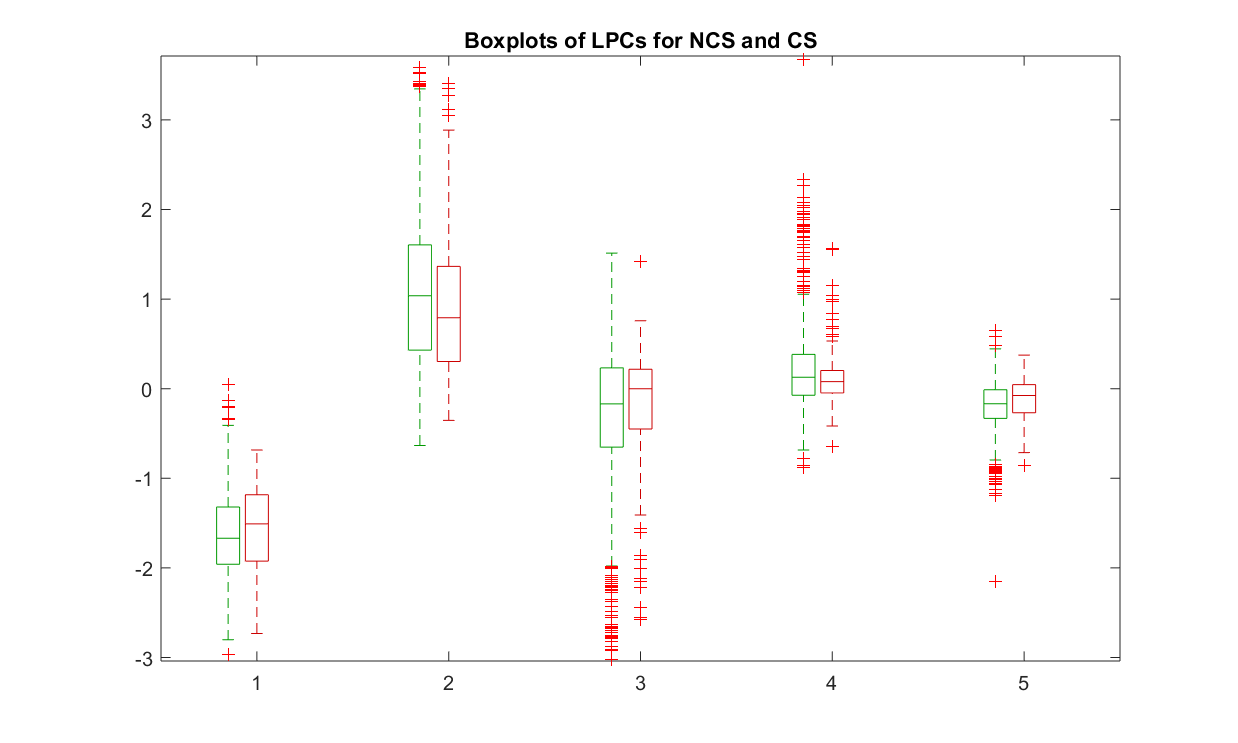


FIGURE MFCC MISSING + PUT LEGEND

Spectrograms of some signal parts where crying is present are also generated to compare the frequency differences between NCS and CS. To do so, a small window length was used to have a good time resolution (window=1s, overlap=25%). The spectrogram of the first 15s of signal 22 is in the figure 4 below.

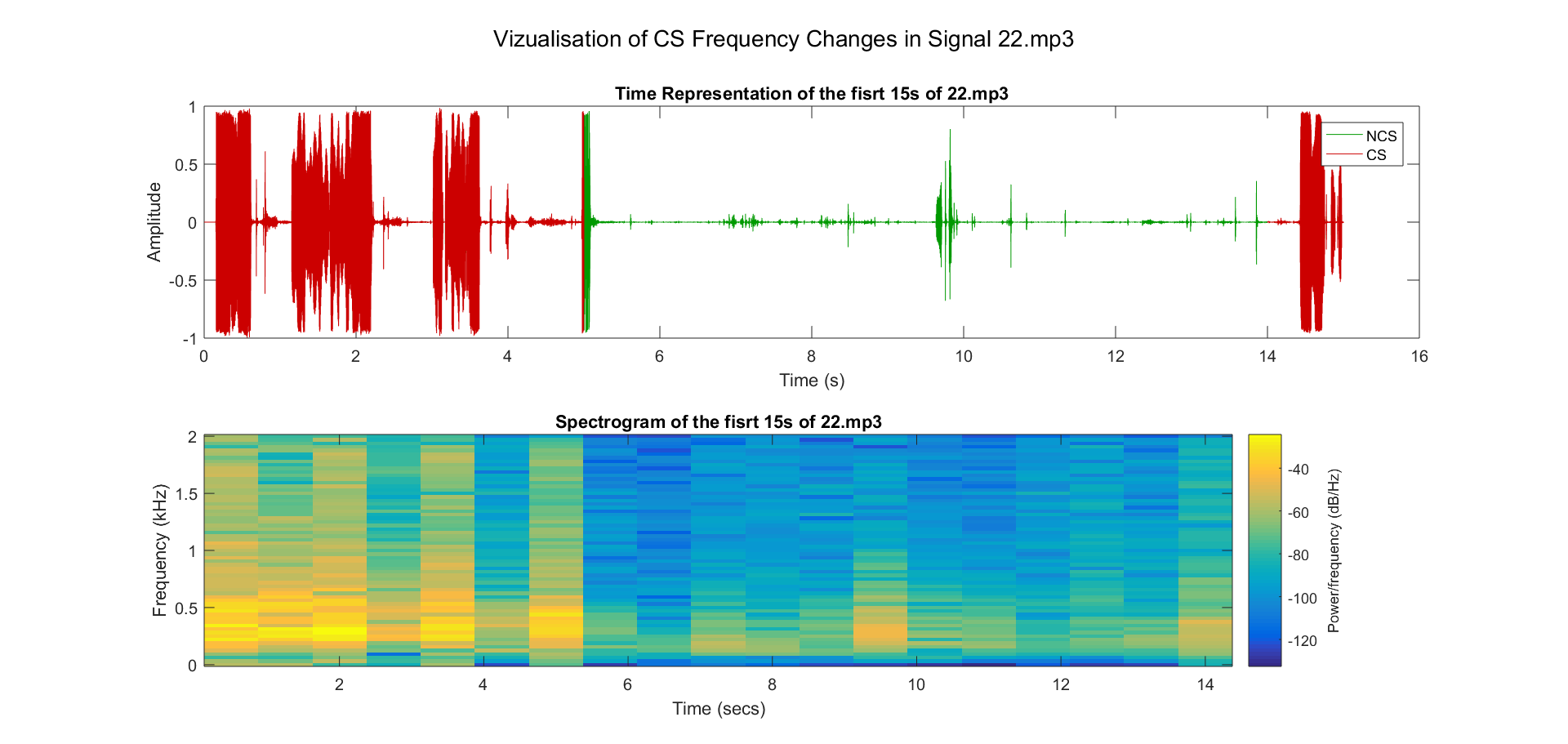


Figure 4: Vizualisation of CS Frequency Changes in the first 15s of Signal 22

Finally, it was decided that the Power Ratio could be enough to differentiate the CS and NCS. When labelling the signals with a window of 1s, a huge difference can be seen in the power spectrum average between 600Hz to 1000Hz (cf figure 5).

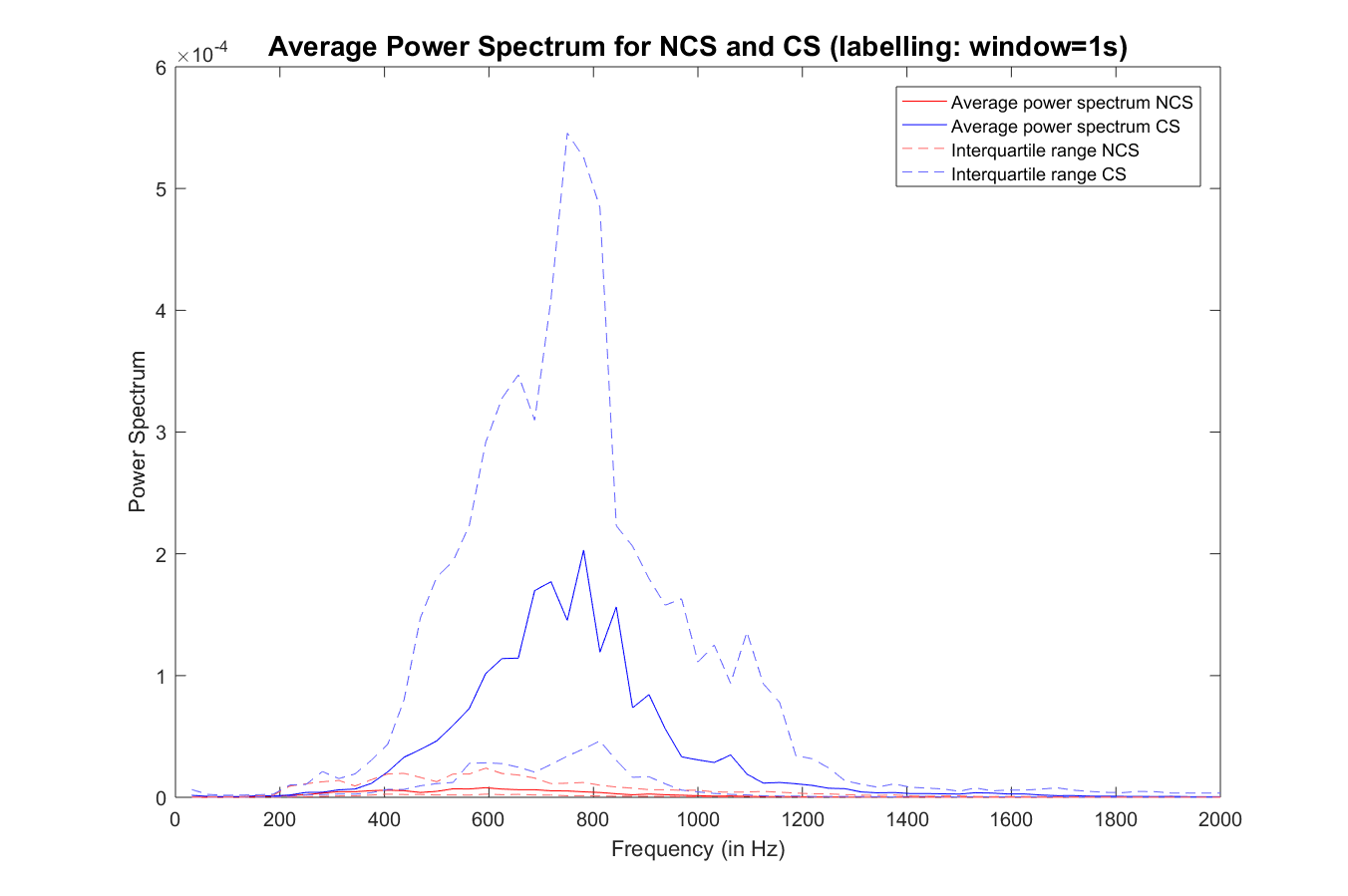


Figure 5: Power Spectrum Average of NCS and CS

A threshold in the power spectrum average must be determined to differentiate a CS from an NCS. ADD CALCULATION OF THE THRESHOLDS

They are then tested thanks to K-fold Cross Validation.

## Cross validation

Cross-validation is a statistical method often used to estimate the skill of machine learning models on a limited data sample. In this particular case, it will be helpful to evaluate the better threshold.

A training dataset (required to determine the threshold) and a validation dataset (which allows threshold testing on new samples) are required. Different cross-validation methods exist to differentiate those two datasets. Exhaustive cross-validation methods learn and test on all possible ways to divide the original sample into a training and a validation set, whereas the non-exhaustive ones do not compute all ways of splitting the original sample.

The k-fold cross-validation method was chosen. It is non-exhaustive but remains a method that does not introduce much bias and allows a quick calculation time. The procedure has a single parameter called k, that refers to the number of groups that a given data sample is to be split into. Each CS and NCS is assigned to an individual group and stays in that group for the duration of the procedure. It will be used for the validation 1 time and k-1 times to train the model.

The general method is as follows[[3]](#endnote-3):

1. Shuffle the dataset randomly.

2. Split the dataset into k groups

3. For each unique group:

1. Take the group as a test data set

2. Take the remaining groups as a training data set

3. Fit a model on the training set and evaluate it on the test set

4. Retain the evaluation score and discard the model

4. Summarize the skill of the model using the sample of model evaluation scores

The value for k is chosen such that each train and test groups are large enough to be statistically representative of the entire dataset. The k-value 5 or 10 are often used as it was shown empirically that they don’t suffer neither from excessively high bias nor from very high variance.[[4]](#endnote-4) In this case, the dataset is composed of 37 signals with a label window of 1s, which allows 2183 observations. 10 was then chosen as the k value.

1. <https://en.wikipedia.org/wiki/Fleiss%27_kappa> [↑](#endnote-ref-1)
2. <https://github.com/dgolden1/matlab_fleiss_kappa/blob/master/fleiss_kappa.m> [↑](#endnote-ref-2)
3. <https://machinelearningmastery.com/k-fold-cross-validation> [↑](#endnote-ref-3)
4. An Introduction to Statistical Learning, 2013 [↑](#endnote-ref-4)