PROJECT 4

NEUROROBOTICS PROJECT REPORT

SOFIIA TATOSH, CECILIA ROSSI
NOEMI GONZATO, FRANCESCA VIRGOLINI
Department of Engineering
University of Padova
February 2023

1 INTRODUCTION

The project's aim was to analyze the brain activity of 5 healthy subjects, involved in an experiment in which they were asked to control the avatar of a wheelchair during a virtual race. Their EEG activity was being recorded while they were sending commands via a joystick. The data were recorded using a 16-channel amplifier, with a sample frequency of 512 Hz. The electrodes were positioned following the 10-20 international layout.

The race itinerary consisted in a pathway composed by the repetition of four different sectors:

1 - The "turn right" and "turn left" sectors, in which the subject was supposed to send a discrete command to the game, in order to make the avatar turn respectively to the right or to the left;

2 - The "light sector" and the "straight" sector, in which the subject was not supposed to send any command.

Moreover, the control had a 20% probability to invert the user's command, and thus to execute the wrong command in the game (like turning right instead of left: we saw that usually those commands were usually followed by a correction made by the user).

Furthermore, it had the same probability of sending a random command about 4 seconds after entering in a light or straight area. These two errors, which will be addressed respectively as "Curve Error" and "Light/Straight Error", are actually not due to a mistake of the user but to the game implementation. They were expected to generate the outbreak of an Error Potential in the EEG pattern corresponding to the temporal window just after the event, more noticeable in some EEG channels than others.

Using the data and the literature provided, we approached the problem of processing the data and training a classifier able to correctly identify the presence of Error Potentials in the 1 second time window following the beginning of the feedback, that occurs when the avatar moves as a consequence of a command being sent.

2 METHODS

2.1 DATA PROCESSING

In this section the materials and methods employed for the analysis of the EEG signals are going to be described. The subjects taken into consideration were five healthy subjects controlling the avatar of a wheelchair during a virtual race via joystick. First of all, the given data were reorganized in a way that was convenient for the subsequent analysis. To do so, the data were divided by subject into 5 directories, each one called 'P'+n, where n represents the label number of the subject which goes from 1 to 5. For what concerns the processing part, the steps implemented for each subject were the following.

DATA CONCATENATION. First of all, a function called ***conc_project*** was implemented, with the aim of taking all the EEG data for each selected subject and creating a single signal of dimensions [samples x channels] which includes all the runs.

LABELLING. The obtained signal was given as input to a new function, ***labelling_project***, which creates the labels for each type of event which characterized the signal. More specifically, among the outputs there were the label vectors useful for selecting the samples of interest for the Error Potential's appearance.

CAR FILTERING. Afterwards, a Common Reference Filter (CAR) was applied as a spatial filter. CAR is a spatial filter often cited in literature for highlighting evoked potentials [1]. It subtracts from each electrode the average potential over the 16 channels at each time step, as described in the following equation:

$$e_i^{CAR} = c_i - \frac{1}{N}\sum_{j}^{N} e_j \tag{1}$$

where e_i represents the raw potential of electrode i and N the number of electrodes. The aim of this rereferencing procedure is to suppress the average brain activity, so as to keep the information coming from local sources below each electrode.

BUTTERWORTH FILTERING. Since the Error Potentials are known to be rather slow cortical potentials, a bandpass filter was applied in the frequency band [1-10 Hz]. More specifically, a 4th order Butterworth filter was used [1, 2, 3, 5].

DOWNSAMPLING. To reduce the dimensionality of the data, the filtered signal was then downsampled from 512 Hz to 64 Hz [2, 3].

CHANNELS AND TRIAL SELECTION. Among the 16 channels, Fz, FCz, Cz and CPz were selected. The choice of these channels follows the fact that error potentials are characterized by a fronto-central distribution along the midline [1]. Furthermore, the time period in which an Error Potential could appear was selected. More specifically, it was assumed that an ErrP could appear in a time window that goes from 200 ms to 1000 ms after every command sent by the subjects, in the case that the command revealed itself to be wrong.

TRAIN AND TEST SET SEPARATION. To implement the leave-one-out validation, for each run (and every subject) the train and test sets were selected and two different matrices, FeatureTrain and FeatureTest, were created.

CANONICAL CORRELATION ANALYSIS. As an alternative to the CAR filter, a spatial filter based on Canonical Correlation Analysis (CCA) was implemented. CCA is a multivariate statistical method that given two datasets aims at finding linear transformations that maximise the pair-wise correlation across the transformed datasets. The CCA spatial filter method transforms the averaged ErrPs to a subspace containing different ERP components, leading to the increase of signal to noise ratio (SNR) [3, 4]. The CCA was applied on the trainset, within the range of [0.2 1]s with respect to the onset. The spatial filtering matrix obtained with the analysis on the trainset was then used to filter the testset data.

GRAND AVERAGES. The grand averages of the filtered signals, belonging to the trials with and without the ErrP, were then computed and visualized, for each subject. The visualization for the population follows at the end of the code.

SAVING RESULTS. All the relevant vectors and variables were saved in a directory named 'Results'.

2.2 CLASSIFICATION

FEATURE EXTRACTION. As the data has been processed, we are ready to calculate the statistical features, that would be meaningful and significant for the classification part. Since the data processing was all done in time domain, and we are dealing with the Error Potential, for which it is important to retrieve the statistical description of the signal in time domain, we identified six features, that are essential and standard in signal description. These were mean, standard deviation, positive and negative peaks, as well as their latency. Overall, as a result, we obtain 24 features per trial - 4 channels x 6 features.

Mean and standard deviation are standard statistics of time series data. To retrieve positive and negative peaks, we considered the maximum and minimum peaks of all peaks extracted. The features were calculated during the classification part, where we had a train/test split procedure.

FEATURE SELECTION. As mentioned above, the amount of features per trial is 24, so in order to optimize classification, we decided to perform feature selection and in that way lessen the amount of features we use. In order to do that, we applied Fisher Score metrics:

$$FS(k) = \frac{|\mu_{C_1}(k) - \mu_{C_2}(k)|}{\sqrt{\sigma^2_{C_2}(k) + \sigma^2_{C_1}(k)}} \tag{2}$$

Briefly, Fisher Score measures the discrimination power of the feature, or simply an ability of the feature k to differentiate well between classes C_1 and C_2. The higher the Fisher Score, the better.

In our classification process, we decided to set a threshold based on which we can reject or accept feature k. So, if $FS(k) >= \epsilon$, we assume the feature is a significant one, but if $FS(k) < \epsilon$, we consider the feature k not essential. In our case, we set $\epsilon = 0.4$. This value was precalculated by trying out the range of ϵ such as $\epsilon_i \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The best performing one was $\epsilon_i = 0.4$.

For CCA-filtered data, on average, we reduced the number of features by almost 75%, so most of the classification iterations have a mean of $\lceil 6.54 \rceil = 7$ features. The maximum feature amount is 11, while the standard deviation is $\sigma^2 = 1.875$. For CAR-filtered data, the situation is the following: average number of features is $\lceil 9.38 \rceil = 10$ with $\sigma^2 = 3.7$, and maximum number of features throughout all iterations is 15.

CALIBRATION PHASE. In order to implement a classifier that could discriminate between Error and Correct

Trials, we used two particular models already implemented through Matlab functions:

- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)

LDA was widely used in the assigned literature [2, 3, 5], and according to the results described in the papers, the halfspaces defined by LDA represented the best complexity-performance trade-off in the case of binary classification of Error Related Potentials. Furthermore, we decided to compare the results of LDA with the results obtained by the training of a QDA, wondering if the addition of a degree of freedom in the parameters of the classifier could improve the overall performance. Also, we resolve not to distinguish the two types of errors in classification - Curve error and Straight/light error, because we did not notice significant differences in the Grand Averages curves corresponding to the types of error.

Since the classification of the Error Potentials should have been subject specific, we implemented a different classifier for every single subject and, according to the instructions given by the assignment we adopted a leave-one-out strategy with respect to the runs, in the splitting of the data between train and test sets. In fact, we divided the data corresponding to a certain subject into 10 folds, one for every run, which could contain different number of events/trials. Then, we iterated over each of these folds, considering, at every iteration, the particular fold as a test set, on which evaluate the performance of the classifier, and all the other 9 folds as training set.

EVALUATION PHASE. In order to evaluate the performance of the algorithm, the labels predicted by the classifier were compared to the ground-truth labels, previously extracted during the processing.

The labels associated to the two classes were:

- 1 for Error Trials (more specifically, -1 for straight/light error, and 1 for curve error. In this way, in both cases the absolute value of the label would have been 1);
- 0 for Correct Trials.

For both train and test set, some indexes were computed, such as:

- True Positive (TP): the number of correct events/trials, properly classified as so by the model (so for which the predicted labels are 0, equal to the proper ones);
- True Negative (TN): the number of error events/trials, properly classified as so by the model (so for which the predicted labels are 1, equal to the proper ones);
- False Positive (FP): the number of error events/trials, classified as correct trials (so for which the predicted labels are 0, but the proper labels would have been 1);
- False Negative (FN): the number of correct trials, classified as error trials (so for which the predicted labels are 1, but the proper labels would have been 0);
- [Train and Test] Error: the number of wrongly classified trials, normalized for the total number of trials classified.
- [Train and Test] Accuracy: a parameter which could be considered as an approximation of the accuracy of the classifier, computed as the number of correctly classified trials, divided by the overall number of trials classified:

$$\frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{3}$$

- [Train and Test] True Positive Rate (TPR): the number of true positive trials, with respect to the sum of all the proper correct trials, so both the correct trials classified as so, and the true correct trials classified as error trials:

$$\frac{TP}{(TP+FN)} \tag{4}$$

- [Train and Test] False Positive Rate (FPR): the number of false positive trials (error trials classified as correct ones), with respect to the sum of both error trials classified as correct ones and error trials correctly classified:

$$\frac{FP}{(FP+TN)} \tag{5}$$

- [Train and Test] Area Under the Curve (AUC): the integral of the ROC curve, so the underlying area of the curve obtained plotting as coordinates the FPR on the x axis and the TPR on the y axis. It was computed using the Matlab built-in function perfcurve. Moreover, the ROC curve was plotted for every iteration, using the values obtained by the above-mentioned function.

Finally, after the end of the 10th iteration, for every subject the Average Performance was evaluated, computing the mean on the accuracies, and the mean on the AUCs and the FPRs. All the results were saved into a 1x5 cell called ***subj_score_LDA*** or ***subj_score_QDA***, according to the particular model used to implement and train the classifier.

Every element of the cell is associated to the specific results attached to the corresponding subject, consisting into a 1x11 cell. In each of those 1x11 cell associated to each subject, the first 10 elements provided are structures, in which the arguments are the evaluation indexes quoted before, referred to the performance of the classifier in the corresponding iteration; the 11th element is a struct containing the averages indexes, in order to deliver an overall estimation of the goodness of the classifier on the subject.

3 RESULTS AND DISCUSSIONS

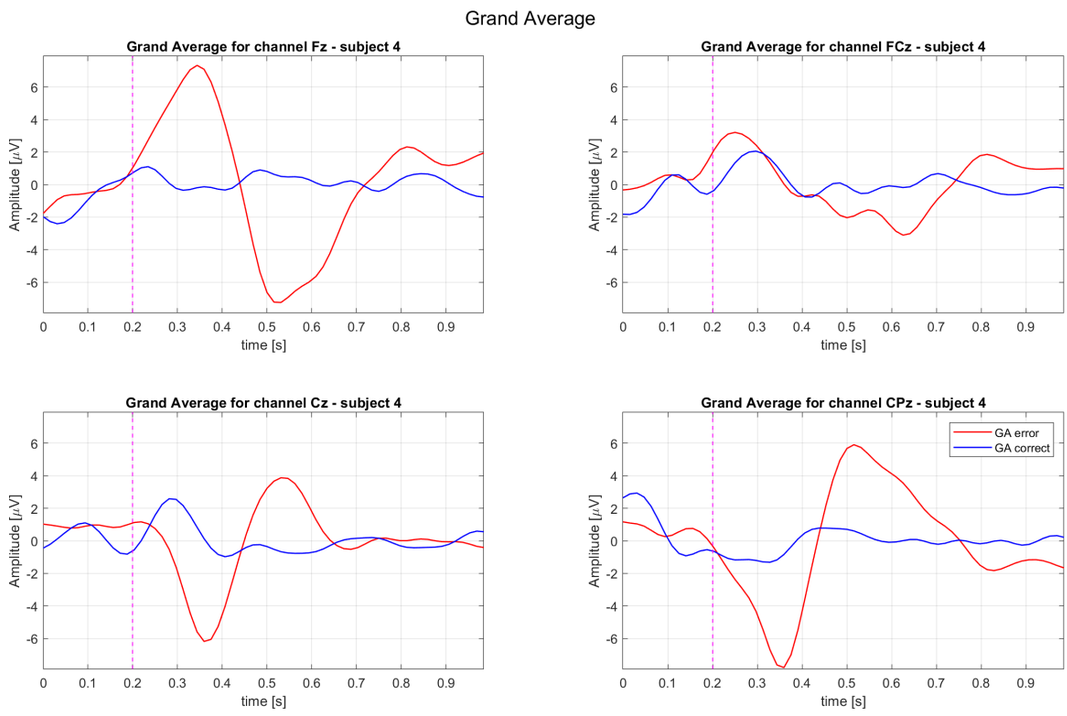

Figure 1: Grand Averages for subject 4: CAR was applied as spatial filter. The red line represents the average on the error trials, while the blue one the average signal for the correct trials. The dotted line represents the time range onset from which the signal was spatially filtered.

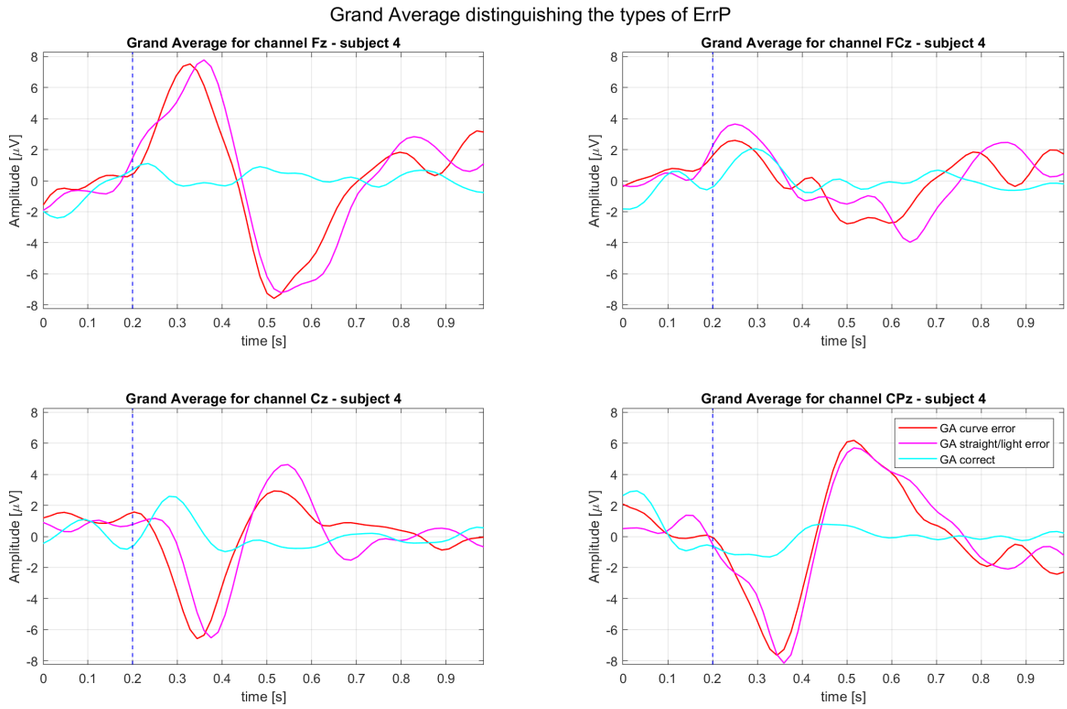

Figure 2: Grand Averages for subject 4: CAR was applied as spatial filter. The red and the pink lines represent respectively the average on the error trials on the curve sections of the game and the average on the error trials on the light/straight sections of the game. The blue one represents the average signal for the correct trials. The dotted line represents the time range onset from which the signal was spatially filtered.

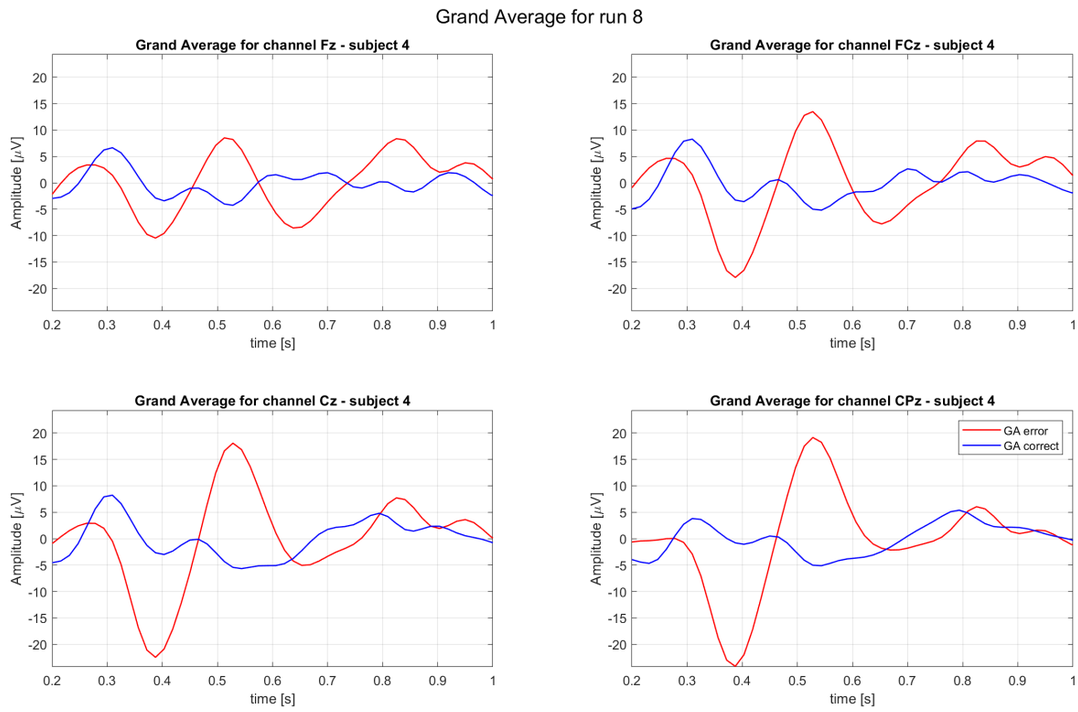

Figure 3: Grand Averages for subject 4 and for run 8. The CCA was applied as spatial filter: the data belongs to the filtered testset. The red line represents the average on the error trials, while the blue one the average signal for the correct trials.

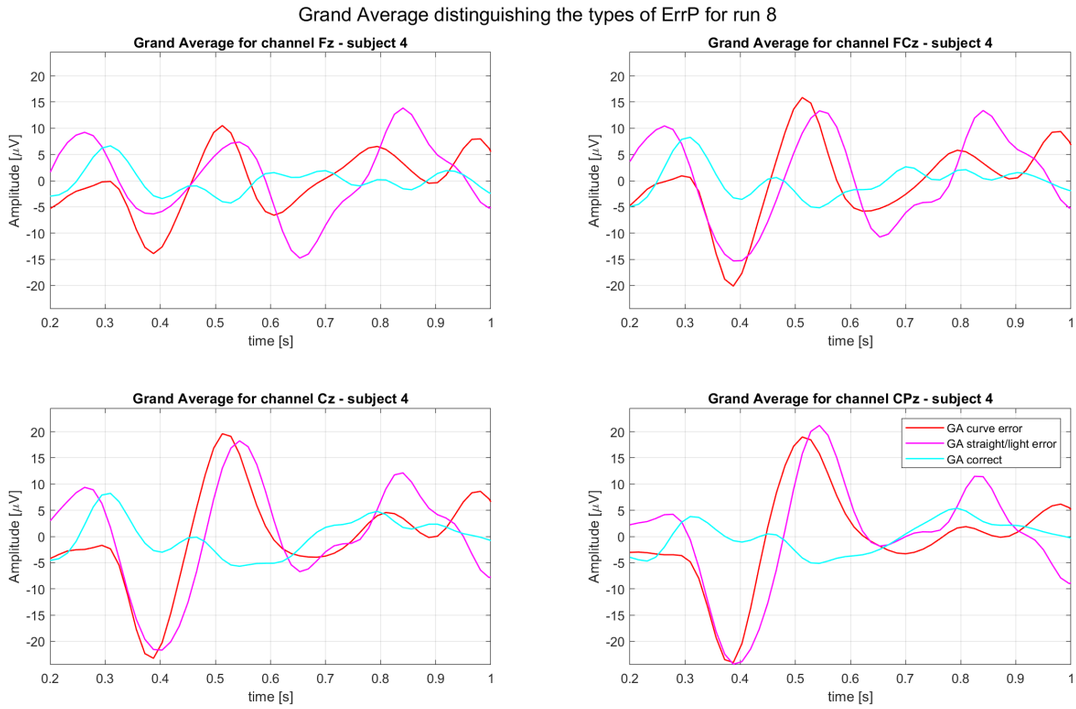

Figure 4: Grand Averages for subject 4 and for run 8. The CCA was applied as spatial filter: the data belongs to the filtered testset. The red and the pink lines represent respectively the average on the error trials on the curve sections of the game and the average on the error trials on the light/straight sections of the game. The blue one represents the average signal for the correct trials.

We have chosen to show in this part of the report the grand averages visualization for the fourth subject (however, the visualization of grand averages was computed for every subject). Figures 1 and 2 show the grand averages of the signals when the Common Average Reference (CAR) filter was applied, while Figures 3 and 4 show the grand averages with the Canonical Correlation Analysis (CCA) filtering method. The visualization of the signals filtered with CAR was done in the time range going from the beginning of the feedback (meaning that a command was sent and the avatar on the game moved, providing visual feedback to the subject) at time 0 until 1 second after. This was not the case for the signal filtered with CCA: since the CCA is a data dependent method, the filtering was executed after selecting the trial and test sets, which consisted of the signals in the time range going from 0.2 s to 1 second after the beginning of the trial. Therefore, the visualization is obtained in this period of time.

Moreover, in figure 2 and 4 we distinguished among the two types of error: the first one is what we call “curve error” and represents the errors which arise in the sectors “Turn left” and “Turn right” of the game; the second one, which we call “straight/light error”, groups all the errors belonging to the sectors “Light” and “Straight”. On the contrary, Figures 1 and 3 represent the comparison between all the error trials (curve errors and light/straight errors together) versus the correct ones.

First of all, we can notice that the trends for the two types of error depicted in figures 2 and 4 are extremely similar: this is particularly evident for the CAR filtering, but it is also true for the CCA one. Moreover, this similarity occurs for every subject. Keeping this in mind, we decided therefore to group all the errors together when considering the features used to feed the classifier.

For the fourth subject, we can notice that the signals obtained averaging with CAR the error and correct trials are very distinguishable. This is true especially for channels Fz, Cz and Cpz. Moreover, in the last two ones, the shape of the error potential can be clearly seen. The shape of the Error Potential is noticeable also for the signals filtered with CCA, especially for channels Fcz, Cz and CPz.

Furthermore, the two filtering methods return very similar outputs, particularly for channels Cz and CPz: the two filtered signals show the very same trend and the two peaks, the negative and the positive one, happen to be at the same time.

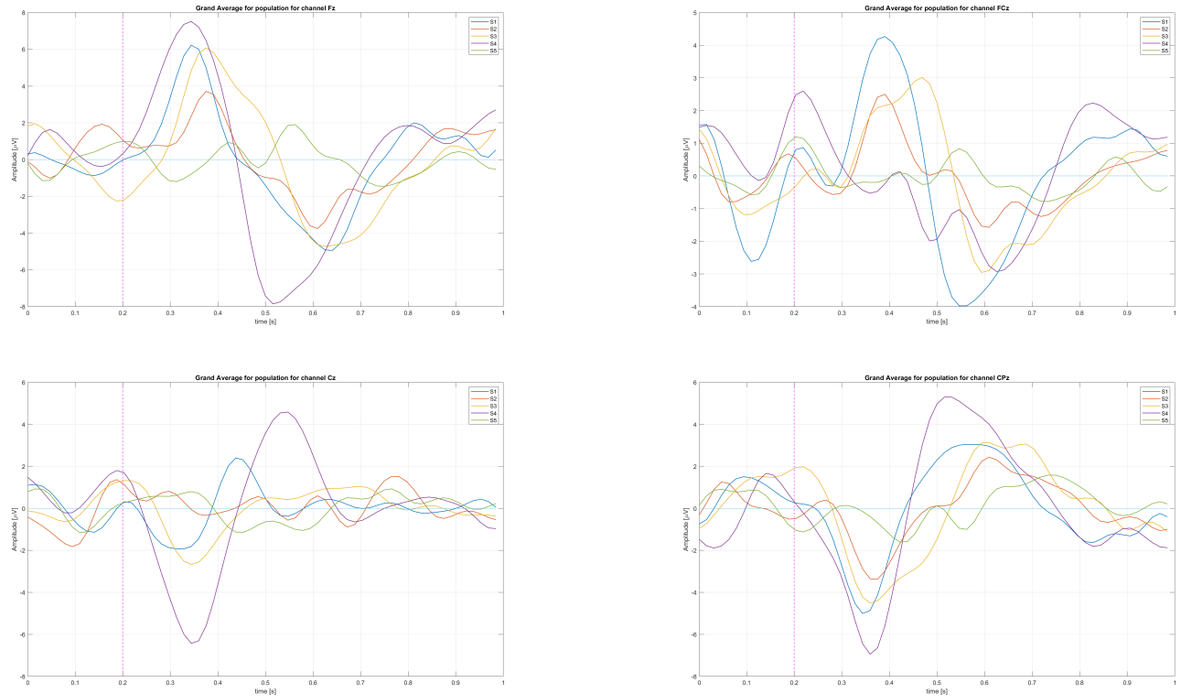

Figure 5: Spatial filter applied: CAR. The 5 lines in each box represent the signal obtained subtracting the average on the correct trials for each subject to the average of the error trials, for channel Fz, FCz, Cz and CPz. The dotted line represents the time range onset from which the signal was spatially filtered.

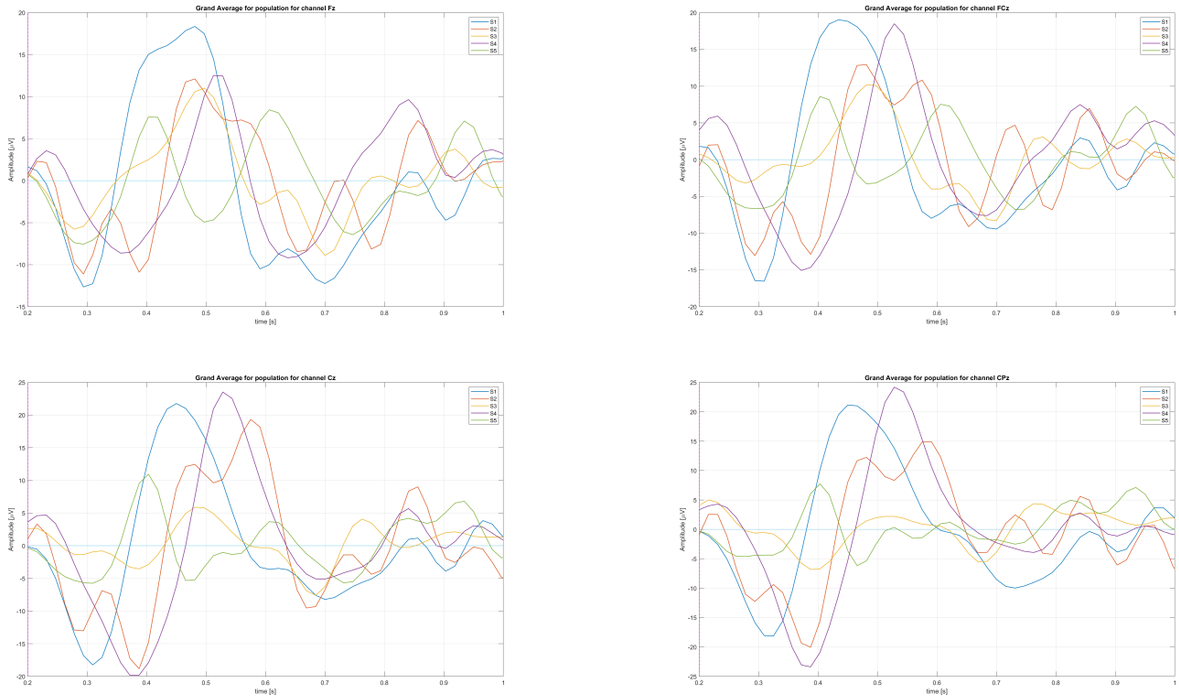

Figure 6: Spatial filter applied: CCA. The 5 lines in each box represent the signal obtained subtracting the average on the correct trials for each subject to the average of the error trials, for channel Fz, FCz, Cz and CPz.

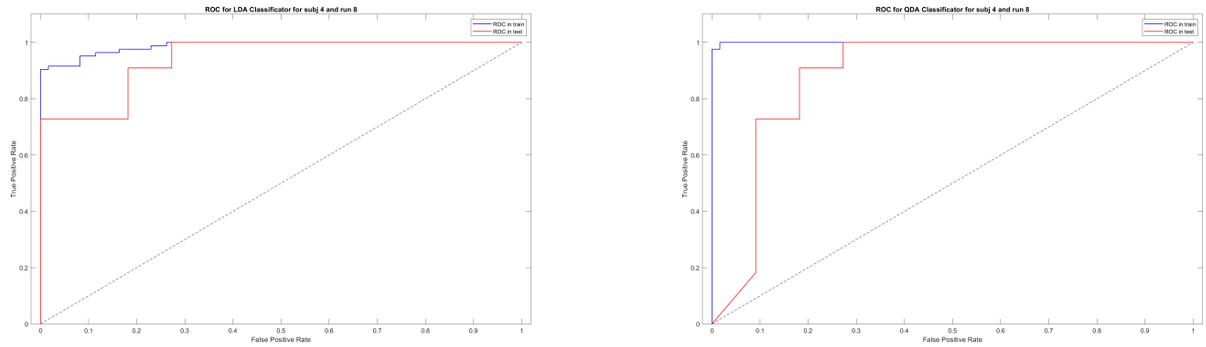

Figure 7: ROC curve plots for subject 4 and for run 8. The classifier applied is LDA and QDA considering all features for training.

For what it concerns the analysis on the overall population:

The visualization of the signals obtained with CAR shows the usual shape of the error potential for more or less all the subjects, with the amplitude and the latency of the peaks changing a bit from one to another. This shape is particularly evident in channels Cz and Cpz, but also in channel Fz. The same trend is also identifiable in the signals filtered with CCA. It is noteworthy to say that in this last case the differences in amplitude and latency for every subject are higher, but the trend of the signal depicted remains more stable across the channels for each subject considered.

After we processed the data, we developed LDA and QDA classifiers and gathered the relevant metrics to be able to evaluate our models. In general, we have eight different cases to observe, these depend on the data filter applied (CAR, CCA), on whether we apply feature selection based on Fisher Score, and the model type itself (LDA, QDA). The obtained average values of the metrics described in Methods section are presented in Appendix (Tables 1, 2, 3, 4, 5, 6, 7, 8).

What we noticed is that indeed with selected features using Fisher Score, we managed to increase the accuracy on test set and reduce the overfitting present in some of the models. The overfitting can be distinguished by looking at the accuracy obtained on train set, where the value is pretty high and $\geq 90\%$, while at the same time, the accuracy on test set is low. An example of such cases can be seen in Table 3, where the train accuracy is 0.94 and test accuracy is 0.66 for subject number 5.

It is also interesting to see, that the performance of the models is a lot better when we apply CCA filter in comparison to CAR. Moreover, the LDA model performs better than QDA model both with and without a feature selection step. We noticed, that the overall performance of the classifiers for subject number 5, based on accuracy metrics for test set, stays in range 66-79%, whereas other subjects even get to the point of 90% of test set accuracy. This could be explained by the possible signal distortions in terms of poor differential features when comparing ErrPs and 'no-error' trials. As for FPR metrics, it is expected that the rate is lower when the accuracy is higher, which we observe in the results. It can be also observed from the Grand Averages plots (Figures 5, 6), that subject 5 has the least distinguishable ErrP in comparison to other subjects, that supports our suggestion.

The trend of the ROC curves (Figure 7) confirms the good results of the classification: the percentage of the true positive rate is much higher than the false positive one and the evolution of the curve is very distinguishable from the bisector line at 45 degrees, which represent the results we could have obtained with a random classifier.

4 CONCLUSION

Looking at the overall results, what emerges is that both CAR and CCA filtering methods were able to filter the signal and to highlight the shape of the recorded Error Potential. Moreover, each one of them led to good results in classification accuracy.

The choice of training two different types of classifiers, a LDA and a QDA one, in order to see if the addition of a further degree of freedom could have increased the classification performance, led us to reconfirm that

the LDA, as literature implied, at least between these two methods, was the best algorithm to detect the Error Potential.

We decided also to see if with a previously done feature selection (based on a 0.4 Fisher Score's threshold) would improve the performances of the classifier. This is exactly what happened: the classification accuracy on the test set improved with the implementation of feature selection, if compared to the results obtained considering all the features together.

Beyond the aim of the project, an alternative method of analysis would have been to implement a validation procedure before operating on the test set. This could have allowed us to choose the best threshold for the feature selection without having to evaluate the final performances of the classifier on the same data on which the classifier was trained, leading to a more robust procedure.

5 APPENDIX

Table 1: LDA classifier mean statistics; All features included; CAR filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.895	0.943	0.834	0.879	0.944	0.231	0.173
2	0.858	0.927	0.806	0.867	1	0.241	0.188
3	0.921	0.968	0.851	0.910	1	0.196	0.151
4	0.955	0.987	0.899	0.925	1	0.108	0.102
5	0.813	0.912	0.734	0.831	0.875	0.383	0.233

Table 2: LDA classifier mean statistics; Fisher Score feature selection applied with threshold $FS(k) \geq 0.4$ for every k^{th} feature; CAR filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.844	0.875	0.765	0.779	0.895	0.311	0.279
2	0.761	0.840	0.708	0.764	0.889	0.355	0.329
3	0.933	0.957	0.900	0.905	1	0.154	0.145
4	0.938	0.958	0.881	0.887	1	0.162	0.146
5	0.861	0.883	0.781	0.798	0.905	0.286	0.276

Table 3: QDA classifier mean statistics; All features included; CAR filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.860	0.989	0.771	0.945	0.888	0.255	0.040
2	0.839	0.997	0.791	0.967	0.909	0.357	0.065
3	0.928	0.997	0.847	0.966	1	0.176	0.049
4	0.921	0.999	0.879	0.984	1	0.183	0.020
5	0.684	0.988	0.660	0.941	0.875	0.363	0.081

Table 4: QDA classifier mean statistics; Fisher Score feature selection applied with threshold $FS(k) \geq 0.4$ for every k^{th} feature; CAR filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.817	0.899	0.752	0.815	0.895	0.332	0.269
2	0.783	0.879	0.743	0.825	0.889	0.389	0.285
3	0.913	0.969	0.868	0.928	1	0.177	0.114
4	0.910	0.978	0.862	0.917	1	0.193	0.106
5	0.854	0.884	0.779	0.792	0.952	0.321	0.333

Table 5: LDA classifier mean statistics; All features included; CCA filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.929	0.963	0.859	0.895	0.944	0.164	0.138
2	0.836	0.919	0.748	0.843	0.933	0.293	0.208
3	0.869	0.926	0.804	0.868	0.929	0.298	0.188
4	0.950	0.978	0.891	0.928	1	0.134	0.091
5	0.776	0.889	0.714	0.795	0.833	0.465	0.298

Table 6: LDA classifier mean statistics; Fisher Score feature selection applied with threshold $FS(k) \geq 0.4$ for every k^{th} feature; CCA filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.882	0.919	0.837	0.852	0.947	0.213	0.193
2	0.844	0.857	0.771	0.777	1	0.236	0.304
3	0.898	0.902	0.862	0.859	1	0.238	0.215
4	0.958	0.961	0.885	0.899	1	0.145	0.155
5	0.843	0.853	0.751	0.773	0.842	0.357	0.375

Table 7: QDA classifier mean statistics; All features included; CCA filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.871	0.994	0.797	0.962	1	0.198	0.030
2	0.787	0.996	0.709	0.968	0.909	0.414	0.036
3	0.885	0.995	0.803	0.958	0.923	0.298	0.068
4	0.888	0.998	0.846	0.977	1	0.228	0.032
5	0.685	0.995	0.695	0.965	0.875	0.536	0.066

Table 8: QDA classifier mean statistics; Fisher Score feature selection applied with threshold $FS(k) \geq 0.4$ for every k^{th} feature; CCA filter applied.

Subject	AUC test	AUC train	Accuracy test	Accuracy train	Max accuracy test	FPR test	FPR train
1	0.877	0.928	0.819	0.865	0.944	0.227	0.179
2	0.865	0.892	0.762	0.809	0.947	0.266	0.265
3	0.891	0.936	0.869	1	1	0.267	0.218
4	0.942	0.961	0.862	0.886	0.933	0.129	0.142
5	0.795	0.846	0.758	0.785	0.941	0.443	0.388

Bibliography

[1] Pierre W. Ferrez and JosÉ del R. Millan. “Error-Related EEG Potentials Generated During Simulated Brain–Computer Interaction”. In: *IEEE Transactions on Biomedical Engineering* 55.3 (2008), pp. 923–929. DOI: 10.1109/TBME.2007.908083.

[2] Iñaki Iturrate et al. “Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control”. In: *Scientific Reports* 13893 (Sept. 2015). DOI: 10.1038/srep13893.

[3] Fumiaki Iwane et al. “Invariability of EEG error-related potentials during continuous feedback protocols elicited by erroneous actions at predicted or unpredicted states”. In: *Journal of neural engineering* 18 (Apr. 2021). DOI: 10.1088/1741-2552/abfa70.

[4] Fumiaki Iwane et al. “Spatial filters yield stable features for error-related potentials across conditions”. In: Oct. 2016, pp. 000661–000666. DOI: 10.1109/SMC.2016.7844316.

[5] Catarina Lopes-Dias, Andreea-Ioana Sburlea, and Gernot Müller-Putz. “Online asynchronous decoding of error-related potentials during the continuous control of a robot”. In: *Scientific Reports* 9 (Nov. 2019). DOI: 10.1038/s41598-019-54109-x.