Predicting a Subject’s Card with the Muse-S Headband

Theodora Gaiceanu[1] **Diego Figueroa**[2] **Daniel Díaz**[3]

[1]th6636ga-s@student.lu.se [2]di4642fi-s@student.lu.se [3]da5736di-s@student.lu.se

Abstract: In this project, we have developed a Brain-Computer Interface system to predict the card a subject is focusing on in real-time with P300 evoked potentials (P300 EPs) recorded with a Muse-S headband. Firstly, we built our dataset, named 4-card dataset: we told subjects to focus on one out of four cards, then showed the cards in intervals of 200 ms, labelling differently the P300 EPs whether the subject was focusing on the current displayed card or not. Secondly, we tested different models to filter and classify the P300 EPs by telling subjects to focus on a card of their choice, then showed cards in intervals of 200 ms to see which one evoked the P300 EPs that correspond to focusing. We conducted offline classification using an existing dataset in order to analyze the performance of several models. Afterwards, we conducted offline classification on the 4-card dataset using the models that performed satisfactory on the existing dataset and some new models. The best model for the existing dataset (obtaining a precision of 0.32 and a recall of 0.69) and for the 4-card dataset (with an average precision of 0.65 and an average recall of 1) used ERP covariances and Xdawn for spatial filtering and Logistic Regression for classification. Lastly, we tested this model for the real-time experiment, obtaining an average precision of 0.07 and an average recall of 0.71.

1. Introduction

The goal of this project is to develop a Brain-Computer Interface (BCI) system that predicts the card a subject is focusing on from brain signals (EEG) recorded with a Muse-S headband.

BCI systems record brain signals, send them to a computer, and translate them into commands [1]. Their applications are broad, and they appear in fields such as medicine, smart environments, education, and neuromarketing. A BCI system has four major components: data recording, data pre-processing (noise reduction and signal enhancement), feature extraction, and classification [2]. The data can be recorded using invasive or non-invasive methods. In this project, we use non-invasive methods, respectively a Muse-S headband device.

There are many different technologies to record brain signals non-invasively. The most popular are: functional magnetic resonance imaging (fMRI), functional near-infrared spectroscopy (fNIRS), magnetoencephalography (MEG), and electroencephalograms (EEGs). The Muse-S headband, for example, is a non-invasive EEG [3] that measures the frequency bands named alpha (frequencies between 8 and 12 Hz), beta (frequencies between 13 and 30 Hz), theta (frequencies between 4 and 8 Hz) , delta (frequencies lower than 4 Hz), and gamma (frequencies higher than 30 Hz) [4]. In comparison with other devices, it is small, cheap, and has a high temporal resolution. EEG devices record the electrical activity using electrodes placed on the scalp of the user [2]. As far as the BCI electrical signals are regarded, they can be analyzed by using several approaches, two of them being: evoked potentials (EP), an approach that studies the effect of the different conditions that trigger the neural response [2], and event related desynchronization/synchronization (ERD/ ERS), an approach that detects the brain oscillations that are not necessarily related to external stimuli [2]. In other words, the EP is the electrical response that is observable in the recorded brain data after being triggered by a stimulus. EP can be further divided into event-related potential (ERP) and steady state evoked potential (SSEP) [2]. ERP is a neural response to a stimulus (an image, an instruction, a physical event). In our project, we ask the user to actively focus on a specific card, therefore we are interested in EP signals. In particular, we have focused on the P300 component of event-related potential (ERP) signals, a pattern detectable approximately 300 ms after the presence of a stimulus.

A BCI system poses several challenges. The signals have noise from other brain activity not related to the experiment [2] and are nonstationary ([5] and [6]). Another important challenge is session/subject variability (the signals change over time for different sessions, sometimes, even within the same session, and they change between subjects). These problems, however, can be solved. The noisiness can be decreased by improving the signal level, decreasing the noise ratio, and applying frequency filters. Transfer Learning [7] can be used in order to utilize old data as calibration/ training data, despite the large variations in the EEG signal from one subject to another or in between different sessions of the same subject.

The remainder of the paper goes as follows: Section 2 describes the equipment used to record the EEG signals; Section 3 presents how the BCI system works; Section 4 details the experiment; Section 5 describes the datasets; Section 6 details the models; Section 7 illustrates the results of the proposed models for both offline and real-time classification; Section

8 summarises the project and presents some future improvements.

2. Equipment

Figure 1. The Muse-S headband [3].

The EEG signals have been recorded with a Muse-S (Figure 1). The Muse-S is a low-cost, portable headband broadly used for guided meditation and sleeping that has shown promising results measuring Event-Related Potentials (ERPs), particularly, the P300 component [3]. The Muse-S records raw EEG data (more precisely, the frequency bands named alpha - frequencies between 8 and 12 Hz, beta - frequencies between 13 and 30 Hz, theta - frequencies between 4 and 8 Hz , delta - frequencies lower than 4 Hz, and gamma - frequencies higher than 30 Hz [4]) with two sensors placed on the forehead and one sensor on the top of each ear. The sensors (or electrodes) are AF7, AF8, TP9, and TP10. The used sampling rate is 256 Hz.

Even though the small number of sensors is a downside, there have been other good reasons to choose this device for our project, such as: the low price (approximately 4000 SEK), the quick calibration (setting up the device takes only a few minutes), and the possibility of detecting ERPs, which our project is based on.

3. Software setup

The code was written in Python and the application uses the demos provided by our project supervisor. The software consists mainly of three parts: data gathering, machine learning process, and deployment. The first part is the receiver, while the second and the third function as the analyser. Figure 2 illustrates the structure of the code used for the experiments.

Receiver
The Muse-S uses the Lab Streaming Layer (LSL) protocol to manage the networking, the time-synchronization, the real time access, the centralized collection, the viewing, and the recording of the EEG signals [8]. Therefore, we have used the *muselsl* Python library to acquire the EEG signals broadcasted by the Muse-S, and the *timeflux* Python library to process them in real-time [9]. Timeflux uses YAML syntax to describe the applications. Valid applications have to be defined as directed acyclic graphs (DAG), and their nodes define the processing steps [10].

The receiver corresponds to the data gathering part from Figure 2. In the code, there is a Receive_LSL timeflux process, having multiple nodes: eeg_LSL_receiver and markers_LSL_receive that receive the data from the LSL stream, and other nodes that publish the raw EEG signals to the broker (that connects the receiver and the analyser part) to deliver them to the other processes (the connection between Data Gathering and Machine Learning process from Figure 2).

Analyser
Figure 2 also illustrates the structure of the analyser, the part of the code that fits models that we later use for prediction/inference (Machine Learning Process and Deployment). Its main node is fit_predict, where the selected model is trained with the EEG signals broadcasted by the receiver and, once trained, it makes the predictions for the real-time experiment. The training data is divided into features and labels, and it is saved in .sav files.

4. Experiment

The experiments are performed in a calm room. An assistant places the Muse-S headband and, occasionally, noise-cancelling headphones on the subject, and instructs him to stay as calm and static as possible. The assistant reads the frequency of the signals and only starts the experiment when the frequency of all four signals is stable. Then, the assistant gives the subject a computer and runs the experiment. The experiment is self-explanatory to avoid further interactions between the assistant and the subject.

The experiment consists of showing a random card every 200 ms (with a random standard deviation of 100 ms), and analyzing how the subject responds when his card is shown. The deck consists of four cards (illustrated in Figure 3) that have been chosen to be as different as possible: the Ace of hearts, the 3 of diamonds, the King of spades, and the 9 of clubs. Each card has a suitable ID.

In the first part of the experiment the program instructs the subject to count the number of times the Ace of hearts appears. The Ace of hearts appears randomly, but to avoid signal overlapping, the program waits for at least three other cards to be shown between each Ace of hearts. The EEG data is then collected, labelled differently (whether the card shown was the Ace of hearts or not; the labels have also timestamps to be able to match each card to the moment in time it appeared), and sent to the analyser as training data (see Data Gathering part from Figure 2). This process is repeated for ten runs, each of them showing thirty cards. After each run the subject is reminded to count the number of times the Ace of hearts appears. The data is saved in epoch batches with an epoch length of 256 time-samples per channel.

In the second part of the experiment the program presents the four cards in the deck and instructs the subject to choose one. The program then shows cards randomly, labels the cards according to the user's decision ('0' if the card is the one picked by the user, '1' if it is a different card), and displays the prediction in real time (see Deployment part from Figure 2).

The experiment takes approximately ten minutes.

5. Datasets

The project uses two different datasets: the BNCI2014009 dataset, used to test the different feature extraction and classification methods, and the 4-card dataset, used to evaluate

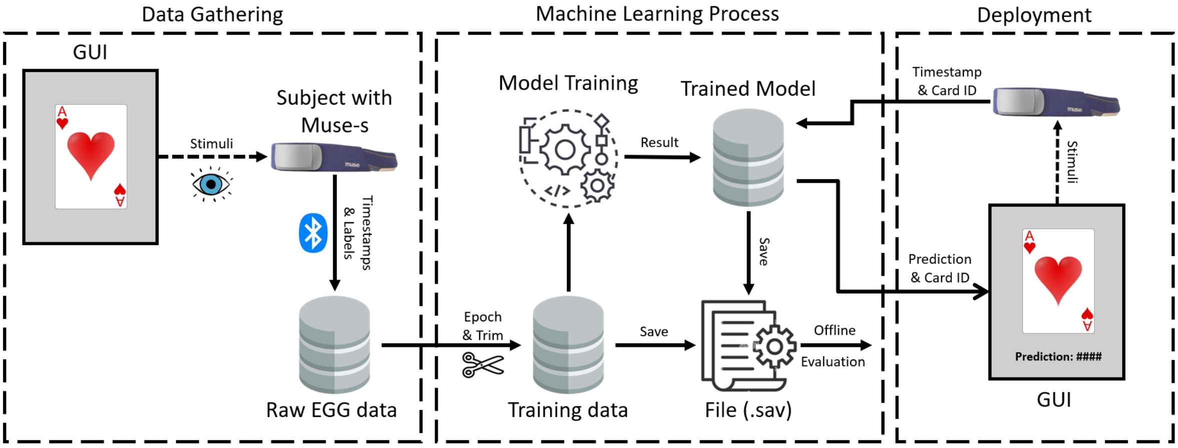

Figure 2. A graph illustrating the structure of the code used for the experiments.

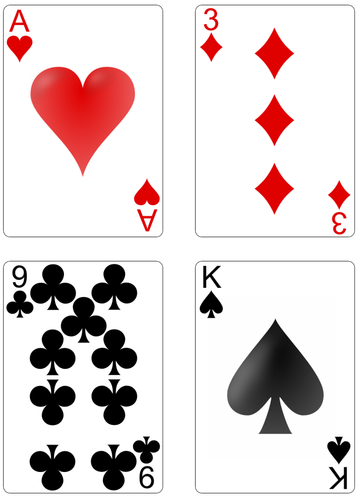

Figure 3. The four cards being used for the experiment.

these methods in an environment similar to the real-time experiments.

BNCI2014009 dataset

BNCI2014009 is a dataset from MOABB, an open-source framework that aims to “build a comprehensive benchmark of popular BCI algorithms applied on an extensive list of freely available EEG datasets” [11]. It consists of P300 evoked potentials recorded in three different sessions from ten healthy subjects with experience in recording EEG data. For simplicity, however, we have reduced the dataset to four subjects.

Here subjects were asked to focus on one out of 36 different characters and the P300 evoked potentials were recorded at 256 Hz with a low-pass, high-pass filter (with cutoff frequencies 0.1 Hz and 20 Hz) [12]. The quality of the data is very-high, as they were recorded under laboratory conditions. We could not obtain the same quality for our experiments.

The total number of datapoints is 1728. But the dataset is imbalanced, as it contains 288 ’target’ labels (corresponding to the character the subject focused on) and 1440 ’non-target’ labels (corresponding to other characters).

4-card dataset

The 4-card dataset is the name of the data recorded from Muse-s when running the first part of our experiment (in which the subject is asked to count the number of Aces of hearts) and it is used in order to see what performances some classifiers have when this data is at input. As this analysis aimed to decide what model was suitable for our experiment, the recorded data from only two subjects were used.

The 4-card dataset consists of the P300 evoked potentials recorded in a session for two different subjects. Each P300 evoked potential is labelled 0 if the subject was focusing on the card producing it, or 1, if not.

Table 1. Labels of the entries in the 4-card dataset.

Label	Subject 1	Subject 2	Total
Target (0)	52	44	96
Non-target (1)	308	256	564
Total	360	300	660

The distribution of the labels in the 4-card dataset (illustrated in Table 1) is, roughly, 1:6. It is uneven because the subject is asked to only focus on one out of the four cards shown and, furthermore, the program waits for at least three non-target cards to be shown after every focused card (always the Ace of hearts).

6. Models

As it was mentioned before, EEG devices extract data by using electrodes placed on the subject’s scalp. Since there are several electrodes and the EEG devices have high sampling rates, the amount of collected data can be very large. Just using all the data results in a very high dimensional feature vector. Consequently, a feature extraction step is needed.

For example, Power Spectral density (PSD) extracts a feature vector based on the frequency content. PSD estimator "represents the proportion of the total signal power contributed by each frequency component of a voltage signal" [13].

Riemannian Geometry

To classify EEG signals, one can: pre-process the signal, extract the sources with spatial filters, extract the feature vector, and classify the feature vector using a vector-based (or distance-based) classifier [14]. All of these steps (especially the second) involve covariance matrices. In this context, a Riemannian Geometry approach can be helpful, showing promising results in the last years ([14], [15]). In this approach, the signal is pre-processed, represented as a covariance matrix, and classified [14], as it will be shown in the following paragraph.

It is known that covariance matrices are symmetric and positive-definite (SPD). A Riemmannian mainfold (M, g) is a "real, smooth manifold M equipped with a positive-definite inner product g_p on the tangent space T_pM at each point p" [16]. Consequently, one could say that the covariance matrix between two electrodes from a EEG device exists on a Riemannian mainfold [15]. Considering two points (two covaraince matrices) C_1 and C_2, the Riemannian distance between them can be expressed as [15]:

$$\delta_G(C_1, C_2) = \sqrt{\sum_{i=1}^{n} log^2 \lambda_i}, \tag{1}$$

where λ_i are the real eigenvalues of matrix $C_1^{-1}C_2, \lambda_i > 0$, and G is the geometric mean of C_1 and C_2 [15].

In this context, ERP covarainces may be used. ERP covariance is an estimation of a special form covariance matrix dedicated to ERP processing [17]. They take the mean of trial components X_i to build a prototyped response P:

$$P = \frac{1}{N} \sum_{i}^{N} X_i \tag{2}$$

and then compute a super trial $\widetilde{X_i}$ by concatenating P and X [17].

$$\widetilde{X_i} = \begin{bmatrix} P \\ X_i \end{bmatrix} \tag{3}$$

The covariance estimation uses the super trial to take into account the temporal structure of the signal [17]. More explicitly, using ERP covariances is a way of designing a covariance matrix from EEG epochs that captures the temporal component (which regular sample covariance matrices do not).

With EEG data, after having a covaraince matrix representing an epoch of data, the distance δ_G is used to calculate the means of the labels. The next step is to do a projection to a euclidean tangent space and classify the data using a distance-based classifier, such as Minimum Distance to Mean.

Spatial Filters

Spatial filters are used for noise reduction, as they increase the signal-to-noise ratio. We have used two different spatial filters: Xdawn and Common Spatial Pattern (CSP) algorithm.

Xdawn is an unsupervised method that estimates spatial filters in order to enhances P300 evoked potentials [18]. It does this by projecting the raw input signal on the estimated evoked space.

In general, the CSP algorithm maximizes the discriminability of two classes by using spatial filters [19]. For two windows X_1 and X_2 of a multivariate signal, the CSP algorithm computes the following [20]:

$$w = argmax_w \frac{||wX_1||^2}{||wX_2||^2} \tag{4}$$

This assures that the ratio of variance is optimal between the two windows [20].

With EEG data, CSP linearly transforms the EEG measurements with the following expresssion:

$$Z_{b,i} = W_b^T E_{b,i}, \tag{5}$$

where $E_{b,i}$ is the single-trial EEG measurement, $Z_{b,i}$ is actually $E_{b,i}$ after spatial filtering, and W_b is the CSP projection matrix [21]. The CSP projection matrix W_b produces features having optimal variances for separating two classes of EEG measurements [21].

Both CSP and Xdawn reduce the number of channels. In general, spatial filtering modifies the amplitude of the EEG in every electrode with respect to the weight composition of the voltage values in different electrodes. [22].

Classifiers

We have used four different classifiers: Logistic Regression, K-nearest neighbours (KNN), Linear Discriminant Analysis (LDA), and Minimum Distance to Mean (MDM).

Logistic Regression and KNN are broadly used traditional classifiers that do not require an explanation. LDA is commonly used in supervised classification tasks. It tries to find the optimal linear combination of features to differentiate distinct classes [23]. For a binary classification, LDA tries to find the optimal hyperplane to differentiate the labels.

MDM classifies the results by the nearest centroid [24]. It computes a centroid estimation with respect to a certain metric (in the EEG case, this metric is the Riemannian distance) for all known classes. For the new points, it uses the nearest centroid.

7. Results

BNCI2014009 dataset

The six different models use: (1) Logistic Regression for classification, (2) Xdawn for spatial filtering and K-nearest neighbours (KNN) for classification, (3) Xdawn for spatial filtering and Logistic Regression for classification, (4) Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification, (5) CSP for spatial filtering and LDA for classification, and (6) ERP covarainces and MDM for classification.

The models were trained on Subject 1 and evaluated on Subjects 1, 2, 3, and 4 (illustrated in Table 2) with data from all three different sessions. The best model is (4), the model that uses Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification. It is the most accurate model for all four subjects, with accuracies of 93%, 96%, 86% and 95%. The best results can be seen in bold in Table 2. The second best model is (3), the model that uses Xdawn for spatial filtering and Logistic Regression for classification. A model that is clearly not suitable for the BNCI2014009 dataset is

Table 2. Accuracy of the models for four subjects from the BNCI2014009 dataset.

Model	Filter	Feature	Classifier	Subject 1	Subject 2	Subject 3	Subject 4
1	None	stacked channels	LR	0.90	0.95	0.82	0.94
2	Xdawn	stacked channels	KNN	0.80	0.87	0.71	0.89
3	Xdawn	stacked channels	LR	0.87	0.93	0.77	0.92
4	Xdawn	ERP covariances	LR	**0.93**	**0.96**	**0.86**	**0.95**
5	CSP	stacked channels	LDA	0.57	0.61	0.57	0.58
6	None	ERP covariances	MDM	0.93	0.96	0.74	0.83

(5), the model that uses CSP for spatial filtering and LDA for classification.

The accuracy of every model is the lowest for Subject 3, suggesting that the data for Subject 3 is more noisy than for the other Subjects.

However, as the data is imbalanced, just the accuracy is not a reliable metric. In case of a naive classification (when all target samples are classified as non-target), we would end up with 83% accuracy. As it can be seen from Table 2, model (4) has a higher accuracy than 83% for each subject, which means it could be a suitable model. But a further analysis was also done.

The confusion matrix was also analyzed for model (4). As it can be seen in Table 3, there are 193 false positives and 41 false negatives. Consequently, 193 out of 288 target samples were misclassified (approximately 67%).

Table 3. Confusion matrix of model (4) for BNCI2014009 dataset

		Confusion matrix	
		Target	Non-target
BNCI2014009	Target	95	193
	Non-target	41	1399

When working with imbalanced datasets, some good metrics for evaluation are the precision and the recall. The precision is expressed as:

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

and the recall is defined as:

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

In the BNCI2014009 case, the precision is 0.32, and the recall is 0.69. All these results mean that model (4) is overfitted, but still suitable for the given task.

The parameters used for model (4) were chosen in the following way: firstly, the default parameters were used, then they were changed and the accuracy was observed. The parameters that gave the best results (the results from Table 2) were kept. The parameteters for model (4) are: the number of Xdawn filters used per class, the regularization algorithm used for covariance matrix estimation,and the regularization algorithm for Logistic Regression. In case of model (4), five Xdawn filters are used, the regularization algorithm for covaraince estimation is Ledoit-Wolf, and L1 regularization is used for Logistic Regression. The pipeline for model (4) is: do spatial filtering by using Xdawn and estimate the covariance matrices in a special form, suitable for ERP processing, do a tangent space projection of the covariance matrices in order to convert them to euclidean vectors (but keeping the inner structure of the mainfold), apply Logisitc Regression to classify the data.

4-card dataset (offline classification)

The data come from two subjects, one of them having 360 samples, and the other one having 300 samples (see Table 1). The models were trained on half of the data (180, respectively 150 samples) and evaluated on the other half. The results are illustrated in Table 4. The most accurate models for Subject 1 are as follows: model (3), which uses Xdawn for spatial filtering and Logistic Regression for classification; model (4), which uses Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification; model (6), which uses Xdawn for spatial filtering and LDA for classification. All of them have an accuracy of 91%. The most accurate models for Subject 2 are: (2), which uses Xdawn for spatial filtering and KNN for classification; (4), which uses Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification; (8), which uses ERP covariances and Logistic Regression for classification; and (9), which uses PSD and Logistic Regression for classification. All of them have an accuracy of 82%. The best results for each Subject can be seen in bold in Table 4. Taking the different accuracies into account, however, the best model for the 4-card dataset is model (4), which uses Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification. The least accurate model for both subjects is model (5), the model that uses ERP covarainces and MDM for classification, with an accuracy of 79% for Subject 1 and of 71% for Subject 2.

An explanation for which every model has a higher accuracy for Subject 1 than for Subject 2 could be that the data is affected by the environment, the conditions, and the subject's overall mental state. Furthermore, as mentioned above, the 4-card dataset is imbalanced: there are roughly six unfocused entries for every focused entry. In case of a naive classifier (which would classify all target samples as non-target), the accuracy of the model for both subjects would be 85%. Model (4) has a higher accuracy than 85% for Subject 1, but it has a slightly lower accuracy for Subject 2. Consequently, to ensure that model (4) is suitable for our project, we look at the confusion matrices for both subjects (illustrated in Figure 4 left and Figure 4 right). Subject 1 does not have false negatives and has 8 false positives, so 30% of the focused labels were misclassi-

Table 4. Accuracy of the models for the 4-card dataset.

Model	Filter	Feature	Classifier	Subject 1	Subject 2
1	None	stacked channels	LR	0.90	0.76
2	Xdawn	stacked channels	KNN	0.90	**0.82**
3	Xdawn	stacked channels	LR	**0.91**	0.77
4	Xdawn	ERP covariances	LR	**0.91**	**0.82**
6	None	ERP covariances	MDM	0.79	0.71
7	Xdawn	stacked channels	LDA	**0.91**	0.80
8	None	ERP covarainces	LR	0.90	**0.82**
9	None	PSD	LR	0.84	**0.82**

fied; Subject 2 does not have false negatives and has 11 false positives, so 40% of the focused labels were misclassified.

The precision of model (4) is 0.7 for Subject 1 and 0.6 for Subject 2. The recall is 1 in both cases, as none of the subjects has false negatives samples. Consequently, model (4) is slightly over-fitted, but still suitable for our experiment.

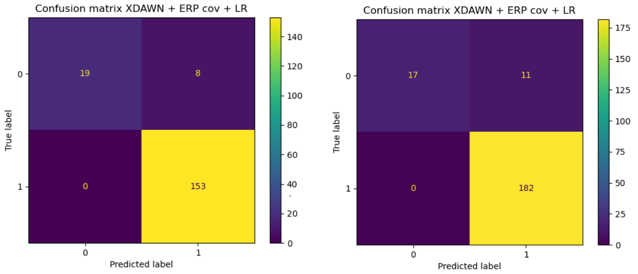

Figure 4. Confusion matrix of model (4) for Subject 1 (left) and Subject 2 (right).

The methodology for choosing the parameters of model (4) for the 4-card dataset was the same as in the BNCI2014009 case: firstly, the default parameters were used, then different combinations were tried, and finally the best combination was kept. The parameteters for model (4) are: the number of Xdawn filters used per class, the regularization algorithm used for covariance matrix estimation, and the regularization algorithm for Logistic Regression. In case of model (4), two Xdawn filters are used, the regularization algorithm for covaraince estimation is Ledoit-Wolf, and L1 regularization is used for Logistic Regression. The pipeline for model (4) is: do spatial filtering by using Xdawn and estimate the covariance matrices in a special form, suitable for ERP processing, do a tangent space projection of the covariance matrices in order to convert them to euclidean vectors (but keeping the inner structure of the mainfold), apply Logisitc Regression to classify the data.

Real-time classification

In the real-time experiment we only used model (4), the model that uses Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification, and that achieves the best performance in both the BNCI2014009 dataset and the 4-card dataset.

We ran the real-time experiment in six different sessions for five different subjects (Subject 3 took part in the experiment twice). The data are different than the data recorded and analyzed in the 4-card dataset. In addition, at this phase of the experiment, there are four cards in the deck and the user is free to choose any card.

At the second session for Subject 3 and at the session for Subject 5, the BCI system did not classify any card as being target. At the rest of the sessions, the BCI system made the predictions illustrated in the Table 5. The fractions represent the number of target predictions over the number of total appearances of a particular card. The card the subject was focusing on is shown in bold in Table 5. The results differ greatly between subjects. The BCI system obtained the following results: a precision of 0.21 and a recall of 0.93 for Subject 1, a precision of 0.07 and a recall of 0.75 for Subject 2, a precision of 0.02 and a recall of 0.66 for Subject 3, and a precision of 0.01 and a recall of 0.5 for Subject 4. In general, the performance of the system in real time is worse than the performance obtained in offline (4-card dataset subsection).

Table 5. Real-time target predictions over total appearances.

Card	Subject 1	Subject 2	Subject 3	Subject 4
A	1/85	0/40	1/85	1/85
3	0/76	**3/41**	0/76	0/69
K	**15/71**	0/38	0/71	0/79
9	0/69	1/31	**2/68**	**1/67**
Total	16/300	4/150	3/300	2/300

We also ran a cross-session experiment for Subject 1, as our application performed the best for this Subject. We used data collected six days prior and, after adding only 60 new data-points, the program attempted to predict Subject's 1 card of choice. The results are illustrated in Table 6. They are quite similar to the ones obtained in a single session (illustrated in Table 5), the precision being 0.14 and the recall being 0.83. This means that that Subject's 1 previous data can be added in Subject's 1 future experiments without significantly decreasing the accuracy of the predictions.

Table 6. Real-time cross-session predictions for Subject 1.

Prediction	A	3	K	9	Total
Target	2	**10**	0	0	12
Non-target	83	**68**	73	64	288

8. Conclusion

We developed a BCI system to predict the card a subject is focusing on in real-time with the signals recorded by a Muse-S headband. We first tested different models with the BNCI2014009 dataset from MOABB and achieved promising results. Then, we tested an extended set of models with the 4-card dataset, a dataset designed and collected by us that consists of P300 evoked potentials, recorded with the Muse-S device, and labelled differently whether the subject was focusing on the card evoking them or not. Once again, we achieved promising results. The best model for both datasets used Xdawn for spatial filtering, ERP covariances and Logistic Regression for classification, and, therefore, it was the only model we used for the real-time experiment. The performances of the models were high in the BNCI2014009 dataset and in the 4-card dataset, and significantly lower in the real-time experiment.

There could be several reasons why the offline predictions are more accurate than the real-time predictions. One could be that, in the offline case, the subjects were asked to focus on the same card (the Ace of hearts), but in real-time they are asked to choose any card they want. Moreover, in the real-time experiment, the fact that the Ace of hearts card (the card the subjects were asked to focus on during the first phase of the experiment) is shown in the deck causes misclassifications. Another reason could be that our application achieves better performances with subjects that are petite (like Subject 1 is) because the band coverages a larger area of the head. Last but not least, some subjects did not focus on just the card they picked when doing the second phase of the real-time experiment, which can have a negative impact on the model performances; they also tried to identify patterns in the shown deck of cards etc. So another aspect that should be taken into consideration is to have an experiment as interesting and as quick as possible, so that the subjects do not get tired or focus on other tasks.

The design of the application can be improved by switching the card a subject is asked to focus on at every run, by adding more cards, and by performing more real-time experiments. Furthermore, more classification methods could be used and the 4-card dataset could be extended to assist in choosing a better model. Doing a preprocessing step before filtering (such as implementing a low-pass, high-pass filter) could also improve the performance of the system by reducing the noise in the signal. Finally, Transfer Learning techniques could be used to reduce the amount of data the BCI system records in the real-time experiment, and to make the data more robust to noise, movements, etc.

References

[1] J. J. Shih, D. J. Krusienski, and J. R. Wolpawc. “Brain computer interfaces in medicine”. *Mayo Clinic Proceedings* **87**:3 (2012), pp. 268–279. DOI: 10.1016/j.mayocp.2011.12.008.

[2] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa. “Brain computer interfacing: applications and challenges”. *Egyptian Informatics Journal* **16**:2 (2015), pp. 213–230. ISSN: 1110-8665. DOI: https://doi.org/10.1016/j.eij.2015.06.002.

[3] *Muse headbands*. URL: https://choosemuse.com/how-it-works/?utm_source=Google&utm_medium=PaidSearch&utm_campaign=Non-Brand&gclid=CjwKCAjwjZmTBhB4EiwAynRmD1WYqQNv7PwqZBayIy8Kw%20s6y8Hepfn9sStjcpadZ8RMN0MYY9_MdqxoCE-0QAvD_BwE (visited on 2022-04-25).

[4] *The science of brainwaves - the language of the brain*. URL: https://nhahealth.com/brainwaves-the-language/.

[5] M. Kreuzer, E. F. Kochs, G. Schneider, and D. Jordan. “Non-stationarity of eeg during wakefulness and anaesthesia: advantages of EEG permutation entropy monitoring”. *Journal of clinical monitoring and computing* (2014). DOI: 10.1007/s10877-014-9553-y.

[6] T. Dikanev, D. Smirnov, R. Wennberg, J. L. P. Velázquez, and B. Bezruchko. “EEG nonstationarity during intracranially recorded seizures: statistical and dynamical analysis.” *Clinical Neurophysiology* (2005). DOI: 10.1016/j.clinph.2005.04.013.

[7] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu. “Transfer learning: a Riemannian geometry framework with applications to brain-computer interfaces”. *IEEE Transactions on Biomedical Engineering* **65**:5 (2018), pp. 1107–1116. DOI: 10.1109/TBME.2017.2742541.

[8] *Lab streaming layer*. URL: https://labstreaminglayer.readthedocs.io/info/intro.html (visited on 2022-04-25).

[9] *Muse LSL*. URL: https://github.com/alexandrebarachant/muse-lsl (visited on 2022-04-25).

[10] *Timeflux. concepts*. URL: https://doc.timeflux.io/en/stable/general/concepts.html.

[11] A. Barachant and S. Chevallier. *Mother of all BCI benchmarks*. URL: http://moabb.neurotechx.com/docs/index.html.

[12] *MOABB dataset BNCI2014009*. URL: http://moabb.neurotechx.com/docs/generated/moabb.datasets.BNCI2014009.html (visited on 2022-04-25).

[13] J. Dempster. “Chapter six - signal analysis and measurement”. In: J. Dempster (Ed.). *The Laboratory Computer*. Biological Techniques Series. Academic Press, London, 2001, pp. 136–171. DOI: https://doi.org/10.1016/B978-012209551-1/50039-8.

[14] F. Yger, M. Berar, and F. Lotte. “Riemannian approaches in brain-computer interfaces: a review”. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2017).

[15] M. Congedoa, A. Barachant, and R. Bhatia. “Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review”. *Brain-Computer Interface* (2017). DOI: 0.1080/2326263X.2017.1297192.

[16] *Riemannian manifold*. URL: https://en.wikipedia.org/wiki/Riemannian_manifold.

[17] *ERP covariances - pyriemann*. URL: https://pyriemann.readthedocs.io/en/latest/generated/pyriemann.estimation.ERPCovariances.html (visited on 2022-04-25).

[18] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert. "Xdawn algorithm to enhance evoked potentials: application to brain–computer interface". *IEEE Transactions on Biomedical Engineering* **56**:8 (2009), pp. 2035–2043. DOI: 10.1109/TBME.2009.2012869.

[19] Q. Ai, Q. Liu, W. Meng, and S. Q. Xie. "Chapter 6 - EEG-based brain intention recognition". In: Q. Ai et al. (Eds.). *Advanced Rehabilitative Technology*. Academic Press, 2018, pp. 135–166. ISBN: 978-0-12-814597-5. DOI: https://doi.org/10.1016/B978-0-12-814597-5.00006-0.

[20] *Common spatial pattern*. URL: https://en.wikipedia.org/wiki/Common_spatial_pattern.

[21] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang. "Filter bank common spatial pattern algorithm on BCI competition IV Datasets 2a and 2b". *Frontiers in Neuroscience* **6** (2012). ISSN: 1662-453X. DOI: 10.3389/fnins.2012.00039.

[22] A. Nouri, Z. Ghanbari, M. R. Aslani, and M. H. Moradi. "A new approach to feature extraction in MI-based BCI systems". In: V. Bajaj et al. (Eds.). *Artificial Intelligence-Based Brain-Computer Interface*. Academic Press, 2022, pp. 75–98. ISBN: 978-0-323-91197-9. DOI: https://doi.org/10.1016/B978-0-323-91197-9.00002-3.

[23] E. Neto, F. Biessmann, H. Aurlien, H. Nordby, and T. Eichele. "Regularized linear discriminant analysis of EEG features in dementia patients". *Frontiers in Aging Neuroscience* **8** (2016). ISSN: 1663-4365. DOI: 10.3389/fnagi.2016.00273.

[24] M. Congedo, P. L. C. Rodrigues, and C. Jutten. "The Riemannian minimum distance to means field classifier". *BCI 2019 - 8th International Brain-Computer Interface Conference* (2019).