Julia Brandstetter, BSc

Online Adaptation of a Brain Computer Interface using semi-supervised learning techniques

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieurin

Master's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr. techn. Gernot Müller-Putz

Institute of Neural Engineering

Co-Supervisor
Dipl.-Ing. Andreas Schwarz

Graz, May 2017

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Acknowledgements

I would like to thank everyone who contributed to the successful conduction of this thesis. I am especially grateful to Andreas Schwarz, who guided me through all ups and downs of my work and was always a great support. Furthermore, credit belongs to Gernot Müller-Putz for giving me the possibility to work in this field and supervising my research.
To my family and friends, who encouraged me to give my best and kept my motivation high. Lastly, I want to thank Matiss Reinfelds for love and support over any distance at all times.

Abstract

A sensorimotor rhythm (SMR) controlled brain-computer interface (BCI) is a communication system that allows severely impaired users to control interactions with their environment through motor imagery (MI), e.g. imagination of hand or feet movement. In co-adaptive BCIs, a closed-loop learning process for control is established: users learn from the feedback of the system while the system recurrently adapts to the user's specific brain signals. Present-day MI based BCIs usually guide the user's interaction with the system in a laboratory environment. In real-life BCI applications, however, the system cannot know the user's intent beforehand and therefore no true labels of the intended action are available for classification. The aim of this work was to find options for implementing semi-supervised learning, learning with both true-labeled and unlabeled data, in an adaptive BCI system. The hypothesis that such a semi-supervised learning system works as good as a standard supervised BCI should be verified.
In first offline simulations, filterbank CSP filters and both an sLDA and a Random Forest classifier were employed in an adaptive two-class BCI scenario updating itself every time five new trials per class were available. Seven different approaches to create a training data pool for semi-supervised learning were tested. Only trials that exhibit a high classification certainty should be included for the retraining of the classifier.
The most suitable method was thereafter selected and implemented in an online two-class BCI system, using fbCSP to maximize class discriminability, a statistical outlier rejection to exclude artefact-contaminated trials and an sLDA for classification. A separate optimizer instance recurrently calculated updates of the filters and classifiers.
In a supporting comparative study, two groups of ten subjects each were recorded in a three-hour measurement including three breaks. One group was measured using a standard supervised adaptive system while the other group was used to test the semi-supervised approach.
Results of the study show that there is no statistically significant difference in performance between the two groups. The hypothesis is thus accepted.
In future, the promising results of this work can be used to further establish semi-supervised learning for BCI systems both in the lab and, more importantly, outside of it.

key words: BCI, adaptive, semi-supervised learning, LDA, RF

Kurzfassung

Ein sensormotorisch-rhythmisch kontrolliertes Brain-Computer Interface (BCI) ist ein Kommunikationssystem, das es schwer beeinträchtigten Personen ermöglicht, Interaktionen mit ihrer Umgebung durch Vorstellung von verschiedenen Bewegungen, z.B. Hand- oder Fußbewegungen, zu kontrollieren. In co-adaptiven BCIs wird ein geschlossener Lernkreislauf geschaffen: Benutzer lernen vom System durch das erhaltene Feedback, während das System sich wiederholt an die spezifischen Gehirnsignale des Benutzers anpasst. Heutige SMR basierte BCIs leiten den Benutzer normalerweise in einer Laborumgebung durch seine Interaktionen mit dem System. In realen BCI-Applikationen kennt das System jedoch die Absicht des Benutzers nicht im Vorhinein, daher ist keine richtige Kennzeichnung der beabsichtigten Aktion für die Klassifizierung verfügbar. Das Ziel dieser Arbeit war es, Möglichkeiten zur Implementierung von semi-supervised learning, also Lernen mit sowohl gekennzeichneten als auch ungekennzeichneten Daten, in einem adaptiven BCI-System zu finden. Die Hypothese, dass ein semi-supervised Lernsystem gleich gut wie ein gewöhnliches supervised BCI funktioniert, sollte verifiziert werden.
In ersten Offline-Simulationen wurden Filterbank CSP-Filter und sowohl ein sLDA- als auch ein Random Forest-Klassifikationsalgorithmus in einem adaptiven zwei Klassen BCI-Szenario verwendet, das immer wenn fünf neue Trials pro Klasse verfügbar waren upgedatet wurde. Sieben verschiedene Ansätze, um einen Trainingsdatenpool für semi-supervised learning zu generieren, wurden getestet. Nur Trials mit einer hohen Klassifikationssicherheit sollten für das Update berücksichtigt werden.
Die am besten geeignete Methode wurde danach ausgewählt und in einem online zwei Klassen BCI-System implementiert. Verwendet wurden fbCSP-Filter, um die Klassendiskriminabilität zu maximieren, eine auf statistischen Parametern beruhende Artefakt-Erkennung, um Trials mit Artefakten auszuschließen, sowie eine sLDA zur Klassifikation. Eine separate Optimierungsinstanz berechnete wiederholt Updates für die Filter und den Klassifizierer.
In einer begleitenden Vergleichsstudie wurden zwei Gruppen mit je zehn Teilnehmern in einem dreistündigen Experiment mit drei Pausen aufgenommen. Eine Gruppe wurde mit einem supervised adaptiven Standardsystem gemessen, die andere testete den semi-supervised-Ansatz.
Die Ergebnisse zeigen, dass es keine statistisch signifikanten Unterschiede in der Leistung zwischen den beiden Gruppen gibt. Die Hypothese ist damit akzeptiert.
In Zukunft können die vielversprechenden Resultate dieser Arbeit verwendet werden, um semi-supervised learning weiter für BCI Systeme im Labor und, noch wichtiger, außerhalb des Labors zu verankern.

Schlüsselwörter: BCI, adaptiv, semi-supervised learning, LDA, RF

Contents

1 Introduction

A brain-computer interface (BCI) is defined as a communication system that provides a user with a new output channel independent from the usual nervous and muscular paths. Many different diseases as well as trauma can impair a person's neural pathways and motor functions and a BCI can be a means to restore these functions. It allows users to interact with their surroundings only by thought or by attention to certain stimuli through the use of brain signals [1]. The beginnings of BCI research go as far back as 1973 [2], and since then, a large number of different methods and approaches to measure and classify brain function has evolved.

There are generally two different ways of measuring brain activity - invasive and noninvasive approaches. In the invasive approach, patients are subjected to brain surgery and electrodes are placed directly on the cortex. Different methods such as recording the electrocorticogram (ECoG) on the cortex [3] or placing electrode arrays within the cortex [4] have been researched. A variety of possibilities for obtaining brain signals noninvasively is available of which the electroencephalogram (EEG) is the most wide-spread for BCIs.
The EEG records cortical potentials of the brain by placing electrodes on the scalp. Neurons in the cortex of the human brain generate action potentials that in turn result in postsynaptic potentials (PSP). The potential differences of the PSP of large groups of neurons are measured in the EEG, the resulting signals ranging in the microvolt scale [5]. Among these potentials, rhythmic oscillations can be found in different frequency bands [6]. There are several advantages of using the EEG for BCIs: Recording the EEG is noninvasive, which means that no dangerous and costly surgery is necessary. Electrodes are placed on the scalp and the system is ready in a short timespan. In addition, the EEG provides a good temporal resolution in the range of milliseconds, allowing for fast reaction of the system. Moreover, it is an comparatively inexpensive method.
However, the major disadvantage of the EEG is the bad spatial resolution it exhibits, because one electrode records the activity of millions of neurons. Furthermore, the skull and tissue between the cortex and the electrodes attenuate the signals considerably, resulting in a poor signal to noise ratio [7].

To be able to control a BCI with EEG signals, voluntary changes and modulation of the recorded potentials must be possible. After an internal or external stimulus, frequency specific changes of the brain activity occur. This leads to a power decrease or increase in distinct frequency bands of the EEG, such as the μ (8-13 Hz) or the β band (13-30 Hz). A relative power decrease within a specific frequency band of the signal stems from a desynchronisation of the activity of underlying neural tissue during e.g. a movement or preparation for a movement. Desynchronisation is therefore associated with action. Similarly, a power increase happens due to a synchronisation of neural activity after the movement and is associated with rest. These phenomena are called event-related desynchronisation/synchronisation (ERD/ERS) [8]. Depending on the nature of the activity, different areas of the brain are stimulated. EEG patterns of sensorimotor areas are connected, amongst others, to movement and can be used as control signals in BCIs. Interestingly, these ERD/ERS patterns appear not only during actual movement, but also

when the movement is only imagined [9]. Such motor imageries (MIs) cause ERD/ERS activity in different areas of the sensorimotor cortex, depending on the type of imagined movement, e.g. right or left hand or foot movement (see Figure1.1) [10].

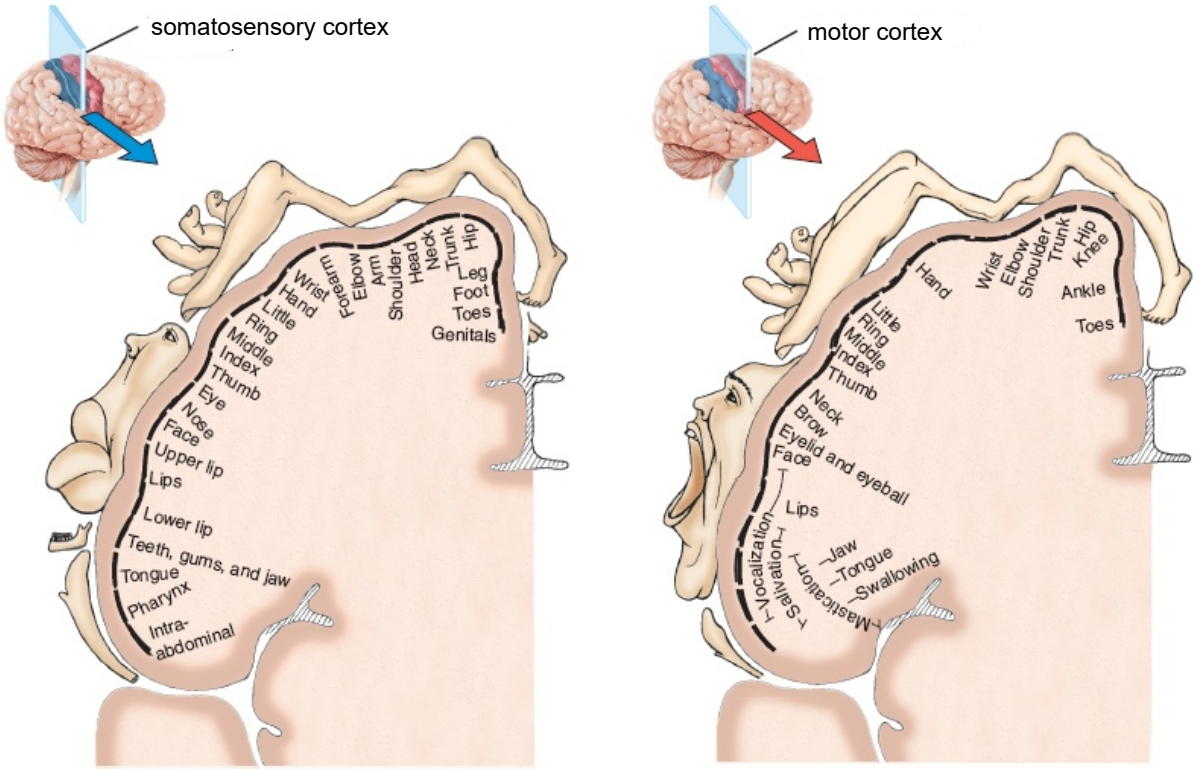


Figure 1.1: Areas of the somatosensory (a) and motor (b) cortex with respect to their represented body parts, modified from [11].

ERD/ERS brain patterns are time-locked to an event. They can be quantified in an established procedure starting with bandpass filtering in the respective frequency range and subsequent squaring and averaging of samples to retrieve the signal power. Afterwards, the averaged reference signal power before the event is subtracted from the result and divided to obtain the relative changes in brain activity [8] [12].

A standard BCI system consists of several components, as shown in Figure 1.2. Ideally, the BCI system is a closed-loop system, which provides fast feedback to the user's input.
The first step is signal acquisition which involves measuring the subject's brain activity invasively or noninvasively. The recorded signal can be classified as evoked by some external stimuli or spontaneous. In a standard EEG-based BCI, the input is recorded by electrodes on the subject's scalp before being amplified and digitised using appropriate hardware. After signal acquisition, the recorded activity is processed in several signal processing steps. The preprocessing block consists of various techniques applied to ameliorate the subsequent feature extraction and classification process. Amplification, filtering and noise and artefact removal procedures are employed to optimally prepare the signal for further use. The feature extraction block aims to retrieve features that best describe the properties of the brain signals and encode the user's command. This is important to further simplify the process of classification. The goal is to maximize the difference between the separate task-related patterns, e.g. motor imageries. Here, a common approach is to use bandpower features in the frequency ranges of interest, e.g. μ and β bands. Numerous other techniques are, however, available. In the next step, a classifier assigns a class

(e.g. a certain motor imagery task) to the previously computed features of brain activity to distinguish between the different MI tasks. Training the classifier with individual training data of the user is an important step in order to achieve a good classification performance. Various classification algorithms with different properties have been adapted for BCIs, such as Linear Discriminant Analysis (LDA) [13] or, more recently, Random Forests (RF) [14]. The classifier's output is transformed into a control signal and consequently operates the application interface. To close the loop of the BCI, feedback is generated by the application interface and observed by the user [1] [7] [12] [15] [16].

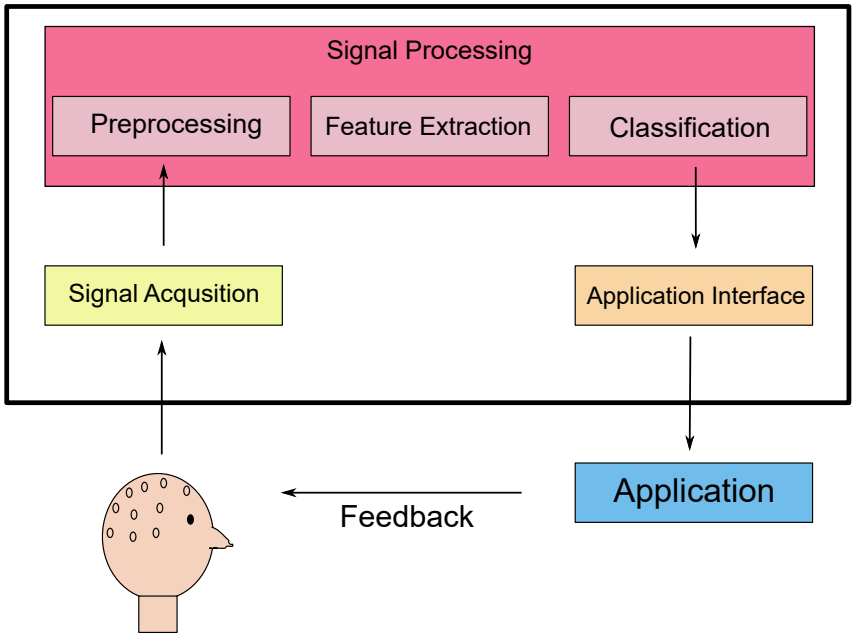


Figure 1.2: Standard model of a BCI system.

Current BCI systems generally use this framework (see Figure 1.2), although every system has its own modifications and adaptations to serve a certain purpose. Often the system needs a relatively long training period during which the user receives no feedback about his input. This can have a considerable negative effect on the user's motivation and concentration. Furthermore, the EEG signal has been shown to be non-stationary [17, 18]. There are changes in the brain patterns both between the calibration and feedback phases in a standard BCI experiment as well as throughout the experiment due to the user's motivation and fatigue. The BCI system's ability to adapt to these modifications is important for the user's performance [17].
First work in researching reliable co-adaptive BCIs has been done by Vidaurre et al. ([19] and [20]). In their experiment, they provided feedback for the user throughout the whole session, starting with a standard classifier trained on data of a large number of subjects, and subsequently adapting the system to the specific user.
Faller et al. [21] later implemented a different adaptive approach for training the feature extraction and classification blocks of an MI-based BCI. The goal was to reduce the training period without feedback significantly. The system consists of a standard BCI system with an LDA classifier and an "Optimizer Instance". The optimizer recurrently receives recorded EEG data from the system and uses this data to retrain the system's classifier. Instead of using a classifier trained on many subjects' data for first feedback, a small initial amount of trials from the specific user were used for first training the system. The updated classifier is then applied on subsequent trials, before the next update takes place. Thus, the subject's training period without feedback is shortened to only a few minutes. Results show a mean peak accuracy of over 75 % for 12

users.
Schwarz et al. [22] introduced a co-adaptive BCI system, based on the same ideas as Faller et al [21], but employing a Filterbank Common Spatial Pattern (CSP) for feature extraction and a Random Forest classifier. Again, recorded training data was sent to the optimizer to retrain classification and adapt the system to the current input of the user. Therefore, feedback could be provided to the subject within minutes. A supporting study with 13 participants yielded an overall peak accuracy of 88.6 %, outperforming the results of Faller et al.[21] significantly.

Semi-supervised learning is a special kind of machine learning algorithm for cases where the amount of data with true class labels is small. It is a combination of supervised learning, which is currently in use for standard BCI systems, and unsupervised learning. Data is present in two different sets. One set consists of observations whose true class labels are known while the other set's observations are unlabeled. In a self-training algorithm, the classifier is first trained only on the labeled data set and then used to label the first part of the unlabeled data. The classifier is thereafter recurrently retrained on both the labeled and unlabeled data sets, using its own predictions as labels for the latter [23] [24].
Several approaches to employ semi-supervised learning in BCI systems have been proposed, e.g. using semi-supervised learning with a Support Vector Machine classifier in a BCI speller system [25], an adaptive MI-based BCI with a semi-supervised Naive Bayesian Parzen Window classifier [26] or a Spectral Regression Kernel Discriminant Analysis classifier [27].

1.1 Motivation and Aim

Present-day standard BCI systems are usually tested in a laboratory environment and based on supervised learning algorithms. In a standard paradigm, the user is guided through the session, performing specific tasks at specific times. System calibration and classification is performed using the true class labels of the subject's input that are known due to the nature of the experiment. On the contrary, the system cannot know the user's intent beforehand in a real-world application of a BCI and therefore, no true labels would be available for the supervised learning of the classifier.
The idea of this work was to combine a state-of-the-art adaptive BCI system with the concept of semi-supervised learning. After a short initial supervised training period, only the output of the classifier should be used as labels for the data in further retrainings of the system.
In a first step, seven different approaches based on two classifiers (LDA and RF) were proposed. The aim of these approaches was to filter trials that exhibit a high certainty of correct classification and should therefore be used for the system's retrainings. These methods were tested offline on data of Schwarz et al. [22] using an adaptive BCI scenario. After analysing and comparing the results, the most promising approach was selected and implemented in an online adaptive BCI system similar to Schwarz et al. [22]. It was evaluated in a comparative online study with two groups of ten able-bodied participants each. One group was recorded using the standard supervised system, the other one was used to test the semi-supervised approach.
The hypothesis is that the system employing a semi-supervised approach works as good as the supervised adaptive BCI system.
Furthermore, the online experiment for both groups was designed to be comparatively long, including three breaks of 20 minutes each. Apart from performance differences between the two groups, we wanted to test possible fatigue among all subjects.
Chapter 2 of this thesis gives an overview of the concept design of this work as well as detailed information about the applied technologies and the comparative study. In chapter 3, findings

of the offline analysis as well as results of the online study are presented. Chapter 4 discusses these results in various aspects and points out possible topics for further investigation. Section 5 concludes the thesis. In the Appendix, important parts of the computational framework and ERD/ERS maps of all subjects are displayed.

2 Methods

2.1 Concept Design

The goal of the first part of this thesis was to find a suitable method that filters trials with high classification certainty. For this purpose, seven different methods for the creation of a data pool with reliable classification predictions that could be used in semi-supervised learning of a BCI system were contrived and subsequently investigated. An offline adaptive BCI scenario with two classes was implemented for testing purposes. It includes a filterbank CSP (FBCSP) filter before the extraction of logarithmic bandpower features and both a Linear Discriminant Analysis and a Random Forest classifier to try out these different approaches. EEG data of Schwarz et al. [22] was used for offline testing. This offline BCI scenario uses the same basic framework as the later developed online system. Figure 2.1 shows this structure; in the case of the offline analysis only the feedback part is naturally not implemented.
After a performance analysis of the offline results, the most suitable semi-supervised method to work with was selected and implemented in an online system for the second part of this work. The two-class online BCI system uses motor imagery of the right hand and both feet as control signals, derived from multichannel EEG signals. Again, an FBCSP was employed to increase the discriminability of the afterwards calculated logarithmic bandpower features. The features are thereafter classified by a shrinkage LDA (sLDA) classifier. The classifier model is repeatedly updated in a separate optimizer instance to adapt to the user's input over time. For the update, recorded data that passes the semi-supervised exclusion criterion is taken into account with its predicted labels. Additionally, the classifier output is used for the calculation of feedback for the user, to keep motivation and focus on a high level. A schematic overview of the online BCI system is shown in Fig. 2.1.

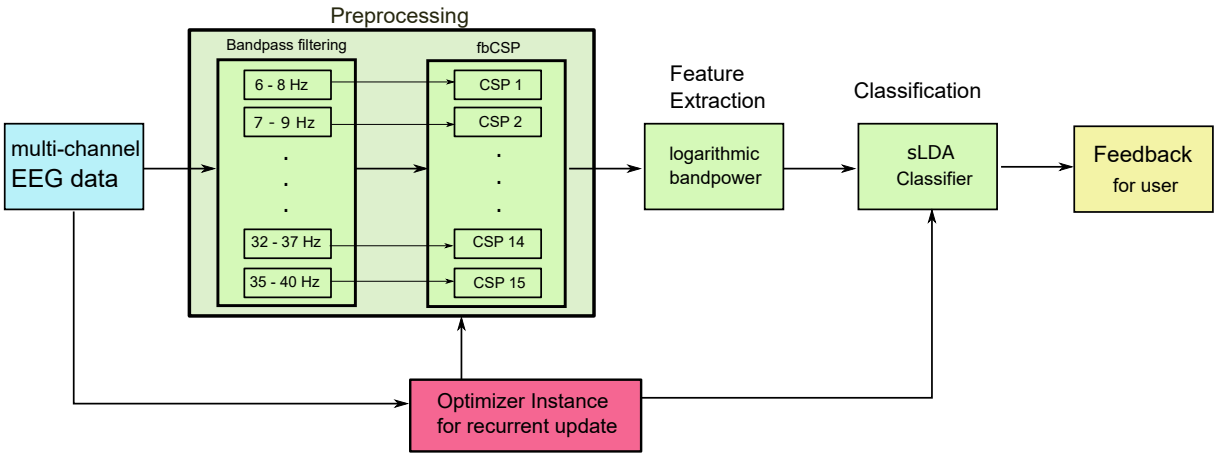


Figure 2.1: Schematic overview of the system's structure for both offline analyses and online execution.

In the supporting study for the proposed system, two separate groups of subjects were measured. For one group, the classifier update was calculated using all available data with their true class labels (supervised learning), while the other group's classifiers were updated with first a baseline of data using true labels and apart from that only with data that passed the exclusion criterion for semi-supervised learning. For this second part of the training data, the predictions of the classifier output were used as class labels.

2.2 Preprocessing and Feature Extraction

2.2.1 Common Spatial Pattern

In order to improve the results of classification, preprocessing of the signal is performed in all standard BCI systems. The raw EEG data suffers from a poor spatial resolution because of volume conduction of the scalp. It is therefore imperative to prepare the signals before classification in a way that the greatest possible class discrimination is achieved [28]. Features which are most effective in preserving class separability should be chosen. This is done by searching for the data subspace that best retains class separability in the lowest dimensional space [29].
The idea to use the common spatial pattern method for filtering of EEG data was first introduced by Ramoser et al. [28]. The goal is to maximize the signal variance under one condition and simultaneously minimize it for the other one. Basically, multi-channel EEG data is linearly transformed into a low-dimensional spatial subspace using a projection matrix. Each row of this matrix consists of weights for the individual EEG channels. Thereby, the variance of two-class signal matrices can be maximized. Both covariance matrices of the two classes are simultaneously diagonalized [30].
In a more detailed description, $\mathbf{X}_H$ and $\mathbf{X}_F$ can be seen as the matrices of EEG data for two different classes (hand and foot motor imagery). Each matrix has the dimensions $N \times T$, where N is the number of channels and T is the number of samples per channel. Then the two normalized covariance matrices of the EEG data can be calculated as

$$R_H = \frac{X_H X_H^T}{trace(X_H X_H^T)} \tag{2.1}$$

and

$$R_F = \frac{X_F X_F^T}{trace(X_F X_F^T)}. \tag{2.2}$$

$\mathbf{X}^T$ stands for the transpose of the data matrix $\mathbf{X}$ and trace() is the sum of the diagonal elements of a matrix.
In a next step, these covariance matrices are averaged over all trials of their class to receive the averaged covariance matrices $\overline{R_F}$ and $\overline{R_H}$. By summing the two, they can be factorized as follows:

$$R = \overline{R_F} + \overline{R_H} = U_0 \Sigma U_0^T, \tag{2.3}$$

where $\mathbf{U}_0$ is the eigenvector matrix and Σ is the diagonal matrix of eigenvalues. A whitening

transformation matrix

$$P = \Sigma^{-\frac{1}{2}} U_0^T \tag{2.4}$$

equalizes the variances in the space extended by $\mathbf{U}_0^T$. With this, the average covariance matrices can be transformed to

$$S_H = P\overline{R_H}P^T \qquad \text{and} \qquad S_F = P\overline{R_F}P^T. \tag{2.5}$$

The matrices $\mathbf{S}_H$ and $\mathbf{S}_F$ now have shared eigenvectors and the sum of their eigenvalues is always one, as shown in 2.6.

$$S_H = U\Sigma_H U^T, \qquad S_F = U\Sigma_F U^T, \qquad \Sigma_H + \Sigma_F = 1. \tag{2.6}$$

It now becomes apparent that the eigenvectors with the largest eigenvalues for $\mathbf{S}_F$ have the smallest eigenvalues for $\mathbf{S}_H$ and vice versa. This property makes the eigenvectors $\mathbf{U}$ useful for maximizing discriminability of two distributions. When projecting the transformed EEG data onto the first and last eigenvectors of $\mathbf{U}$, i.e. the largest eigenvalues for each class, optimal feature vectors for the discrimination of two EEG classes can be obtained. This projection matrix is given by

$$W = U^T P. \tag{2.7}$$

Using $\mathbf{W}$, the EEG data can be decomposed into uncorrelated components:

$$Z = WX. \tag{2.8}$$

By determining the inverse matrix of $\mathbf{W}$, $\mathbf{W}^{-1}$, the original EEG data $\mathbf{X}$ can be reconstructed from $\mathbf{Z}$, which is a matrix of common and specific EEG source elements for the two different tasks. The columns of $\mathbf{W}^{-1}$ are the common spatial patterns and can be considered as EEG source distribution vectors. The first and last columns are most important since they explain the largest variance of one and the smallest variance of the other class [28] [30] [31].

2.2.2 Filterbank CSP

The performance of the common spatial pattern method greatly depends on the frequency range of the selected EEG data. Selecting an inappropriate scope of frequencies for the calculation of CSP features leads to low resulting accuracies of the system. Often a very broad or manually chosen, subject-specific frequency band is used to ensure functionality of the CSP method. Several algorithms are available for the choice of specific frequency ranges, such as Common Sparse Spectral Spatial Patterns (CSSSP) [32], Common Spatio-Spectral Pattern (CSSP) [33], Subband Common Spatial Pattern (SBCSP) [34] or Filterbank Common Spatial Pattern (FBCSP) [35].
In this work, the filterbank CSP approach was applied. First, the EEG signal is bandpass filtered into certain frequency bands in the frequency ranges of interest using IIR filters. Next, each of these bands is used to calculate optimized signal projections with the common spatial pattern algorithm [35].

In this work, a filterbank of 15 bandpasses was used to filter EEG data in the areas most interesting for motor imagery - especially μ-bands. In Table 2.1, the upper and lower cutoff frequency for each of the 15 filters can be seen. The butterworth IIR bandpass filters were of fourth order.

Table 2.1: Cutoff frequencies for the 15 filterbank bandpasses.

bandpass	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lower cut-off [Hz]	6	7	8	9	10	11	12	14	17	20	23	26	29	32	35
higher cut-off [Hz]	8	9	10	11	12	13	14	19	22	25	28	31	34	37	40

Each bandpass filtered signal was used to create patterns with the CSP algorithm. Of these, only the six most discriminative common spatial patterns, in accordance with Schwarz et al. [22], were taken, which corresponded to the first and last three columns of the matrix $\mathbf{W}^{-1}$, thus resulting in a total number of 15 filtered signals $\times$ 6 columns = 90 features for further classification.

2.2.3 Logarithmic Bandpower Features

In the last step before classification, features have to be extracted from the CSP-filtered signals. Changes in signal power resulting from variations in signal amplitude and frequency, which for their part stem from brain activation due to the motor imagery task, are commonly used. To calculate signal power, the recorded data samples are squared and a moving average filter over one second is applied. In a final step, the calculated band power is logarithmized and the resulting features fed into the classifier.

2.3 Classification and Classifier Updates

For the actual classification of the extracted features, two different methods are used in the offline analyses of this work. These classification algorithms are the Linear Discriminant Analysis and the Random Forest. For the experimental part in the online system, only the LDA classification is operated.

2.3.1 Accumulated Training Data vs. Sliding Window Algorithm

In the offline adaptive BCI scenario for testing the semi-supervised approaches, training data for the classifier update is gathered in two different ways. In the first one, all available data that has accumulated up to the time of an update is taken as possible training data for the classifier that has to be filtered by the semi-supervised exclusion criterion. Consequently, the amount of training data increases over time. For the second option, a sliding window algorithm is implemented. Only a certain constant amount of the available data is considered for the classifier update by sliding a window over all available trials, adding the newly accumulated ones while at the same time discarding the oldest ones.
In the online system, all available data is taken into account for updating the classifier (accumulative algorithm).

2.3.2 (shrinkage) LDA

Linear Discriminant Analysis is a method of linear classification, which means that the decision boundary between the regions of different classes is a linear function. In the two class case, it is defined by a one-dimensional hyperplane.
A discriminant function is used to directly assign every input to a class. The simplest linear discriminant function is

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0, \tag{2.9}$$

where $\mathbf{w}$ is the so-called weight vector and w_0 is a bias.
In the case that the discriminant function $\mathrm{y}(\mathbf{x}) \geq 0$, the input sample is assigned to class 1, otherwise to class 2. The decision boundary is located at $\mathrm{y}(\mathbf{x}) = 0$.
The vector $\mathbf{w}$ is orthogonal to every vector within the decision surface, and determines the orientation of the surface. The normal distance from the origin to the decision surface is

$$\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||} = \frac{w_0}{||\mathbf{w}||}. \tag{2.10}$$

It is evident that the bias determines the location of the decision surface.

$$r = \frac{y(\mathbf{x})}{||\mathbf{w}||}. \tag{2.11}$$

gives a signed measure of the distance r from the point x to the decision surface.

In a two class problem, the input vector is projected down to one dimension with

$$y = \mathbf{w}^T\mathbf{x}. \tag{2.12}$$

Because of the considerable loss of information occurring with the down-projection, the data of the two classes is now likely to overlap. It is therefore necessary to adjust the components of the weight vector to select a projection that maximizes class separation.
This process starts with the calculation of the mean vectors of the two classes using

$$\mathbf{m}_1 = \frac{1}{N}\sum_{n\in C_1}\mathbf{x}_n \qquad \text{and} \qquad \mathbf{m}_2 = \frac{1}{N}\sum_{n\in C_2}\mathbf{x}_n. \tag{2.13}$$

N_1 is the number of points of class C_1, N_2 the number of point of class C_2.
The easiest method of class separation is the separation of these projected means, so the weight vector w is chosen in a way that maximizes

$$m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1). \tag{2.14}$$

$$m_k = \mathbf{w}^T\mathbf{m}_k \tag{2.15}$$

is the mean of the projected data of each class C_k.
Additionally, $\mathbf{w}$ has to be constrained to have unit length so that this measure will not get very large.

However, there is still the problem of overlapping because of strongly non-diagonal covariances of the class distributions. Therefore, a function that maximizes the between-class variance while at the same time minimizing the within-class variance has to be found. The within-class variance is calculated as

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2, \tag{2.16}$$

where $y_n = \mathbf{w}^T \mathbf{x}_n$, for each class.
The total within-class variance for the whole data set is then

$$s_1^2 + s_2^2. \tag{2.17}$$

The ratio of between-class variance and within-class variance is defined as the Fisher criterion:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)}{(s_1^2 - s_2^2)}. \tag{2.18}$$

It can also be written in a form to highlight the dependence on $\mathbf{w}$ as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}, \tag{2.19}$$

where $\mathbf{S}_B$ is the between-class covariance matrix, given by

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T, \tag{2.20}$$

and $\mathbf{S}_W$ is the within-class covariance matrix, given by

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T. \tag{2.21}$$

Through differentiation with respect to $\mathbf{w}$ it can be shown that J($\mathbf{w}$) is maximized when

$$(\mathbf{w}^T \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} = (\mathbf{w}^T \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w}. \tag{2.22}$$

Here, scalar factors can be dropped, since only the direction of $\mathbf{w}$ is important. Consequently,

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1). \tag{2.23}$$

If the within-class covariance matrix is isotropic, $\mathbf{w}$ is proportional to the difference of the means.
In Fisher's linear discriminant, the projected data (projected onto one dimension) is used to find

a discriminant function, constructing a threshold y_0 [13] [24] [36].
Linear Discriminant Analysis is the optimal classification method for Gaussian distributions with the same covariance matrix for all classes. However, this is only true if following three assumptions are fulfilled: 1) The features of each class are Gaussian distributed. This is satisfied for logarithmic bandpower features. 2) The Gaussian distributions of all classes have the same covariance matrix, which means that the data is linearly separable. Even when this criterion is not met, the LDA is still quite robust. 3) The true class distributions are known. Obviously, the third assumption is never fulfilled in real life and leads to particularly great errors if the amount of training data is small compared to the dimensionality of the features.
As described above, a classification function assigns a class label to a given input based on $\mathrm{y}(\mathbf{x})$. A linear classifier trained on spatial features can be seen as a spatial filter. A spatial filter is used to determine a source from a signal. Spatial filters depend not only on the scalp distributions of the source, but also on the distributions of the other sources and, additionally, noise.
In this sense, a linear classifier can be interpreted as a backward model to recover a signal from sources. The result is that every EEG channel is weighted by a weight vector and summed up [13] [24] [37].

The empirical covariance is normally used to estimate the covariance matrix in the calculation of weights for the LDA, due to its good and unbiased properties. However, in cases of high data dimensionality and few data points, these estimates can get too imprecise. Large eigenvalues are estimated too large and small ones too small, which leads to a systematic error and a decline of classification performance. Therefore, a form of regularisation called shrinkage can be applied. A tuning parameter γ is introduced to counteract the estimation error, so that the covariance matrix is defined as

$$\tilde{\mathbf{S}} = (1-\gamma)\mathbf{S} + \gamma\mu\mathbf{I}, \tag{2.24}$$

where μ is the average eigenvalue trace.
Since $\mathbf{S}$ is positive semidefinite, an eigenvalue decomposition can be carried out:

$$\mathbf{S} = \mathbf{V}\mathbf{D}\mathbf{V}^T. \tag{2.25}$$

Applying this formula to the former calculation of $\tilde{\mathbf{S}}$, the result is

$$\tilde{\mathbf{S}} = (1-\gamma)\mathbf{V}\mathbf{D}\mathbf{V}^T + \gamma\mu\mathbf{I} = \mathbf{V}((1-\gamma)\mathbf{D} + \gamma\mu\mathbf{I})\mathbf{V}^T. \tag{2.26}$$

$\mathbf{S}$ and $\tilde{\mathbf{S}}$ have the same eigenvectors (the columns of $\mathbf{V}$). It is apparent that large eigenvalues are shrunk and small ones elongated depending on γ, so they fit better to the average. If γ is zero, the formula yields the unregularised LDA, whereas with $\gamma = 1$, the covariance matrix is spherical. For features in BCI studies, this corresponds to controlling the weight vector between the two extremes of a spatial filter with the spatial structure of the noise and a spatial pattern. In case the covariance of the noise is known, the unregularised version brings the best result. The optimal value for the parameter γ can be calculated analytically [37].

2.3.3 Random Forest

A random forest classifier is a classification method introduced by Leo Breiman [14]. It is defined as a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x},\Theta_k)\}$, where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree gives a unit vote for the most popular class at input $\mathbf{x}$ [14].
Random forests are based on decision trees and a randomisation technique called bagging. They are a combination of tree predictors, in a way that each tree is only dependant on the values of a random vector sampled independently and with the same distribution for all trees in the forest.
With the use of decision trees, the feature space can be divided into a set of regions and a simple model can be fitted into each of those regions. The splitting of regions continues until some stop condition is fulfilled. At the top of the tree, there is the full dataset. Features that fulfill the condition of the first junction are assigned to the left branch, the others to the right one. For every branch, additional conditions can be introduced and the features will be split further. Different criteria such as the misclassification error or the Gini index can be used to determine node impurity and hence decide if a region should be split further or not. In the end, every terminal leaf represents one region (in our case this means a set of training data of the same class).
The proportion of observations of a class k in a node m, which represents a region R_m with N_m observations, is given by

$$\hat{p} = \frac{1}{N_M} \sum_{x_i \in R_m} I(y_i = k). \tag{2.27}$$

The observations in node m are classified to the majority class of this node.
Bagging, which stands for bootstrap aggregation, is a way to reduce the variance of an estimated prediction function. It works particularly well for procedures with high variance and low bias. The idea is to average over many models with noise that are unbiased and such decrease variance. Random forests employ a modification of bagging by constructing a large ensemble of decorrelated trees and then averaging them.
Decision trees are known to be relatively unbiased if they are grown deep enough, but contain a lot of noise and can thus be improved through averaging. Since every decision tree in bagging is identically distributed, the average expectation of a collection of trees is the same as the expectation of each one of the trees. This means that also the bias of the tree ensemble is the same as the bias of the individual trees. A performance improvement of the classifier can therefore only be realised through decreasing of variance. The variance of the average of B identically distributed variables is given by

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2, \tag{2.28}$$

where ρ is the positive pairwise correlation between the variables and σ^2 is the variance of each variable. As can be seen in 2.28, increasing B decreases the second term of the equation. The first term remains, resulting in the size of the pairwise correlation of the bagged trees being the limiting factor for the reduction of variance through averaging.
Decision trees are already typically quite different from each other even when they are constructed with similar training sets. A small variability in the data set results in a great difference in the prediction models. However, the correlation between trees can be further decreased. The concept in random forests is to reduce the correlation between the trees to improve the

variance reduction of bagging without increasing variance too much. This can be accomplished by random selection of the input variables in the tree growing process. Before each split during the growth of the tree, $m \leq p$ of the input variables are randomly selected for splitting, where m is typically $\sqrt{p}$. After a number of B such trees $\{T(x;\Theta_b)\}_1^B$, every tree gives a classification prediction. The classification output is obtained through a majority vote of the individual tree predictions in the random forest. Reducing m will also reduce the correlation between trees and subsequently improve variance of the average according to 2.28.

The random forest algorithm can be outlined as following:

1. for every tree $b = 1$ to B:
   (a) draw a bootstrap sample of size N from the training data. (bootstrapping)
   (b) grow a random forest tree T_b to the bootstrapped data, by recursively carrying out these steps for each terminal node of the tree until the minimum node size is reached:
      i. select m variables at random from the p variables available.
      ii. pick the best variable/split-point among the m.
      iii. split the node into two daughter nodes.
2. output the collection of trees $\{T_b\}_1^B$.

To make a classification prediction at a point x, let $C_b(x)$ be the class prediction of the b-th random forest tree. Then, $C_{rf}^B(x) = \text{majority vote}\{C_b(x)\}_1^B$ [14] [24] [36] [38].
There are two parameters of random forests that have a great influence on their performance: the number of trees that are to be constructed and the number of randomly chosen data dimensions in every node [14]. It is necessary to build a sufficient number of trees to reduce the average variance, whereas a large number of trees also increases the computational effort of the classification. For this trade-off, Steyrl et al. [38] have carried out extensive tests on BCI data sets and concluded that a number of 1000 tree classifiers and $\sqrt{\#\text{number of features}}$ should be used. These recommendations were followed in the offline data analysis of this work.

2.4 Artefact Removal - Outlier Rejection

Artefacts are unwanted potentials in EEG data that can modify the neurological control signal of a BCI. It is even possible that they act as the unwanted BCI control signal themselves instead of the desired brain pattern. Artefacts can be separated into two groups: physiological and nonphysiological artefacts. Nonphysiological artefacts stem from sources in the hardware or surroundings, e.g. 50 Hz power line noise, impedance changes of the electrodes or quantification and amplifier noise. They are normally well controlled by technical measures such as filters, shielding, etc. Physiological artefacts arise from various sources within the subject itself and include heart activity (electrocardiogram - ECG), respiration, eye movement (electrooculogram - EOG) or body movement (electromyogram - EMG).
EOG artefacts involve blinking or eye rolling and appear as high-amplitude sequences in the EEG. EMG artefacts, on the other hand, usually generate bigger disturbances in a large frequency range. They can be caused by e.g. swallowing, teeth grinding or head movements.
Although users are usually instructed not to blink or swallow during trials, it is not possible to completely exclude the occurrence of physiological artefacts. To ensure proper function of

a BCI and prevent artefacts from influencing the classifier, it is necessary to recognize and exclude them [39]. Therefore, a statistical outlier rejection is applied both in offline simulations and online sessions of this work, based on Delorme et al. [40], which was also used by Schwarz et al. [22] and Faller et al. [21].
Before using recorded trials in retrainings of the classifier, they were automatically checked for being outliers. Data was evaluated with four different approaches to detect artefacts:

1. Rejection by amplitude: Trial data is inspected and rejected depending on standard thresholding. If any sample in a trial exceeds a certain threshold value, the whole trial is dismissed. Especially EOG artefacts such as blinking can easily be detected with this method. The threshold was set to ± 100 μV.

2. Rejection by variance: The variance of every channel was calculated and trials were rejected when more than five times the standard variance was exceeded.

3. Rejection by probability: Artefact-afflicted trials often have an unusual course over time and are therefore unexpected. By determining the joint-probability of a trial's values and comparing it to the probability distribution of all trials.

4. Rejection by kurtosis: This is a second method to find unusual data distributions. The kurtosis of all values in a trial is calculated and evaluated. Highly positive or negative results for the kurtosis measure indicate physiological or nonphysiological artefacts, respectively. A threshold is introduced to reject trials with extreme kurtosis values [41].

2.5 Semi-Supervised Learning Algorithms

Semi-supervised machine learning is a mixture of supervised and unsupervised learning. A small part of the data used for training the classifier is labeled with their known class labels (supervised), whereas the other, larger, part of the data does not possess true class labels (unsupervised). For this part of the samples, the classifier's own prediction is taken as their label. In a first classifier training, only data with true labels is used. This classifier then predicts classes for the following data. Both data sets are then taken into account for further trainings of the classifier [23] [24].
The amount of initial trials that were used with their true class labels was set to 35 trials per class (TPC) in the offline testing on data of Schwarz et al. [22], after results for simulations with different amounts of starting trials were evaluated. For convenient application in the online study, 40 TPC were included with their true class labels in classifier training before starting the semi-supervised learning update. Moreover, results of Vidaurre and Blankertz [42] show that around 40 TPC is sufficient for initial calibration.
To ensure high classification performance it is important that only trial data that exhibits a highly reliable classification prediction is taken into account for retraining of the classifier. Therefore, samples with a high uncertainty in the class prediction should be rejected. However, it cannot be completely ruled out that trials with a wrong class label are still included in retraining. The resulting pool of reliable trial data along with an initial amount of trial data with known true labels can then be used to recurrently adapt the classifier. In this work, five methods employing an sLDA classifier and two for an Random Forest (RF) classifier to select data with high certainty classification predictions were developed and tested.

2.5.1 sLDA-based Methods

Boxplot Methods

For the first boxplot method, the class probabilities of the recorded trials are collected and their mean values calculated. Then, the first and third quartile (the 25%- and 75%-boundaries) of these means are determined. Similar to creating a boxplot graph, only trials whose mean class probabilities are within the limits between the first and third quartile are taken into account for further retraining. All other trials whose mean values lie outside these boundaries are rejected. In the second boxplot approach, peak values of class probabilities of the trials are taken into account. As in the first method, first and third quartiles of the peak values of all trials are determined and trials sorted accordingly.
A schematic of both procedures is shown in Figure 2.2.

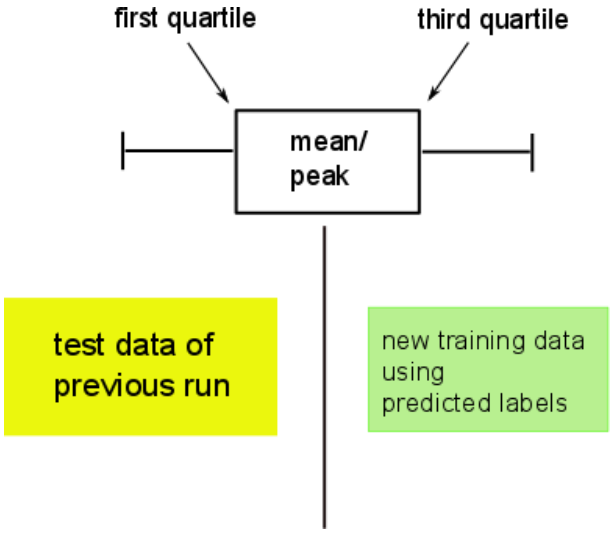


Figure 2.2: Boxplot methods to filter data.

With these techniques, it is intended to filter trials in a way that outliers on both ends, very good as well as rather bad trials, could be rejected and average ones with most useful information kept.

Cumulative Method

In the third technique, a cumulative vote is implemented. Predicted class labels for every sample in the feedback period of the recorded trials are counted and summed up. The class label of the majority of samples in a trial is then assigned to that trial. To sort out unwanted uncertain trials, a threshold is introduced. Only trials whose difference in counts of the both class labels is above this threshold are considered for further retrainings. All other trials are dismissed. For the threshold, several different options were tried out. Finally, a difference between the two class counts of 20% had to be reached for a trial not to be rejected. A schematic graph of this method is shown in Figure 2.3.

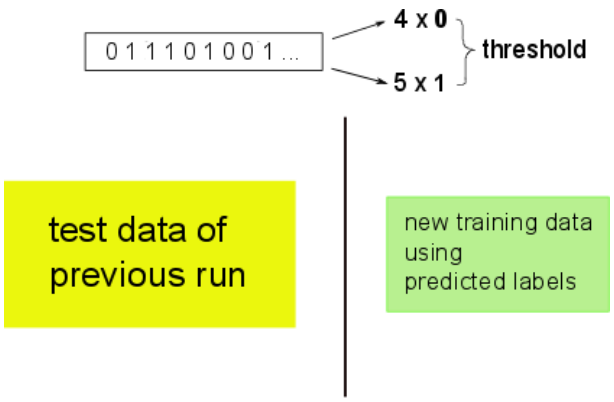


Figure 2.3: Concept of the cumulative method.

Mean and Peak Method

In another approach, the mean values of the class probabilities of the trial data are computed. A threshold is introduced which has to be reached by a trial's mean class probability to be included in further classifier trainings.
Similar to this, in a fifth approach, peak values of the class probabilities are determined and evaluated. Trials whose peaks pass a certain threshold are considered for retraining of the classifier.
Figure 2.4 shows an overview of these methods.

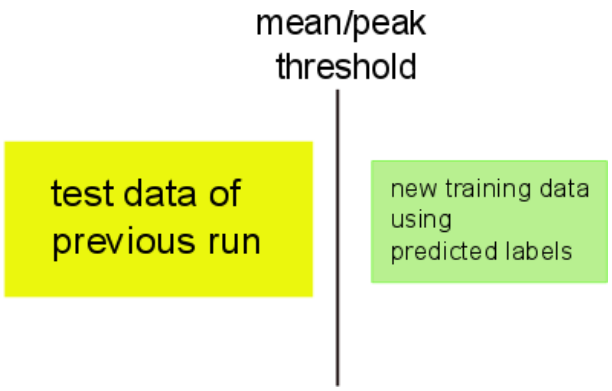


Figure 2.4: Mean and Peak Methods for data pool creation.

2.5.2 Random Forest-based Methods

Cumulative Method

Analogous to the cumulative method using an LDA classifier, also for the RF, a majority vote is implemented. Trials whose decision tree counts of the two class labels reaches a certain difference are deemed to be reliable and therefore added to the training data. For a schematic figure of this method, see Fig. 2.3 above.

Gini Method

This method, in contrast to the aforementioned, does not use the classifier's predictions, but instead the signal features for sorting. The features of the classifier are sorted by importance according to the Gini index and, subsequently, only the best 90% are kept for further training data.

After evaluating the simulation results obtained with all the aforementioned approaches, the boxplot peak method was selected for use in the supporting online study and subsequently implemented in the online system.

2.6 System Setup

Based on the system models of Faller et al. [21] and Schwarz et al. [22], the proposed BCI system is divided into two main parts. The online system framework works as standard BCI system including preprocessing and classification of trials, as well as giving feedback to users based on their actions. In addition, the framework sends all acquired trial data to the optimizer. The separate optimizer is used to collect trial data and recurrently retrain the classifier at certain times, including the newly accumulated data that has passed the outlier rejection. The updates for both the CSP filters and the classifier model are sent back to the system.
These two main components are implemented in Simulink/Matlab each, and communicate via TCP/UDP [43]. They can run on two different computers as well as on two different Matlab instances on the same PC. Figure 2.5 shows the schematic structure of this extended system.

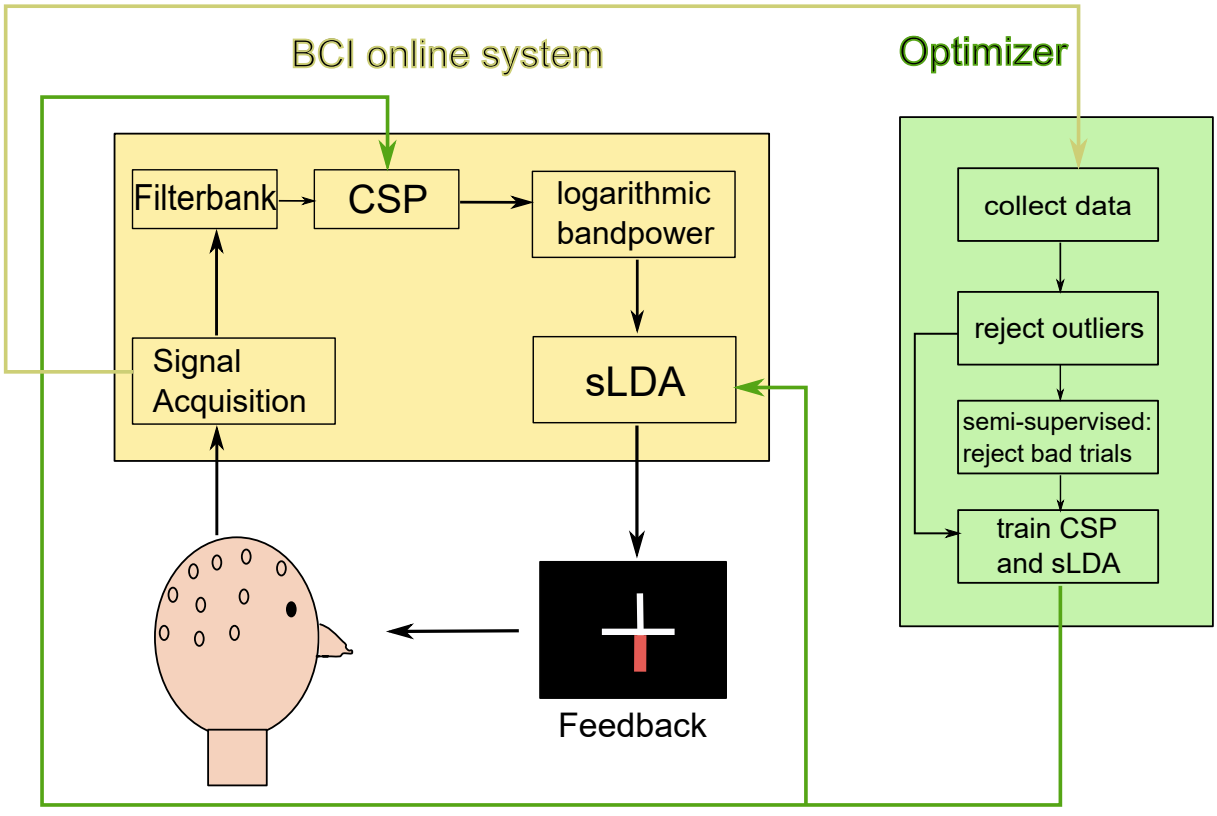


Figure 2.5: Schematic overview of the proposed online system including the separate optimizer instance.

2.6.1 Software and Toolboxes

Following software and toolboxes were used for simulation with offline data and realization of the online study:

1. Matlab 2013a (Mathworks Inc., Natick, USA) for implementing and running the complete online system.
2. Matlab 2015b (Mathworks Inc., Natick, USA) for statistical analysis of the results.
3. Random Forest mex implementation for Matlab [44]
4. TCP/UDP/IP Toolbox 2.0.6 [43]
5. Fast Serialize [45]
6. TOBI SignalServer + Client [46]
7. Graz BCI libraries [47]

2.6.2 Online System and its Components

Both, the adaptive BCI system and the optimizer instance are implemented in Matlab, using Graz BCI libraries. A more detailed overview of the system is shown in Figure 2.6. Figure 2.7 shows the Simulink model of the online system.

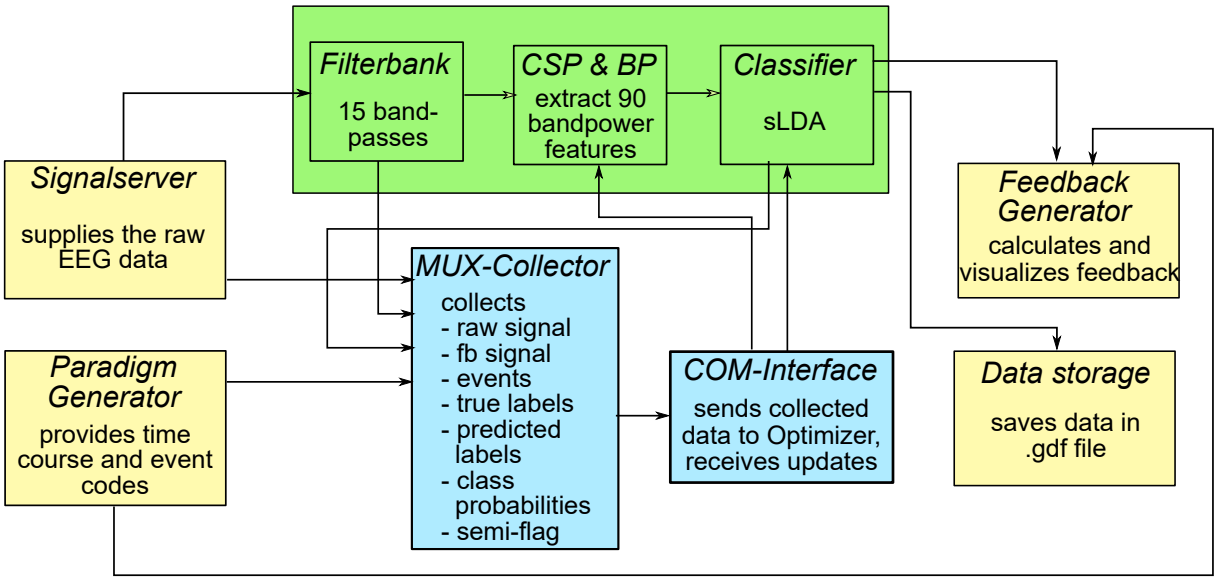


Figure 2.6: Schematic overview of the online system as implemented in Simulink.

First, the Tobi Signalserver [46] feeds raw EEG data recorded from the 13 electrode channels into the system. The signals are then processed in the 15 bandpass filters of the filterbank, resulting in a total of 195 signals (13 for every bandpass), and afterwards passed on to the CSP block, where every filterbank input is handled by its own specifically trained CSP filter. In the end, 90 signals are extracted by taking the first and last three columns of every CSP-filtered signal. 90 features are thereafter extracted through calculation of logarithmic bandpower. The

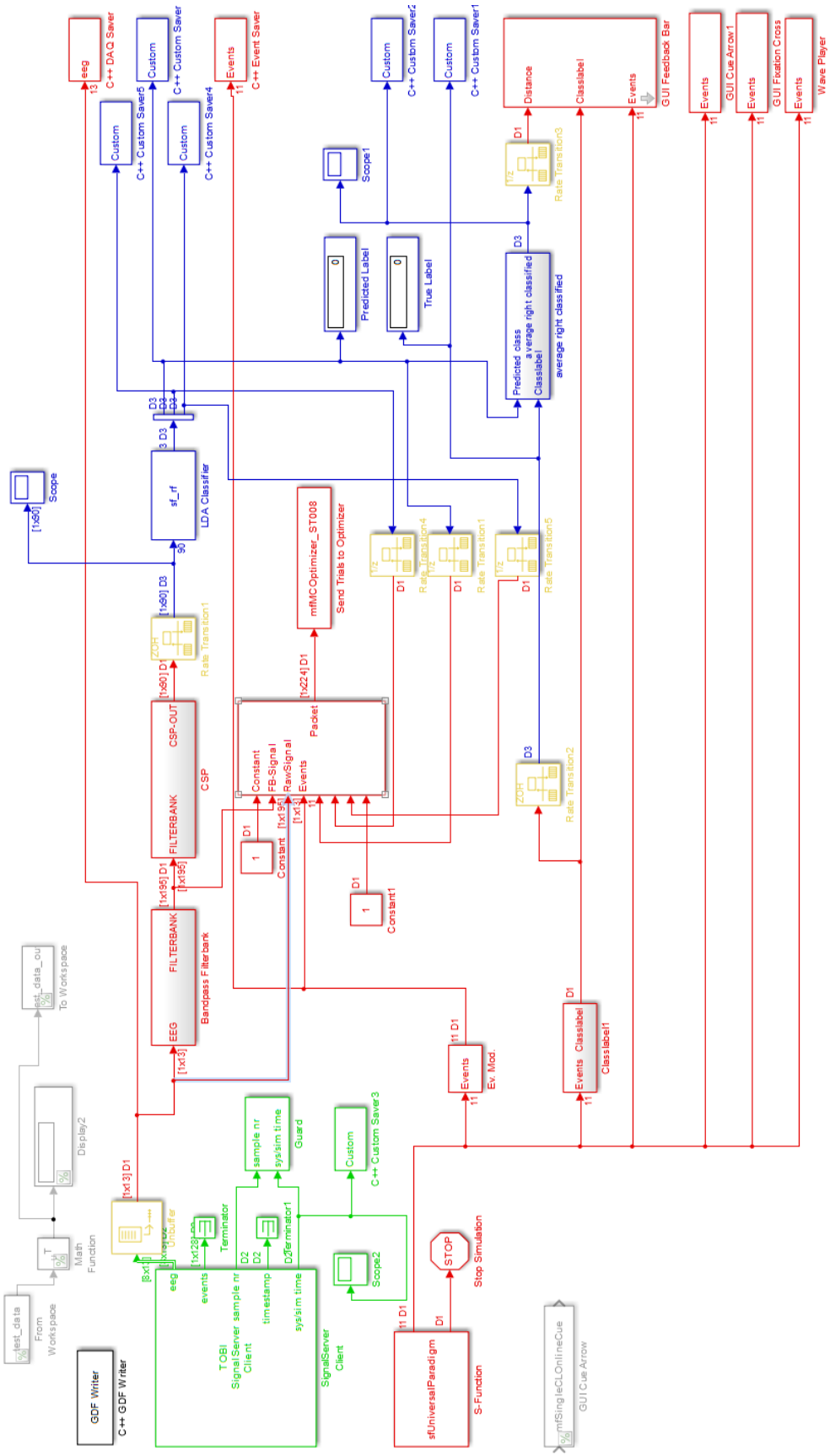


Figure 2.7: The online system as implemented in Simulink.

sLDA classifier then assigns a class label to these inputs. Classification happens 16 times per second.
The paradigm generator determines the time course of the experiment by creating events in a specific order. Changes in the paradigm, e.g. length of one run or number of classes, can be made in the paradigm .xml-file that contains all information for events over the course of one trial (succession, time points, class labels, length) as well as one run. This block sends information about true class labels to the feedback generator and event codes to the MUX section that collects data for the optimizer.
Feedback is provided by the feedback generator which compares true class labels from the paradigm generator with the predicted labels generated by the classifier. The result is visualised as a red feedback bar on the user's screen. The user receives feedback on his/her performance as soon as the classifier is trained for the first time.
Data is saved to a .gdf-file for every run. This file consists of raw EEG data, outputs of the classifier (predicted class label and class probabilities), true class labels and event codes provided by the paradigm generator.
The link between the BCI system and the optimizer instance is an s-function for sending and receiving trial data. Raw EEG data, signals of the filterbank, the classifier's output (predicted label and two class probabilities), event codes, true class labels and a flag indicating the learning type (supervised or semi-supervised) are collected and sent to the optimizer via the s-function. Moreover, the s-function receives updates for the classifier and the CSP filters from the optimizer.
The BCI system and the optimizer communicate via TCP/IP. Figure 2.8 shows a diagram of the workflow of the interface.

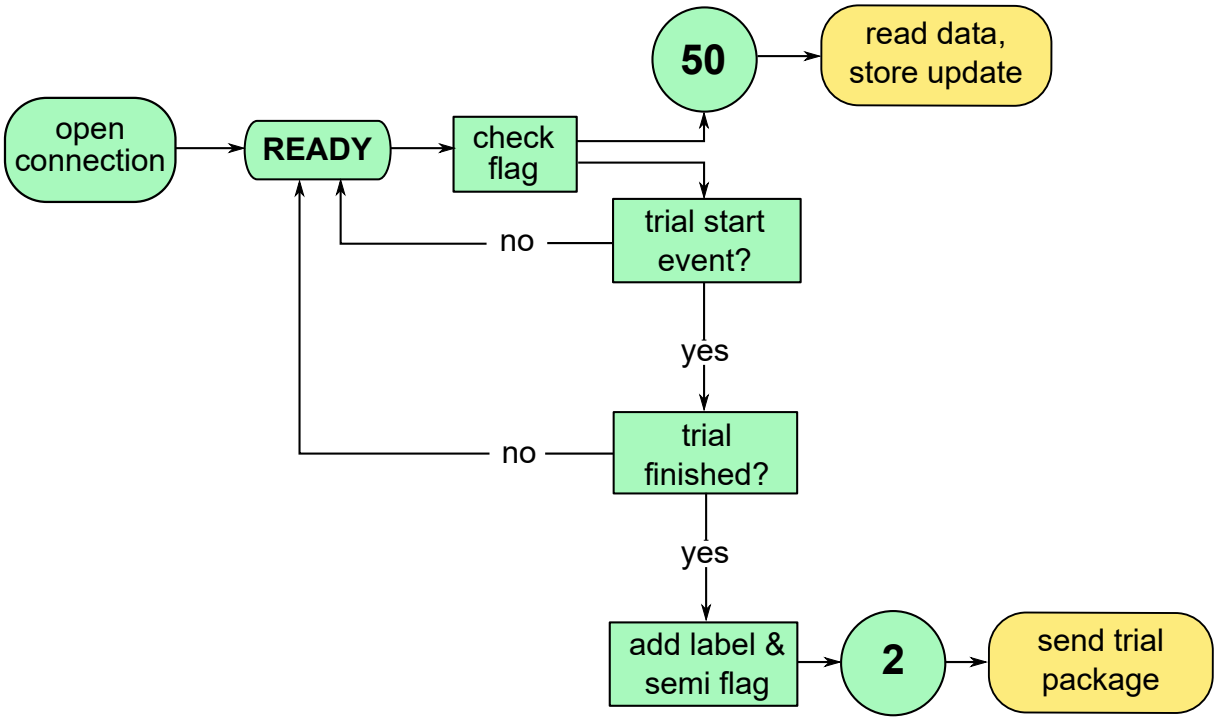


Figure 2.8: Schematic of the workflow of the interface for communication between the BCI system and the optimizer.

Different flags indicate different actions. After the interface first opens the connection to the optimizer, it waits for further cues. If it receives 'flag 50', an update for the classifier and CSP model is accepted, read and stored for use in the BCI system. Apart from that, the interface

checks the incoming data stream for the trial start event which originates from the paradigm generator. Upon finding this event, data is stored until the trial is finished. The length of every trial is defined by the paradigm generator. After collecting EEG data from the start trigger until the end of the fixed trial length, the trial's class label and a flag to indicate supervised or semi-supervised learning is added to the data. Afterwards, 'flag 2' and subsequently the whole package are sent to the optimizer.
Since parts of the system work at different rates (e.g. 256 Hz EEG data stream, 16 Hz classification), multiple rate transitions are built into crucial junctions of the model.

Optimizer

The separate optimizer communicates with the BCI system through a TCP/IP connection. The optimizer is realized as a Matlab function. A detailed overview of the optimizer's structure is depicted in Figure 2.9.

After the optimizer is started for the first time in a session, it remains active and ready until it is shut down again. In this readystate, incoming flags are checked and, depending on the flag's number, following actions are executed:
For 'flag 0', the connection to the BCI system is closed and the optimizer shut down, after data in the optimizer's workspace is saved. This trigger is usually sent after a user's session has finished.
In case of 'flag 1', the connection is closed and all the stored data saved from the workspace to a separate file. This is the normal procedure after one run is finished, but the session still goes on. In this manner, the optimizer instance is not shut down and newly recorded data of every run is added to the stack of already accumulated data, making it possible to collect trials over multiple runs without any losses.
Both for flags 0 and 1, all data from the optimizer's workspace is saved in a separate file in case the optimizer shuts down unexpectedly during a session. In such a case, the optimizer function can be started again and the data file is loaded into the workspace. This means that the session can be continued from the point of interruption without any data loss or need to repeat parts of the recording.
During a run, trials are sent from the system to the optimizer. 'Flag 2' indicates an incoming trial package and opens the connection between system and optimizer. Data is received and added to the already stored trials, after being separated into EEG data, class label, predicted label, class probabilities and the learning mode flag. The total amount of accumulated trials per class is then checked and compared to the minimum number of trials per class necessary for the next retraining. If enough trials are collected, the outlier rejection controls all data and dismisses trials that are afflicted with artefacts. The following step depends on the number of trials that have already been gathered. For an initial period of 40 TPC, the next move is checking the amount of trials left after the outlier rejection and subsequently deciding to re-train again or not. If this initial amount of trials has already been recorded and the learning mode flag is set to semi-supervised learning, the exclusion criterion for the training data pool (boxplot peak method) is applied. Thereafter, the algorithm again examines the number of remaining trials. Retraining of the CSP filter and the sLDA classifier is performed if enough data has accumulated and passed the outlier rejection and exclusion criterion. For the CSP filter, features from the feedback period (second 4.75 to second 7.75) of every trial are used. The sLDA is trained on features at second 2.5 after presentation of the visual cue in the paradigm (following [22]).
The update for CSP and sLDA is then packaged using the Fast Serialize toolbox [45] and sent to the system ('flag 50'), which upon arrival immediately starts to apply it to newly incoming EEG data. Furthermore, every new CSP and classifier model is separately saved in a .mat file

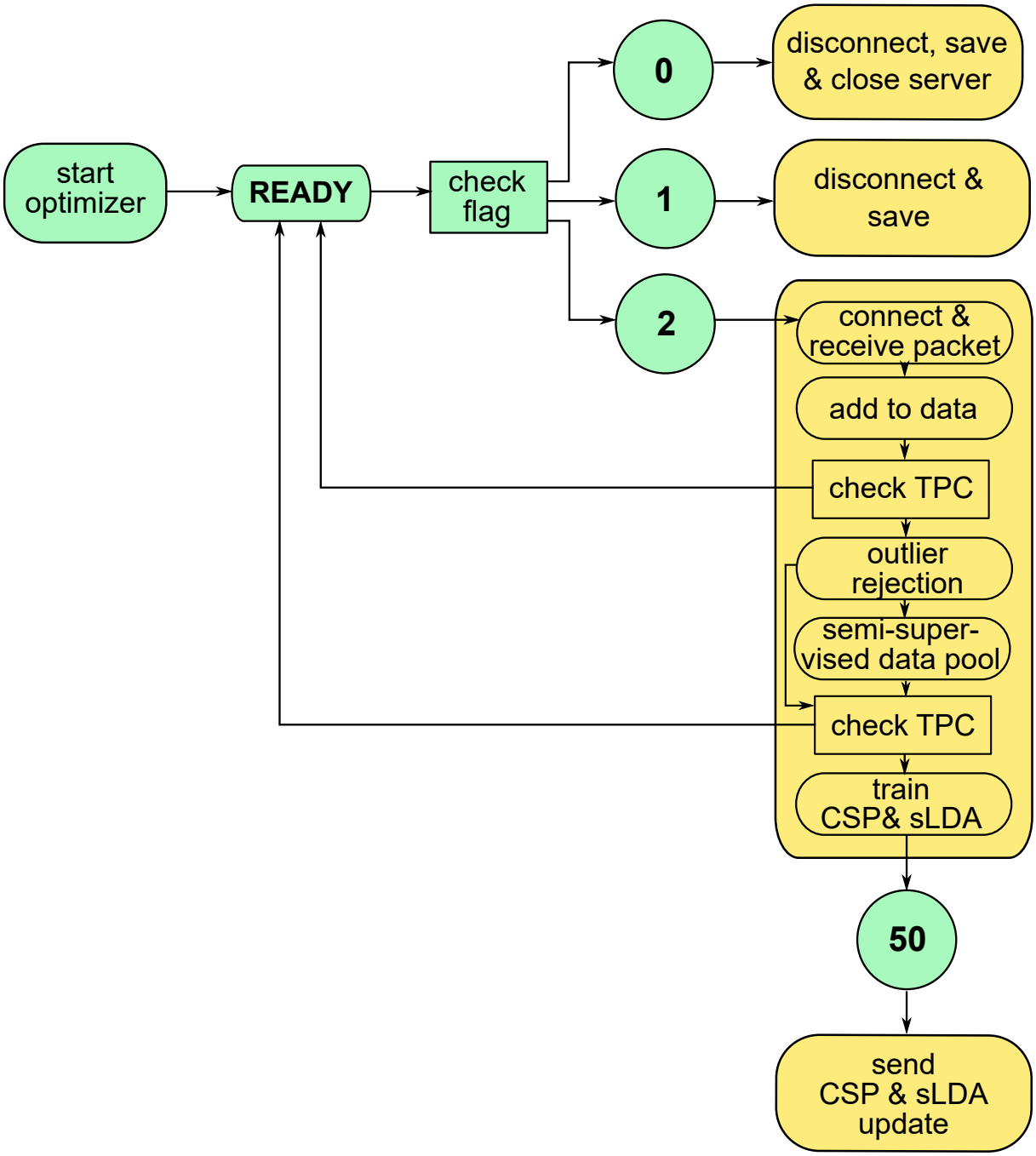


Figure 2.9: Structure of the separate optimizer.

by the interface.
With reference to Faller et al. [21] and Schwarz et al. [22], the classifier is trained for the first time after 10 TPC have passed the outlier rejection, providing the user with feedback after only a few minutes of recording. Further retrainings of the classifier happen every time that 5 new TPC pass the outlier rejection. After an initial phase of data recording (40 TPC), the system can be switched into semi-supervised learning mode. From then on, all newly added trials are filtered with the aforementioned boxplot peak method to decide if they should be added to the training data or not. Following every retraining, the number of TPC necessary for the next training is updated to be again checked before and after an outlier rejection and possible exclusion criteria are performed.

2.7 Paradigm and Experimental Setup

The online part of this work was tested on 20 healthy volunteers, dividing them in two groups of ten subjects to make a comparison between the semi-supervised approach and the normal supervised system possible.

2.7.1 Participants

Twenty healthy volunteers participated in the online study. They were assigned into two independent groups. The first group was trained with supervised learning while for the second group, semi-supervised learning started after the initial training phase.
The supervised learning group comprised of 10 healthy subjects, 4 female and 6 male, between ages 18 and 26 (median age 24). Three subjects (2 male, 1 female) of this group were non-naive users.
For the semi-supervised learning group, 10 healthy subjects, 5 male and 5 female, from ages 20 to 35 (median age 24) were measured. Four subjects of this group, 2 male and 2 female, were non-naive users.

2.7.2 Paradigm and Feedback

The standard Graz-BCI paradigm [9] with two classes was used for the experiment. Each trial in the paradigm takes 8 seconds. A detailed overview of it is shown in Figure 2.10.

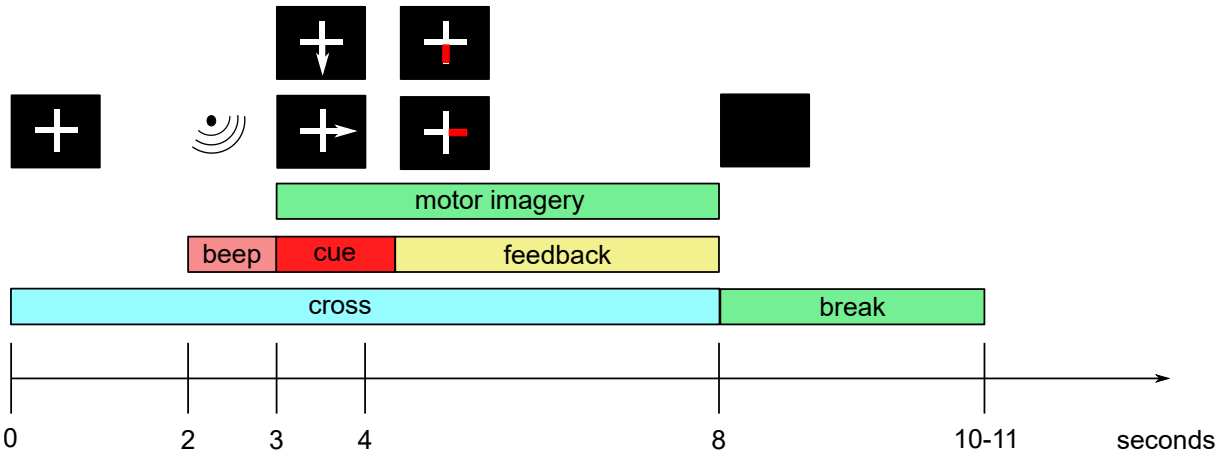


Figure 2.10: Overview of the Graz BCI paradigm.

At second 0, the trial starts with the appearance of a white cross on the black screen, in front of which the user is seated. At second 2, an auditory cue in the form of a beep is given to get the subject's attention and focus on the subsequent task. Then, at second 3, the visual cue appears to indicate which motor imagery task should be performed. An arrow to the right signals motor imagery of the right hand, whereas an arrow down means motor imagery of both feet. The arrow disappears after 1.25 s. The motor imagery task should still be performed until,

at second 8, the cross disappears as well.
For right hand motor imagery, users were instructed to imagine repeated opening and closing of the whole hand, like they are squeezing a small ball. For motor imagery of both feet, users should imagine repeated lifting and lowering of their toes and forefeet from the ground (repeated plantar flexion/extension of their feet).
Feedback is shown starting from second 4.25 (when the cue arrow vanishes) in form of a red bar that evolves along the direction of the previous cue arrow. The length of the bar depends on the strength of the user's performance, resulting from comparing true and predicted class labels of the EEG data. Only positive feedback is shown: if the number of correct classifications during the last second is more than half of the total amount of classifications in that period, a red bar appears, its length depending on the ratio of correct classifications and total classifications. If less than half the classifications are correct, the bar disappears. The calculation of feedback is updated 16 times per second. The feedback bar is only presented after the classifier has been trained for the first time. Figure 2.11 shows the user's screen with fully positive feedback. In case of a worsening performance, the red bar shortens accordingly.

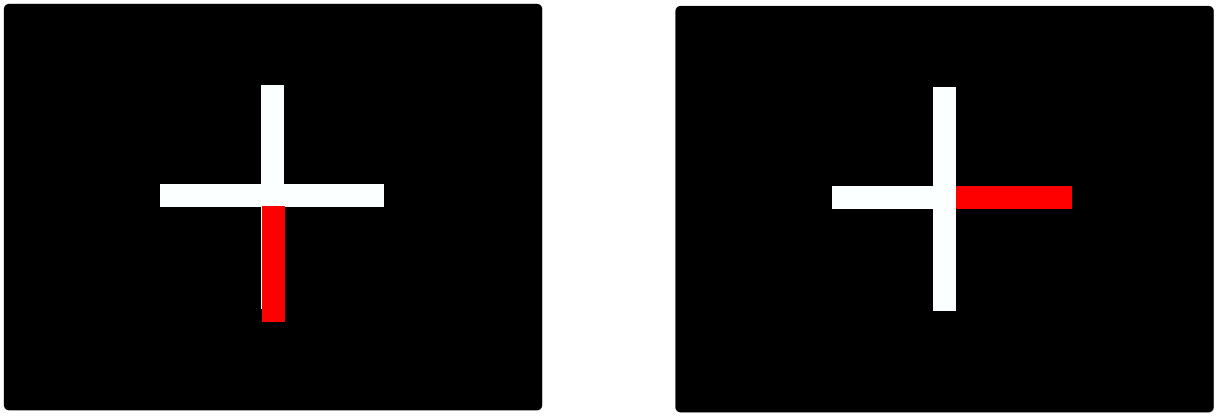


Figure 2.11: Feedback for good performance of feet (left) and hand (right) motor imagery.

Between consecutive trials, there is a randomised break of 2 to 3 s. During trials, the user is instructed not to blink or swallow as far as possible. These movements can be performed during the breaks to avoid artefacts [9].

2.7.3 EEG Recording

All experiments were carried out in the laboratory environment of the Institute of Neural Engineering at Graz University of Technology. Subjects were comfortably seated in a shielded dark measurement box at about 80 cm distance from a computer screen. The paradigm was shown on the screen and two speaker boxes were placed at its sides to present the auditory cues.
To record EEG data, a g.tec GAMMA electrode cap (g.tec, Graz, Austria) was mounted with 13 active g.tec Ladybird electrodes (g.tec, Graz, Austria) according to the international 10-20-system [48]. Positions of the electrodes can be seen in Figure 2.12 and in Table 2.2. The ground and reference electrodes were placed on AFz and the left ear lobe, respectively.

A g.tec GAMMA box and g.tec USBamp (g.tec, Graz, Austria) amplified the signal and connected the electrodes with the computer. The USB-Amp allowed for filtering data with both a notch filter (50 Hz) and a bandpass filter between 0.1 and 100 Hz.

Table 2.2: Positions of the electrodes in the online experiment.

Electrode	1	2	3	4	5	6	7	8	9	10	11	12	13	GND	Ref.
	FC3	FCz	FC4	C5	C3	C1	Cz	C2	C4	C6	CP3	CPz	CP4	AFz	left ear lobe

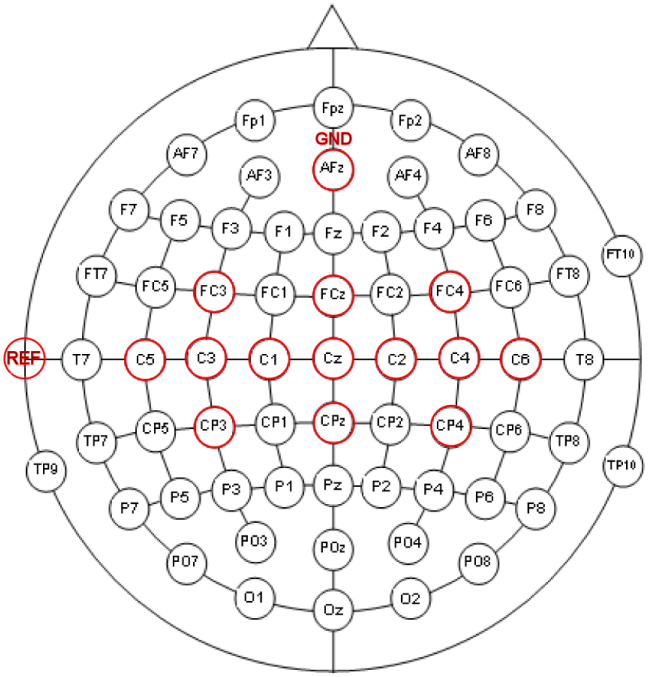


Figure 2.12: International 10-10 electrode setup. Electrode positions in the online experiment are indicated in red, including ground and reference electrode ([48], modified from [49]).

The sampling rate of EEG data was 256 Hz.

2.7.4 Experimental Design and Session Course

The two groups of ten healthy volunteers each were measured in the online study. Subjects were randomly assigned to one of the two groups. The first group was trained with supervised learning while for the second group, semi-supervised learning started after the initial training phase.

We extended the length of the measurement sessions by incorporating 20 minute breaks to make inferences about the adaptation of system and user over a longer time span.
Every session started with preparation of the subject (e.g. mounting of the electrode cap and starting of the system), which took approximately 10 to 15 minutes. Afterwards, the course of the session and the paradigm was explained. Every user was instructed about the tasks which should be performed. The instructions took about another 10 to 15 minutes.
After completing these preparations, the experiment was started. A schematic of one session can be seen in Figure 2.13.

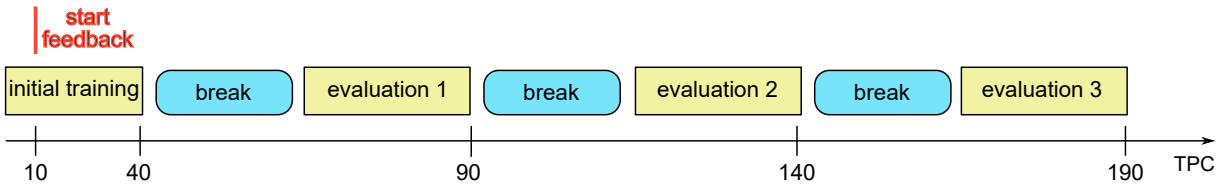


Figure 2.13: Course of one session, including an initial training phase, three evaluation periods and three breaks in between.

Figure 2.14 shows the whole experimental design including the split into two groups of subjects: First there is the initial training phase, then the first break, followed by three consecutive evaluation periods with two breaks in between.

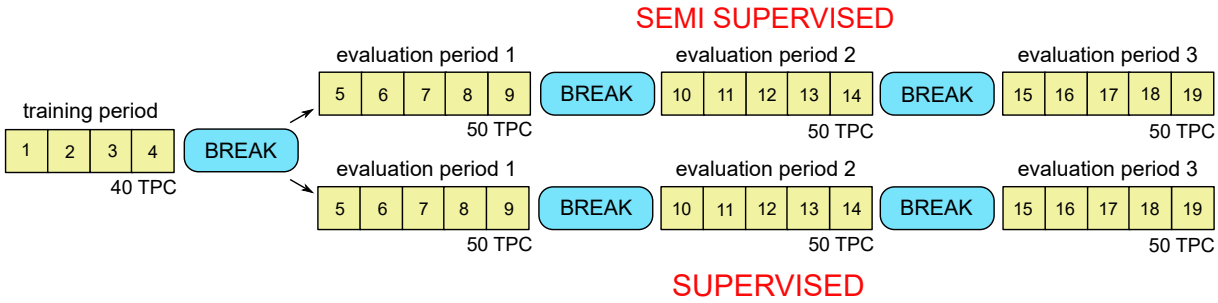


Figure 2.14: Overview of the experimental setup for two groups of subjects (supervised and semi-supervised).

Recording was started with an initial training phase of the system, consisting of 4 consecutive runs with 10 TPC each. One run lasted about 4 minutes. After every run, the subject was checked upon and it was made sure that users still felt comfortable. Feedback started after the first run, after 10 TPC had passed the outlier rejection.
After this first phase, there was a break of 20 minutes in which the subject watched a nature documentary on the screen in the measurement box. Subjects were allowed to sit back, relax and drink while watching. The goal of these breaks was to distract subjects from the MI tasks and change their mindset to something completely different, so that possible performance changes afterwards could be examined. For further analysis, EEG data was also recorded during the breaks.

The first evaluation period started after the first break, dividing the subjects in two groups: for one group, classification updates were continued with supervised learning, while for the other one, the classifier's updates were switched to semi-supervised learning. In the first evaluation period, 5 consecutive runs á 10 TPC were recorded.
The second break also lasted 20 minutes. Subjects continued to watch the previously started nature documentary and again, EEG data was recorded throughout.
For the second evaluation period, again 50 TPC were collected in 5 runs, before starting a third break of 20 minutes. During this last break, one episode of the TV show "The Simpsons" was watched by the users. In the last evaluation period, as before, 5 runs were recorded, resulting in 50 TPC.
Altogether 190 TPC were recorded per subject. Every session took around three hours, including preparation time.

3 Results

The results of this work are divided into two sections: One section for the offline simulations to find the best semi-supervised learning data pool approach and one section for the online study conducted thereafter.

3.1 Offline Analysis

Simulations for testing different approaches for a semi-supervised learning BCI system were performed using EEG data of 18 subjects recorded in Schwarz et al [22].

3.1.1 Accumulative Training Data Algorithm

Table 3.1 shows the peak, mean and median accuracies over all subjects for all semi-supervised methods as well as the supervised system using the accumulative algorithm for training data selection. Mean and median accuracies for every subject were calculated from the feedback period of second 4.5 to 7.5 of each trial. Additionally, semi-supervised learning was also tested without any preceding exclusion criteria to examine if the application of exclusion methods is even meaningful. Furthermore, the average number of retrainings for all approaches is shown.

Table 3.1: Results for offline simulations of all methods using accumulative data algorithm. Results showing statistically significant differences to ground truth are indicated with a star.

method	**peak in %**	**mean in %**	**median in %**	**retrainings**
LDA boxplot mean	87.53	74.21	75.75*	4
LDA boxplot peak	89.61	75.55	77.81	3
LDA mean threshold	87.62	74.68*	76.36*	6
LDA peak threshold	86.18	74.42*	76.26*	7
LDA cumulative	90.41	77.00	79.03	2
LDA without exclusion	86.32	74.51*	76.43*	7
ground truth: supervised LDA	*88.84*	*77.37*	*79.47*	*7*
RF gini index	66.99*	57.10*	57.22*	7
RF cumulative	90.63	76.14	78.09	2
RF without exclusion	87.02*	75.90	77.86*	7
ground truth: supervised RF	*91.30*	*78.42*	*80.21*	*7*

Statistical significance of the results was tested with a Wilcoxon signed rank sum test at a significance level of $\alpha = 0.05$. Results that are statistically significantly different from the ground truth (supervised LDA/RF) are marked with a star. Other results show no significant difference from the supervised method.

Mean accuracies over all subjects and trials were calculated for the offline results. Figure 3.1 shows accuracies for the different LDA-based methods over the trial length of 8 seconds. The supervised ground truth is displayed in black. At second 3, the black vertical line indicates the appearance of the visual cue and the start of the motor imagery.

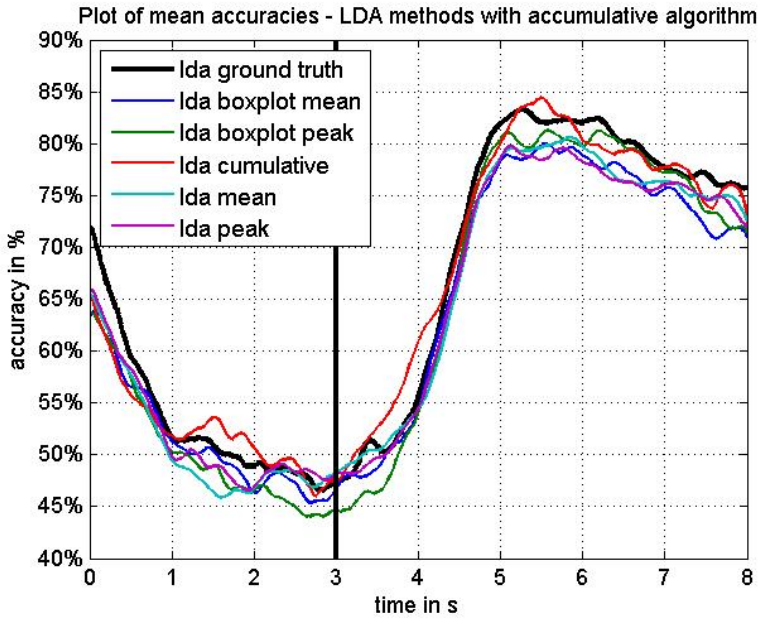


Figure 3.1: Mean accuracies for all semi-supervised LDA-based methods and the supervised ground truth (black) for accumulative training data algorithm.

Results for the RF-based methods are shown in Figure 3.2, where again the ground truth is indicated in black and the black vertical line shows the onset of motor imagery.

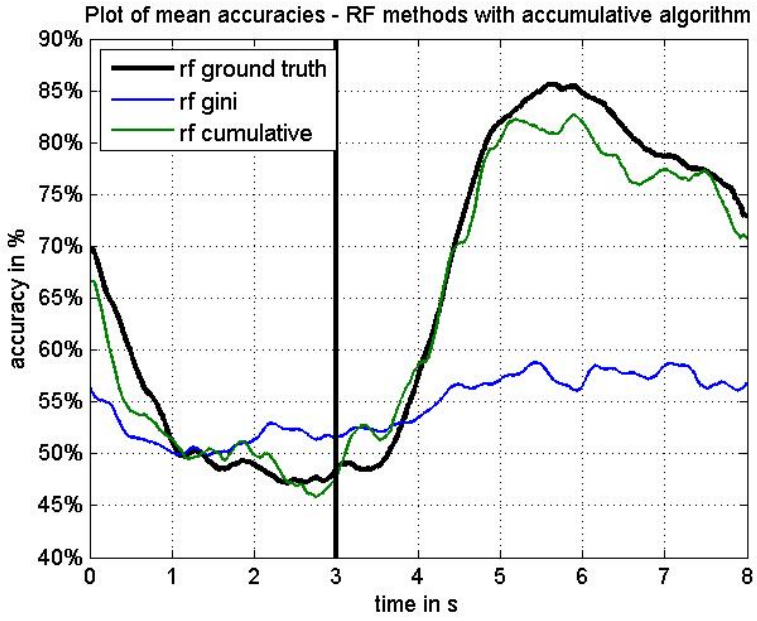


Figure 3.2: Comparison of mean accuracies of RF-based semi-supervised methods with ground truth (black) for accumulative training data algorithm.

3.1.2 Sliding Window Training Data Algorithm

The results for offline simulations using a sliding window algorithm on the training data are shown in Table 3.2. Again, peak, mean and median accuracy values over all subjects for all methods are displayed along with average numbers of trainings. Mean and median accuracy were calculated from second 4.5 to 7.5 of the trial (feedback period).

Table 3.2: Offline simulation results for sliding window algorithm. Results with a star indicate statistically significant difference to ground truth.

method	**peak in %**	**mean in %**	**median in %**	**retrainings**
LDA boxplot mean	87.45	74.09*	75.53	4
LDA boxplot peak	89.96	75.32	77.21	3
LDA mean threshold	88.09	75.08*	76.68*	6
LDA peak threshold	87.31	74.63*	76.11*	6
LDA cumulative	90.23	76.54	78.75	2
LDA without exclusion	87.36	74.74*	76.11*	7
ground truth: supervised LDA	*89.89*	*77.80*	*79.34*	*7*
RF gini index	63.72*	54.92*	55.06*	7
RF cumulative	91.09	76.94	79.02	1
RF without exclusion	85.86*	75.07*	76.49	7
ground truth: supervised RF	*90.79*	*78.69*	*80.45*	*7*

The results were tested statistically using a Wilcoxon signed rank sum test at a significance level of $\alpha = 0.05$. All methods that showed significant differences in peak, mean or median accuracies compared to the ground truth supervised LDA or RF are indicated by a star.

Mean accuracies over all subjects and trials were calculated for the sliding window algorithm. Results for the LDA-based methods for the sliding window algorithm are shown in Figure 3.3, where the supervised LDA is displayed as the black curve and the black vertical line marks the

appearance of the visual cue in the trial.

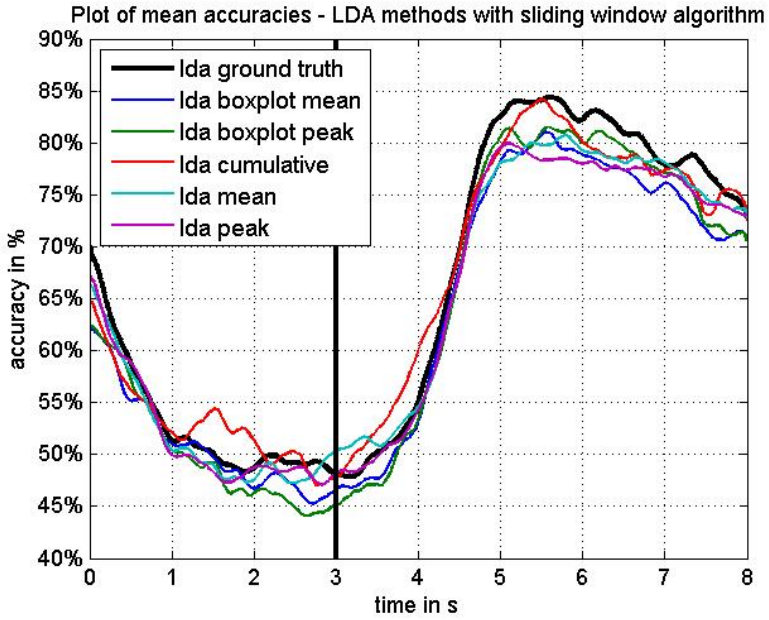


Figure 3.3: LDA-based semi-supervised methods and ground truth for the sliding window algorithm.

Figure 3.4 shows the offline mean accuracy results for the RF-based approaches, again highlighting the supervised classification method in black and the onset of the visual cue with the black vertical line.

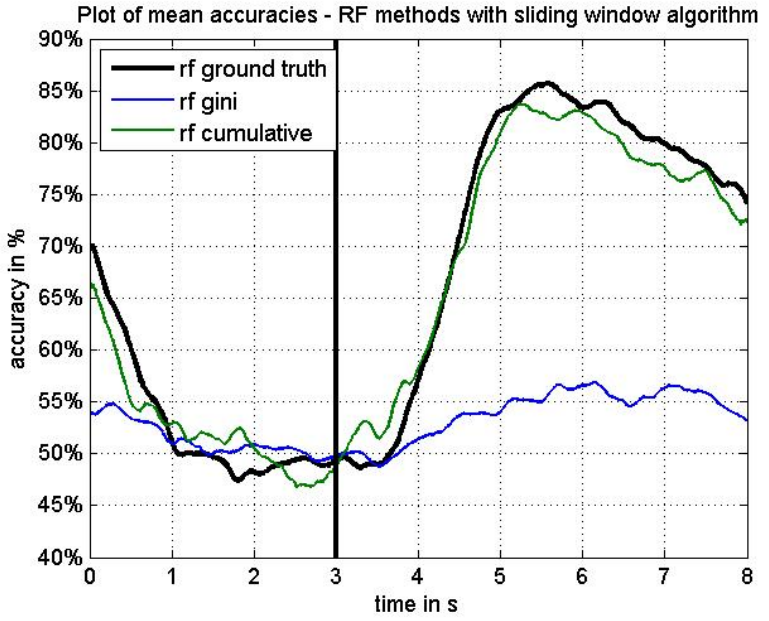


Figure 3.4: Semi-supervised RF-based methods and ground truth for the sliding window algorithm.

Based on the findings of the offline simulation analysis, the boxplot peak LDA-based semi-supervised learning method was implemented for testing in the online system.

3.2 Online Study

Results of the online experiment include subject accuracies (mean, peak, median), statistical tests of accuracies to determine significant differences between groups and stages, numbers of classifier retrainings, power spectral density plots and ERD/S maps. Detailed accuracy results for all individual subjects of both groups are shown as well as overall results in the first part of this section.

3.2.1 Accuracies and Statistical Testing

To get an understanding of the comparative quality of subjects' performances, the accuracy for the system working at chance level was calculated. For subjects to perform better than chance level in 180 recorded trials per class, they had to exceed a classification accuracy of 54.31%, e.g. achieve more than 196 correct classifications (determined at a significance level of $\alpha = 0.05$, using adjusted Wald interval).
First, accuracies for both groups were calculated and plotted.
Figure 3.5 shows mean accuracies over all experiment stages for every subject (grey graphs) and over all subjects (red graph) for the supervised learning group. The black dotted line indicates chance level and the black vertical line marks the appearance of the visual cue.

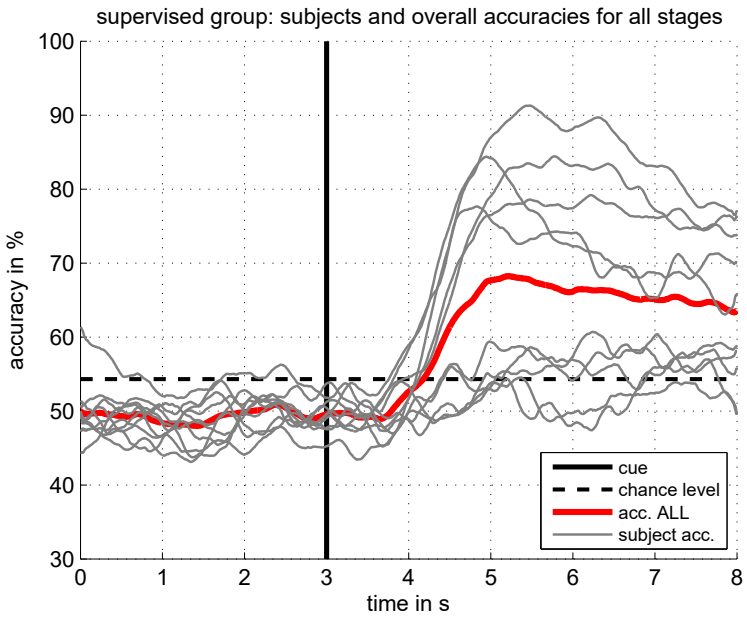


Figure 3.5: Mean subject's and overall accuracies over all experiment stages for the supervised learning group.

Accuracies for all four experiment stages of the supervised learning group are shown in Figure 3.6. Again, grey graphs indicate individual subject accuracies while graphs in colour mark the mean over all subjects.

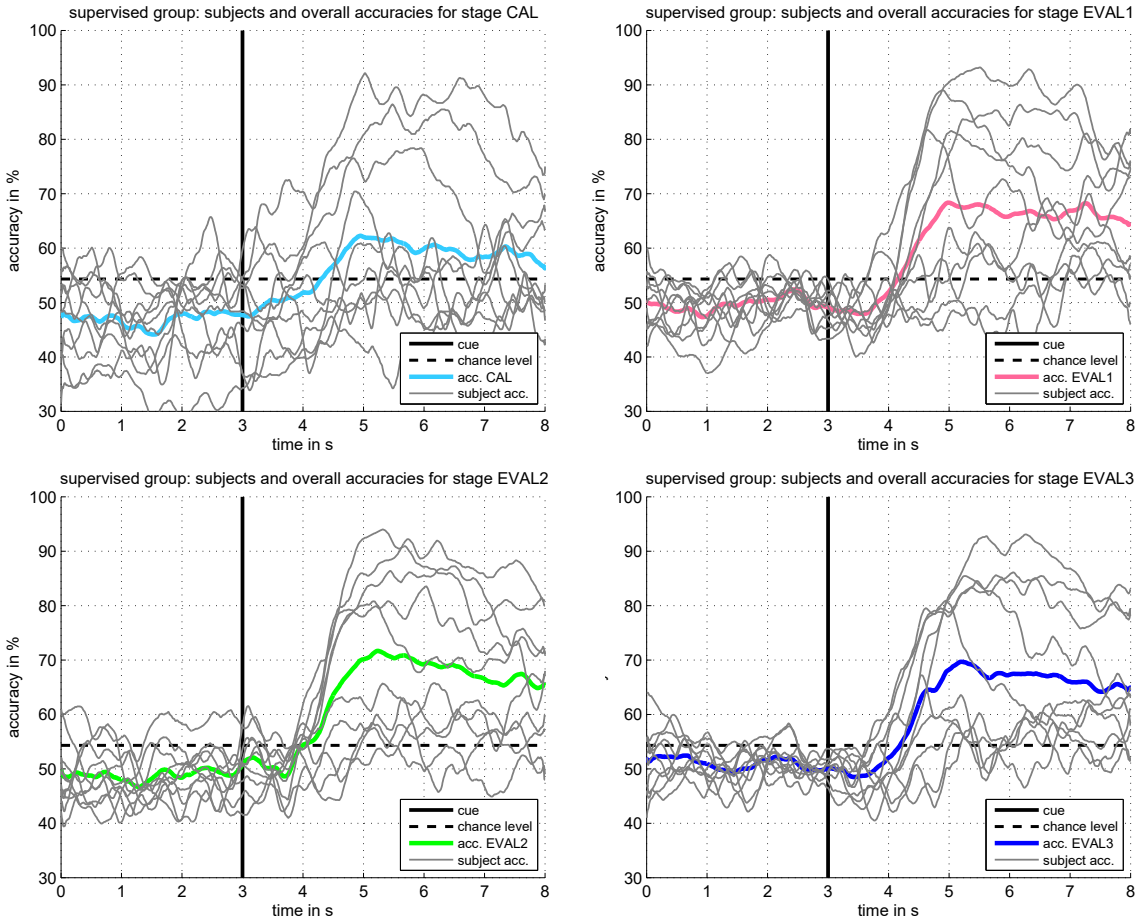


Figure 3.6: Mean accuracies (colour) and individual subject accuracies (grey) for all four experiment stages for the supervised group: top left calibration phase, top right evaluation phase one, bottom left evaluation phase two and bottom right evaluation phase three.

Accuracy results for the semi-supervised group were calculated in the same way for all individual stages of the experiment as well as one overall accuracy.
Figure 3.7 shows the individual subject's (grey) and mean (colour) accuracies over all four stages. Chance level is marked by the dotted black line, while the vertical black line indicates the appearance of the visual cue in the trial.

A plot of subject and mean accuracies of all four experiment stages for the semi-supervised learning group is shown in Figure 3.8. Grey curves are individual subject accuracies while graphs in colour mark the mean over all subjects.

Results for peak, mean and median accuracies in the feedback period from second 4.5 to second 7.5 of the trials, including the standard deviation, were calculated for the two groups and are displayed in Table 3.3.

Table 3.3: Peak, mean and median accuracies and standard deviations over all experiment periods for each group.

group	**peak in %**	**mean in % (4.5-7.5s)**	**median in % (4.5-7.5s)**
supervised	71.88 ± 13.26	66.06 ± 12.80	66.40 ± 13.12
semi-supervised	80.03 ± 15.90	74.69 ± 16.04	74.92 ± 16.23

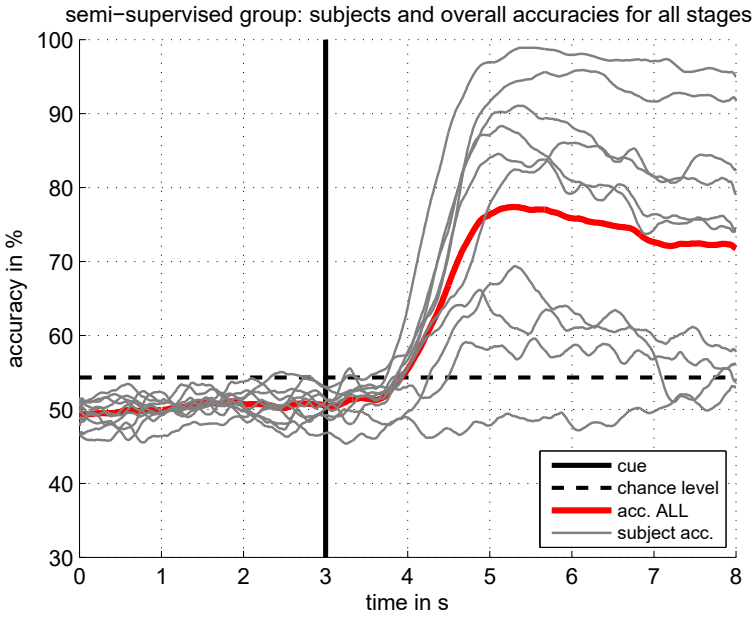


Figure 3.7: Mean accuracies over all experiment phases for individual subjects (grey) and over all subjects (red).

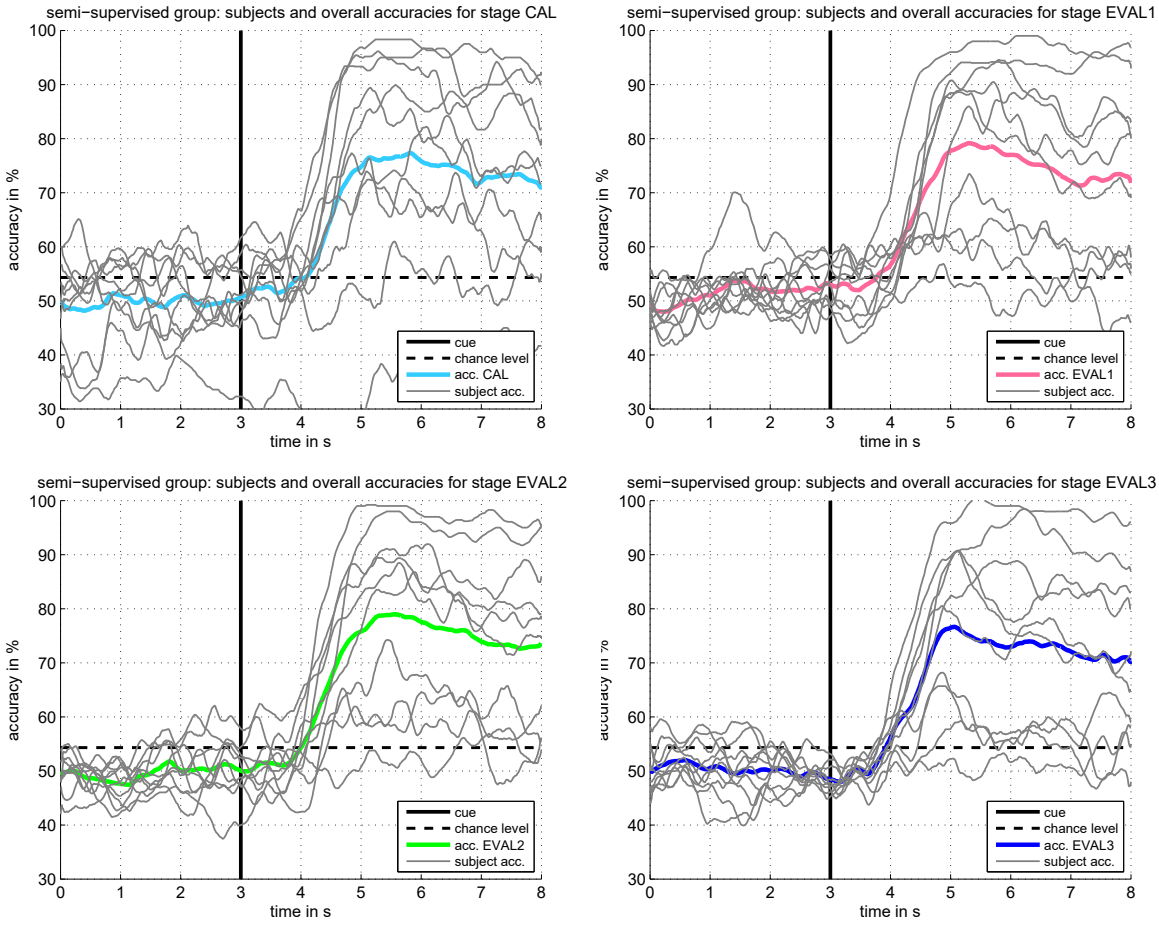


Figure 3.8: Mean accuracies (colour) and separate subject accuracies (grey) of all experiment stages for the semi-supervised learning group: top left the calibration phase, top right first evaluation period, bottom left the second evaluation phase and bottom right the third evaluation period.

For a more elaborate analysis of the different phases of the experiment, peak, mean and median accuracy values were also determined separately for every stage. Table 3.4 shows these results for both groups.

Table 3.4: Peak, mean and median accuracies (in feedback period) of all session stages for both subject groups.

group	experiment period	peak in %	mean in % (4.5-7.5s)	median in % (4.5-7.5s)
supervised	initial calibration	70.50 ± 13.15	59.96 ± 14.58	60.33 ± 14.55
	evaluation 1	76.70 ± 12.27	66.55 ± 13.75	66.70 ± 14.35
	evaluation 2	77.00 ± 12.91	68.52 ± 13.94	68.90 ± 14.13
	evaluation 3	75.60 ± 12.50	66.89 ± 13.12	67.70 ± 13.39
semi-supervised	initial calibration	82.17 ± 17.32	74.29 ± 19.20	74.33 ± 19.41
	evaluation 1	83.10 ± 14.54	75.31 ± 15.66	75.60 ± 15.90
	evaluation 2	83.60 ± 14.28	75.75 ± 15.40	76.20 ± 15.28
	evaluation 3	81.20 ± 15.53	73.23 ± 16.12	73.10 ± 16.47

For statistical testing, a two-sample t-test was performed first to determine statistically significant differences between the overall mean and peak accuracies of the two groups. Assumptions of normal distribution and equal variances of the samples were verified using a Shapiro-Wilk and an F-test, respectively. The two-sample t-test showed that there was no significant difference between the supervised and the semi-supervised overall mean accuracies in the feedback period of 4.5 to 7.5 s of the trials (significance level $\alpha = 0.05$, $p = 0.2003$). Additionally, no significant difference between the supervised and the semi-supervised learning group in overall peak accuracies was found (significance level $\alpha = 0.05$, $p = 0.2293$).
Boxplots of the mean accuracies of the feedback period and the peak accuracies for both groups are shown in Figure 3.9. The red line indicates the median and the boundaries of the blue box mark the second and third quartile.

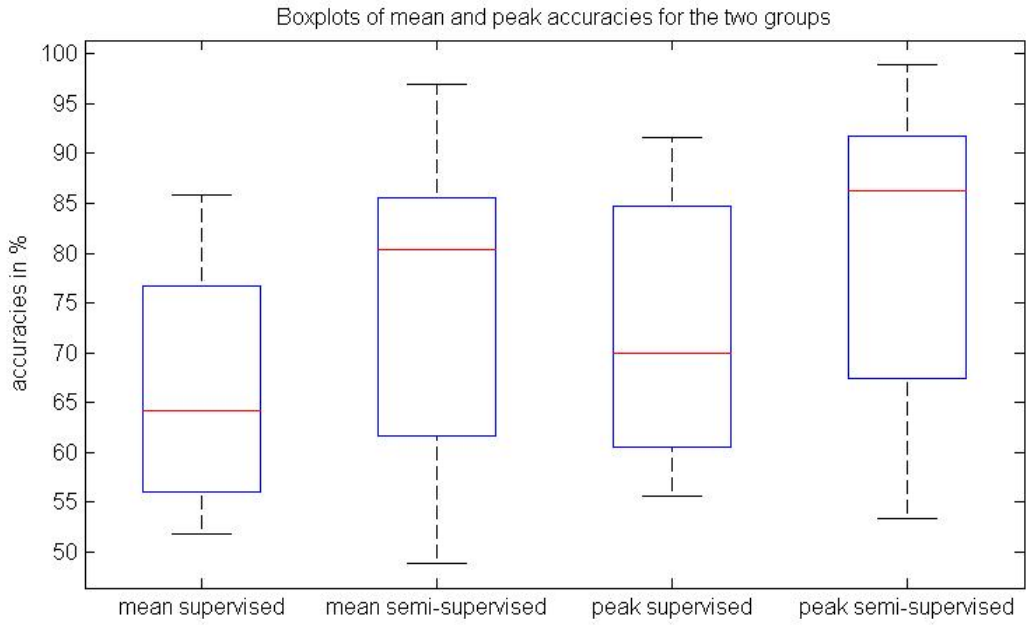


Figure 3.9: Boxplots of mean and peak accuracies for each group.

A mixed Analysis of Variance (ANOVA) (one between-subjects factor: group, one within-subjects factor: stage) was carried out to detect any statistically significant differences between the mean (of feedback period) and peak accuracies of the individual stages of the two groups. Before performing this test, the sphericity assumption was assessed using a Mauchly test.
There was no significant difference in mean accuracies of the feedback period (second 4.5 to second 7.5) in any of the experiment stages for the two groups ($F_{3,54} = 2.058$, $p > .05$). Furthermore, there was no significant difference in peak accuracies in any of the experiment stages for the two groups ($F_{3,54} = 1.22$, $p > .05$).

To investigate possible differences in mean (feedback period) and peak accuracies between the individual experiment stages, every stage (4 in total) was analysed against every other one, resulting in 6 tests in total, for each of the groups separately. A one-way repeated measures Analysis of Variance (ANOVA) was used to determine statistical significance at a significance level of $\alpha = 0.05$. Again, a Mauchly test was performed in advance to check for sphericity of data and p-values of the results were corrected for accordingly.
Results show that there is no statistically significant difference in mean accuracies of the feedback period (4.5 to 7.5 seconds of the trials) between the four different phases of the supervised learning group ($F_{3,27} = 3.165$, $p > .05$).
Additionally, no statistically significant differences could be reported in peak accuracies of the four stages for the supervised learning group ($F_{3,27} = 2.027$, $p > .05$).
For the semi-supervised learning group, there also were no significant differences in mean (feedback period) accuracies between any combination of stages ($F_{3,27} = 0.731$, ns). Furthermore, no significant differences of peak accuracies between any of the four stages were detected ($F_{3,27} = 0.635$, ns).

Group 1 - supervised learning algorithm

Detailed results for peak, mean and median accuracies for every subject in this group are shown in Table 3.5.

Table 3.5: Peak, mean and median accuracies over all stages for every subject of the supervised group.

subject	peak in %	mean in % (4.5-7.5s)	median in % (4.5-7.5s)
A1	60.56	56.31	56.39
A2	78.06	72.01	72.78
A3	91.67	85.81	87.22
A4	59.67	52.23	52.32
A5	61.94	56.15	56.67
A6	61.67	56.01	56.11
A7	55.56	51.76	51.94
A8	79.72	76.65	77.22
A9	85.23	74.11	72.22
A10	84.72	79.58	81.11
average ± std	71.88 ± 13.26	66.06 ± 12.80	66.40 ± 13.12

Figure 3.10 shows the accuracies for the individual experiment stages for subjects A1, A2, A3, A4 and A5.

Accuracies for all experiment stages of subjects A6, A7, A8, A9 and A10 are shown in Figure 3.11.

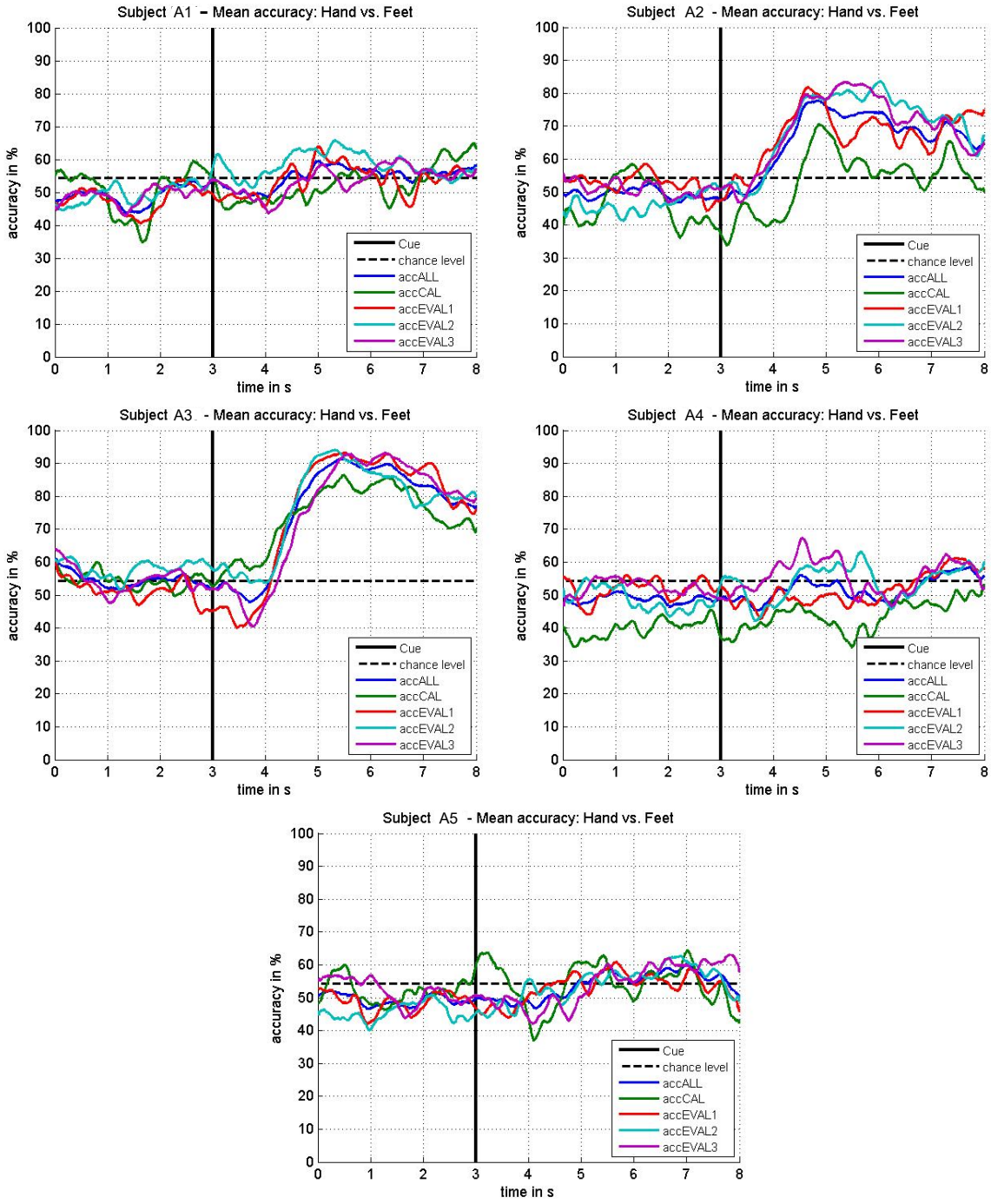


Figure 3.10: Mean accuracies for all stages for subjects A1, A2, A3, A4 and A5.

Group 2 - semi-supervised learning algorithm

The results in peak, mean and median accuracies for all subjects of the semi-supervised group is displayed in Table 3.6.

Figure 3.12 displays overall accuracies of all experiment phases for subjects B1, B2, B3, B4 and B5.

In Figure 3.13, overall accuracies of all stages for subjects B6, B7, B8, B9 and B10 are shown.

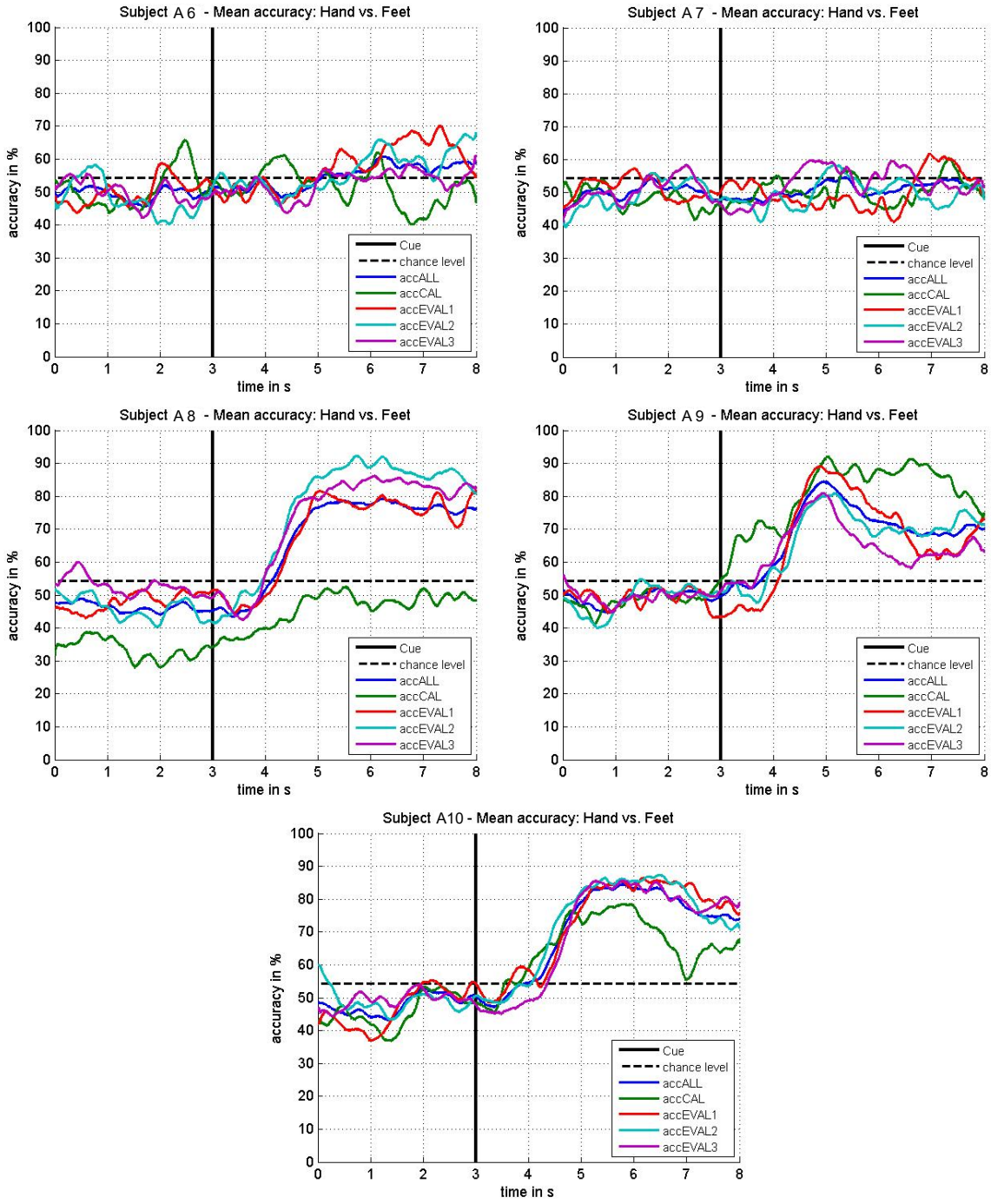


Figure 3.11: Overall accuracies for all stages for subjects A6, A7, A8, A9 and A10.

Table 3.6: Peak, mean and median accuracies over all stages for every subject of the semi-supervised group.

subject	**peak in %**	**mean in % (4.5-7.5s)**	**median in % (4.5-7.5s)**
B1	53.33	48.92	48.89
B2	96.39	92.24	93.06
B3	60.56	57.73	57.78
B4	86.94	81.02	82.22
B5	98.89	96.95	97.22
B6	70.00	61.89	62.78
B7	85.56	79.83	79.72
B8	91.71	85.63	85.91
B9	89.44	81.05	80.56
B10	67.50	61.66	61.11
average ± std	80.03 ± 15.90	74.69 ± 16.04	74.92 ± 16.23

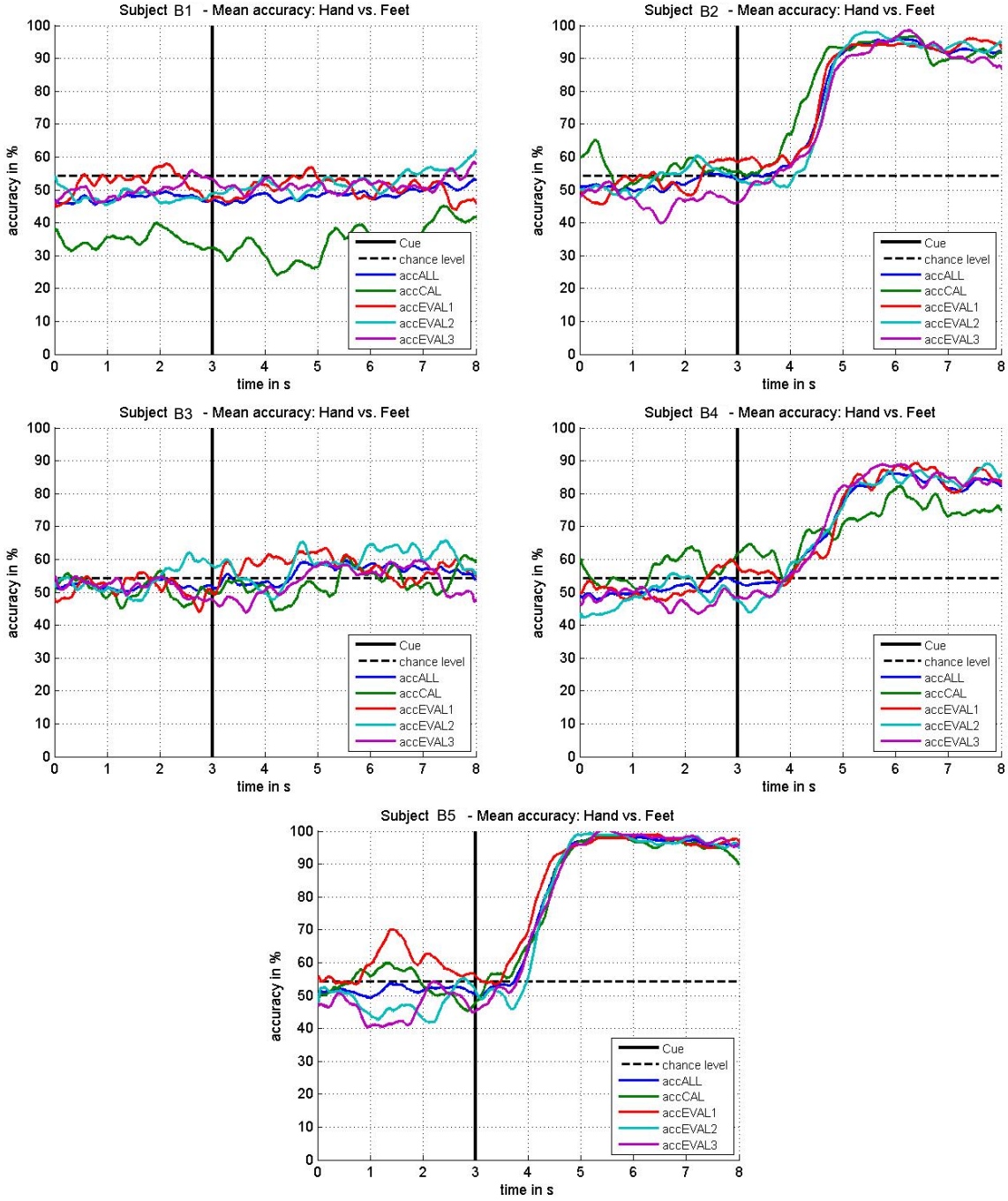


Figure 3.12: Overall accuracies for all experiment stages for subjects B1, B2, B3, B4 and B5.

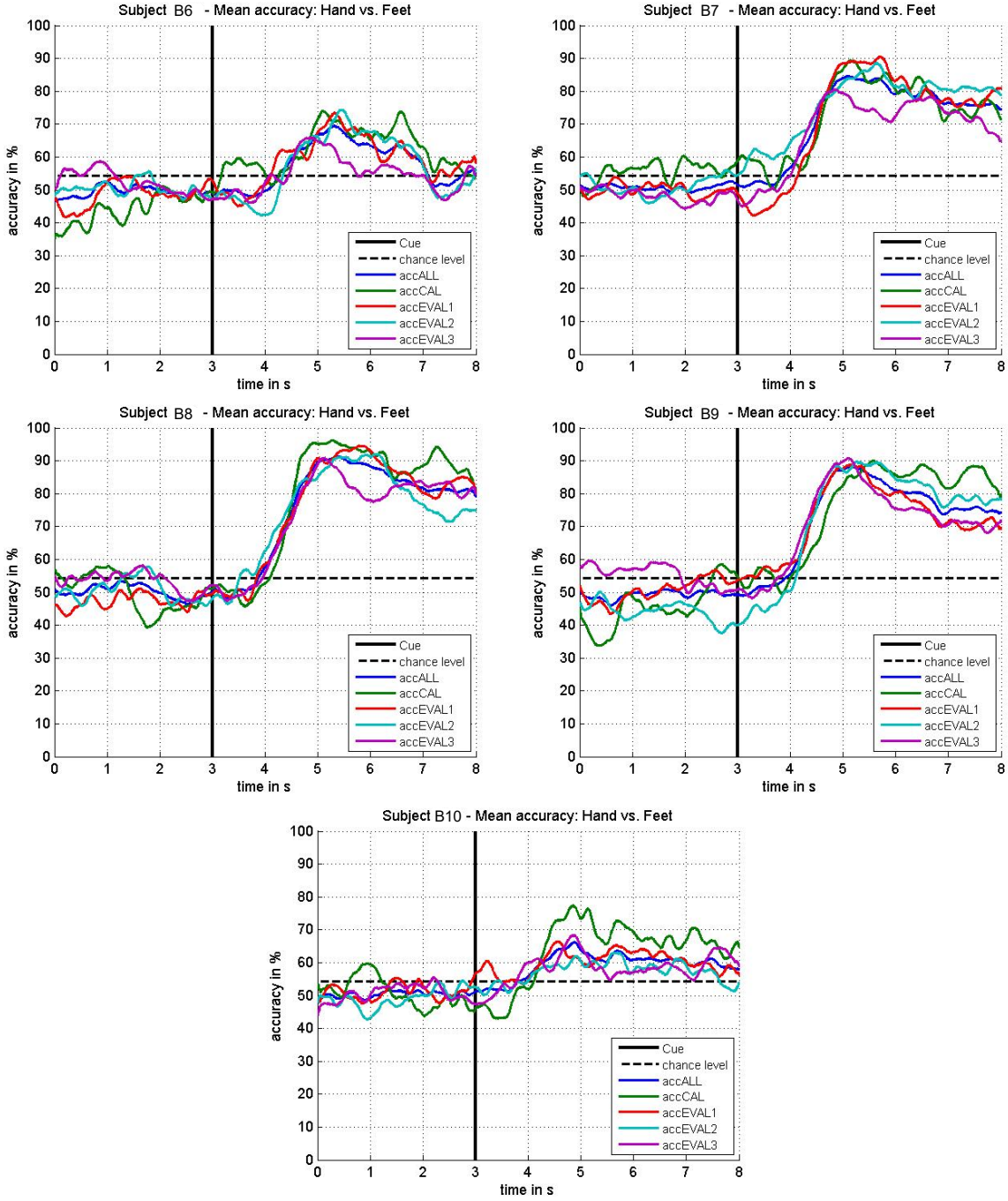


Figure 3.13: Overall accuracies for all experiment stages for subjects B6, B7, B8, B9 and B10.

3.2.2 Retrainings

For evaluation of the amount of trials dismissed in the semi-supervised learning algorithm, a comparison of numbers of retrainings in the two groups is displayed in Table 3.7. A retraining of the classifier happens every time 5 new TPC pass the outlier rejection and, for the semi-supervised group, the boxplot peak exclusion criterion. The average number of retrainings for each group is shown in the bottom row.

Table 3.7: Number of classifier retrainings for all subjects and on average for both groups.

subject	**retrainings supervised group**	**retrainings semi-supervised group**
A1/B1	27	7
A2/B2	29	15
A3/B3	31	13
A4/B4	29	10
A5/B5	29	18
A6/B6	22	8
A7/B7	27	11
A8/B8	25	13
A9/B9	28	12
A10/B10	23	7
average	27	12

For the supervised learning group, the classifier was retrained 27 times on average. On the contrary, classifier retrainings happened 12 times on average in the semi-supervised learning group.

3.2.3 Power Spectral Density

Power spectral density was calculated for every subject for channels C3, Cz and C4 using Laplacian derivation in the feedback period (4.5 - 7.5 s). Figure 3.14 shows the results for subjects of the supervised group. In Figure 3.15, PSD estimations of channels C3, Cz and C4 are shown for every subject of the semi-supervised group.
Blue graphs mark motor imagery of the right hand while green graphs depict MI of both feet. Differences in power between the individual channels are related to event-related (de)synchronisation for the two classes. Event-related desynchronisation and a subsequent power decrease of class hand is mainly apparent in channel C3 (see e.g. subjects A3 in Figure 3.14 and B5 in Figure 3.15), the cortical area connected with the right hand. ERD and the corresponding power decrease for class feet is visible especially for channel Cz (see e.g. subjects A10 in Figure 3.14 and B5 in Figure 3.15), indicating the activated cortical area for MI of both feet.

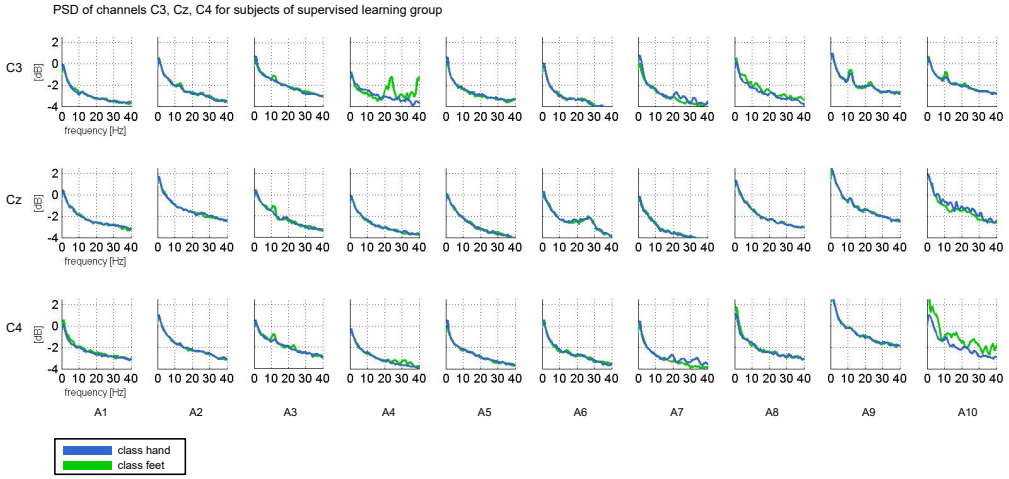


Figure 3.14: Power spectral density of channels C3, Cz and C4 for the supervised learning group. The abscissa shows frequencies [Hz], the ordinate depicts power in dB for the individual channels.

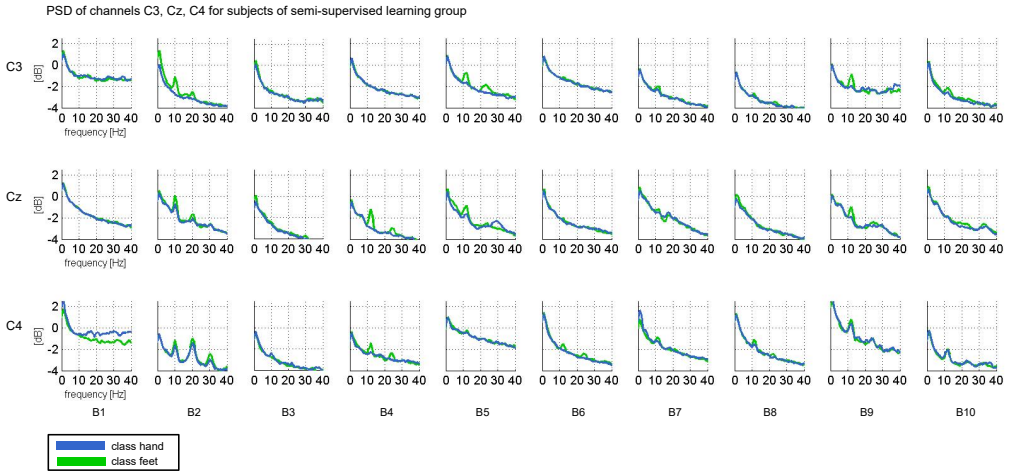


Figure 3.15: Power spectral densities for channels C3, Cz and C4 for subjects of the semi-supervised learning group. The abscissa shows frequencies [Hz], the ordinate depicts power in dB for the individual channels.

3.2.4 ERD/ERS Maps

Time-frequency maps were calculated for all subjects for channels C3, Cz and C4. Figure 3.16 shows one exemplary result of subject B2 (semi-supervised group) for class hand. The time-frequency plot for class feet of the same subject is shown in Figure 3.17. Yellow to red colours indicate weak to strong ERD, green to blue colours mark ERS. The ordinate shows frequency [Hz], while time [s] is plotted on the abscissa. The zero-point in the abscissa indicates the start of the MI period.

The reference period shows no clear ERD/ERS, indicating that there were no artifacts in this period. Distinct event-related desynchronisation patterns can be observed in the μ-band around 10 Hz for MI of the right hand. ERD is strongest in channel C3, which is the cortical area correlated with right hand MI. Weaker ERD patterns are also evident in the frequency range about 20 Hz (see Figure 3.16).
The time-frequency map for class feet shows less clear patterns, although ERD in the μ-band is visible for channels Cz and C4.

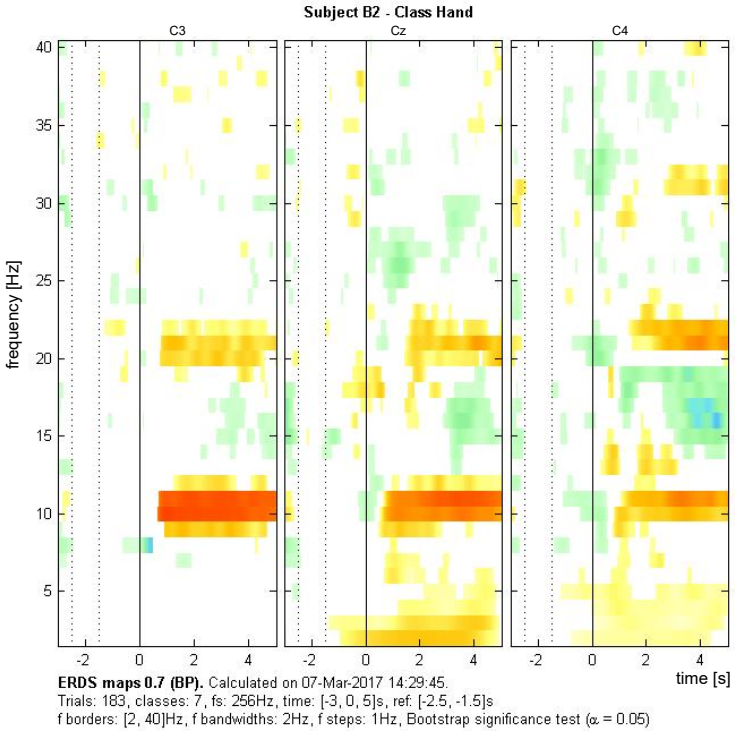


Figure 3.16: ERD/ERS map of subject B2 for class hand.

In Figure 3.18, the time-frequency map for class hand of subject A4 is depicted (supervised group). The ERD/ERS map for class feet of the same subject is shown in Figure 3.19. For this subject, no distinct patterns in either class are evident.

ERD/ERS maps of all other subjects are attached in the Appendix section.

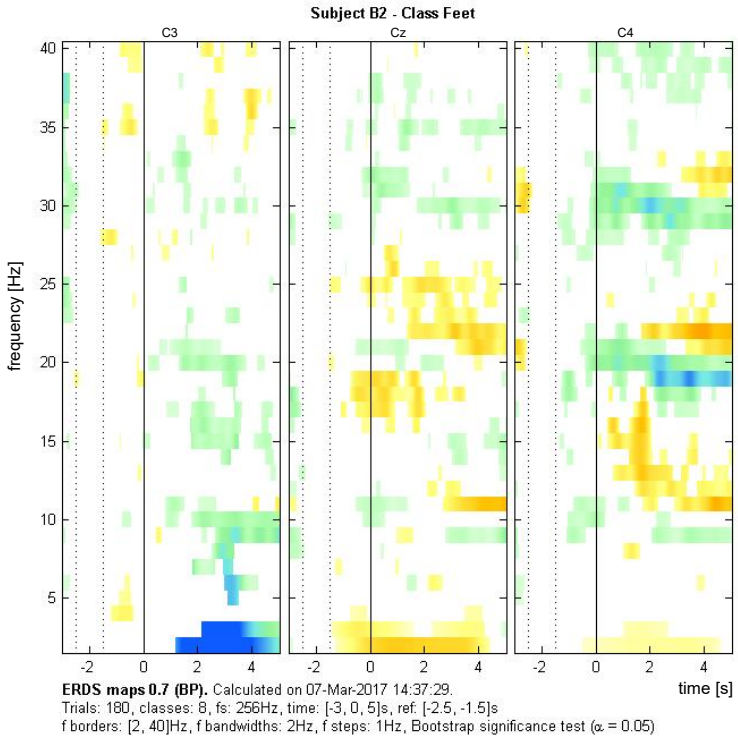


Figure 3.17: Time-frequency plot of subject B2 for class feet.

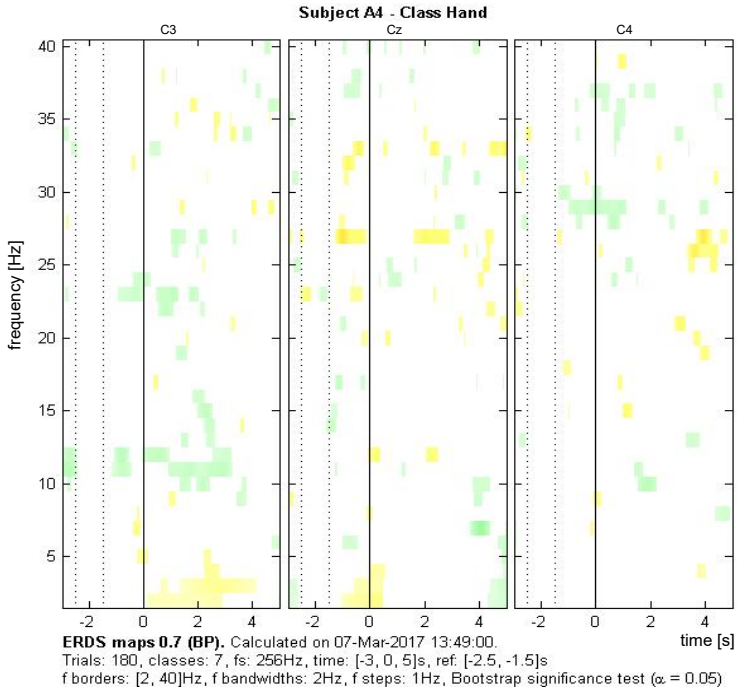


Figure 3.18: ERD/ERS map of class hand for subject A4.

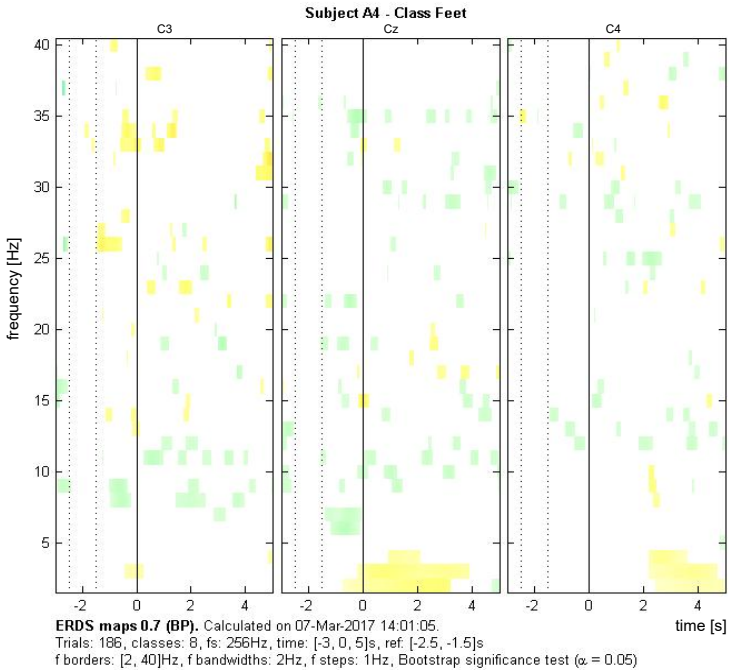


Figure 3.19: ERD/ERS map of class feet for subject A4.

4 Discussion

In this thesis we could show that implementing semi-supervised learning in an adaptive BCI system yields results comparable to those of a standard adaptive system. There was no significant difference in subjects' performances between the semi-supervised system and the standard supervised one in any experiment stage ($F_{3,54} = 2.058$, $p > .05$, $F_{3,54} = 1.22$, $p > .05$). This outcome is a first step towards further testing of real-life BCI applications in which the user's intent is not available to the system.
Apart from subjects' performance, we analysed power spectral densities and time-frequency maps to confirm performance results also from a neurophysiological point of view.
Regarding possible fatigue due to the long session duration, no significant performance drops over the course of the experiment could be found, although subjects reported exhaustion and difficulties to concentrate in later stages of the session.

4.1 Offline Results

A statistically significant difference in accuracies between a semi-supervised exclusion criterion and the supervised ground truth indicated that an approach was inadequate for further pursuit, since the goal was to find a solution which should yield approximately the same results as the supervised learning system.
Firstly, results for semi-supervised learning without any exclusion criteria suggest that implementing a suitable approach to sort out uncertain trials before every retraining is reasonable and necessary to achieve good classification performance.
In all semi-supervised approaches, it is still possible that trials with a wrong class label are included in the training.
Both the accumulative and the sliding window training data algorithm give similar results for all techniques, as evident in Tables 3.1 and 3.2. For example, the boxplot peak LDA-based method yields a peak and mean accuracy of 89.61 % and 75.55 %, respectively, for the accumulative training data algorithm, and a peak and mean accuracy of 89.96 % and 75.32 %, respectively, using the sliding window algorithm. However, the total amount of available recorded data (80 TPC) together with the number of initial training data (35 TPC) was generally not large enough to eventually remove all true-labeled trials from the training pool with only predicted-labeled data remaining for training. Therefore, the good performance of the sliding window algorithm could not be assessed for larger amounts of training data, as would be recorded for the online study. Moreover, comparable studies ([22] , [21]) also did not use a sliding window algorithm for training data selection.
When comparing the tested methods and ground truths in Tables 3.1 and 3.2, differences in the number of classifier trainings are noticeable. This was to be expected, since the approaches for semi-supervised learning, apart from removing outliers, also exclude data from the training pool that shows small classification certainty. This results in less available trials for the classifier than for the ground truths. It is also evident that the better the results of a semi-supervised method

are, the less trainings of the classifier take place. The best-performing semi-supervised method, the cumulative approach, in fact trains the classifier only two times on average, while for the mean threshold method, yielding significantly worse accuracies, the classifier is trained six times, nearly as much as for the ground truth. There seems to be a trade-off between the strictness of the method that rejects unreliable trials and the performance of the system. Apparently, approaches that perform better dismiss a larger number of trials from trainings, which means that less data is available for the classifier. Interestingly, a higher training number seemingly cannot compensate the inclusion of 'bad' trials in the training. Still, for the adaptive approach of fitting the classifier to the specific user input, recurring training is necessary.
As a compromise between high classification performance and number of classifier trainings, we chose the boxplot peak LDA-based method in an accumulative training data algorithm to be implemented in an online BCI system and tested in a comparative study of two groups.

4.2 Online Results

4.2.1 Hardware

Implementations and tests of the offline analysis as well as all online measurement sessions were conducted using a notebook with Windows 10 as operating system, an Intel Core i5 processor and a RAM of 8GB. No performance issues with this setup occurred during both offline analysis and recordings. The proposed BCI system therefore can be operated by most standard notebooks without need for special equipment, making it possible to transfer recordings out of a laboratory environment easily.

4.2.2 Software

The software of the proposed BCI system is based on Schwarz et al. [22] and Faller et al. [21]. Communication between the BCI system and the optimizer instance happens via TCP/IP. In this online study, both the BCI system and the optimizer were running on the same notebook in two different Matlab instances.
In the BCI online system, the classifier mode (LDA or RF) as well as the learning mode (supervised or semi-supervised) can both be changed easily in one variable each by the operator, resulting in high flexibility of the system.
Both the optimizer and the online system were tested thoroughly before the start of the measurement sessions.
The whole system setup proved to be robust and unaffected even by unforeseen computer restarts that were necessary during two recording sessions.

Optimizer

The separate optimizer instance of Schwarz et al. [22] was modified and extended to include the possibility of semi-supervised learning of the classifier. Switching between supervised and semi-supervised learning is done by a flag sent to the optimizer from the online system. Additionally, saving of the optimizer workspace was added after every close-down. This should prevent any loss of data and proved to be very useful indeed when the whole system unexpectedly had to be restarted in the middle of a measurement session. None of the already recorded data and trained

filter and classifiers were lost and the session could be continued from the point of interruption after a short time. On the other hand, the time needed for saving of the optimizer workspace increased significantly during the course of a session, due to the growing amount of stored data. Towards the end of a session, this process could take up to one to two minutes after a run. However, improved robustness and compliance of the optimizer instance surpass possible short delays of the recording process.
Since the amount of available data in a session was quite large, outlier rejection and retrainings of the filters and classifier could take more time (up to 50 s) towards the end. However, no trials were lost and communication between the online system and the optimizer worked without problems.

4.2.3 Performance

To verify the hypothesis of this work and be able to make valid comparisons, two independent groups of subjects were tested: One group was measured using the 'ground truth' supervised learning system and one group used the semi-supervised learning approach. Both groups consisted of approximately the same ratio of non-naive and naive subjects to ensure that training effects had no impact on a group's results.
The subjects themselves did not know which group they were assigned to and the overall experiment setup was the same for both groups to make results comparable.

Accuracies

The supervised group reached an overall peak accuracy of $71.88 \pm 13.26\%$, while the overall peak accuracy of the semi-supervised group was $80.03 \pm 15.90\%$. Mean overall accuracies for the two groups were $66.06 \pm 12.80\%$ and $74.69 \pm 16.04\%$, respectively, which both lies well above the calculated chance level of 54.30%. Generally, subjects in the semi-supervised group turned out to achieve better BCI control. Although differences in overall peak and mean accuracies between the two groups seem quite large, a two sample t-test showed no significant differences: This is a first indication that the starting hypothesis that the semi-supervised system would perform as good as the supervised one could be accepted.
For more elaborate testing, the separate stages' peak and mean accuracies of the two groups were compared in an ANOVA test. Again, no significant differences were found. Therefore, it can be stated that the semi-supervised learning system works as well as the standard system with supervised learning. In the semi-supervised learning approach, trials are sorted out not only by the outlier rejection, but also by the boxplot peak exclusion criterion. Therefore, less trials are included in classifier trainings, and apart from the initial calibration phase, only trials with a high information content are used. This basically corresponds to an optimization procedure of the normal supervised system.
Apparently, most trials included in the retrainings are labeled correctly, although it is still possible that data with the wrong class label passes all criteria and influences the training. However, it seems that the boxplot peak exclusion criterion for the semi-supervised learning method dismisses trials with low information content well enough for the classifier to be trained on mostly correctly classified data (additional to the true-labeled initial training data amount). This can be seen as a first step towards a real-life MI-based BCI application in which information about the true label, e.g. the user's intentions, is not available.

Furthermore, we were interested in a possible impact of the duration of the experiment on the

subjects' performances. Considering Table 3.4, it seems that accuracies generally still improve after the calibration phase, but decline again during the third evaluation period. Mean accuracy plots for the individual subjects (Figure 3.10 - 3.13) on the one hand show an increase in performance after the calibration phase for e.g. subjects A2 and B4, while other subjects, such as A9 or B9, display a higher accuracy during the calibration phase and a performance decrease in evaluation periods two and three, respectively. To evaluate statistical significance of these observations, both groups were tested separately for differences in peak and mean accuracies of the four stages of the experiment using an ANOVA test. Neither group showed statistically significant differences between any combination of two stages (six combinations in total), which confirms that all aforementioned observations are not statistically relevant. On the contrary, most subjects reported that they felt tired after their recording session and that the duration of the experiment was exhausting. Particularly subjects who did not achieve high classification accuracies described a loss of motivation over the course of the session. Apparently, fatigue was experienced subjectively by the users, but could not be found by statistical tests. Although this is a good result regarding the system's performance over an extended period of time, the users' personal feelings concerning this matter cannot be disregarded and dismissed easily.

A more detailed look at the individual subjects' results shows diverse outcomes. Not all of the subjects were able to achieve a mean classification performance better than random. Although the calculated chance level of 54.30% was outperformed by every subject for at least part of the feedback period, three of them could not achieve a better-than-random mean accuracy throughout the whole feedback period after presentation of the visual cue.
The best-performing subjects were subjects B2 and B5. Subject B2 achieved a peak and mean accuracy of 96.39% and 92.94%, respectively, while subject B5 reached a peak and mean accuracy of 98.89% and 96.95%, respectively. By contrast, subject A7 only obtained a peak and mean accuracy of 55.56% and 51.76%, respectively. The mean accuracy over the feedback period of 51.76% also clearly lies beneath chance level.
Regarding possible differences of naive and non-naive users it is evident that both factions are comprised of well-performing and badly-performing subjects. The results do not show that all non-naive subjects achieve high classification accuracies.

Looking at Figures 3.5 - 3.8 showing the accuracy curves of the two groups, it is evident that some subjects exhibit a high accuracy peak immediately after the appearance of the visual cue, but cannot maintain this accuracy level over the whole feedback period. This short-lived brain state can be induced by the visual cue and can be interpreted as a priming effect. To achieve high classification performance over a longer time, a conscious effort in motor imagery has to be made, so that longer lasting brain patterns develop [50].

When looking at Tables 3.3 and 3.4, the relatively high values of standard deviation for overall mean, peak and median accuracies in both groups stand out. For example, the standard deviation in median overall accuracies for all experiment stages is 13.12% in the supervised group and even 16.23% in the semi-supervised group. In comparison, Faller et al. [21] reported a slightly lower standard deviation of overall median accuracies of 11.3%. In Schwarz et al. [22], standard deviations for overall median accuracies of 7.3% were found, which is considerably lower than in the current study. It seems that subjects participating in the study of this work incidentally covered a broader spectrum of BCI control abilities than subjects of the studies of Faller et al. [21] and Schwarz et al. [22]. Since the performance quality of naive BCI users is unknown beforehand, these differences are unpredictable.

Retrainings

As expected and already verified in the offline analysis, an obvious decrease of times the classifier was retrained during the experiment occurred in the semi-supervised group. This is due to a decreased amount of available training data, since the exclusion criterion dismisses some trials that pass the outlier rejection. While in the supervised learning group, the CSP filters and LDA classifiers were retrained 27 times on average over the four experiment stages, re-trainings happened only 12 times on average for the semi-supervised group. This is a decrease in retrainings by approximately 55%, illustrating the strict trial dismissal of the semi-supervised exclusion criterion. As discussed for the offline results, only this strict sorting out of trials with low information content ensures that the classification performance of the system does not suffer. On the other hand, less retrainings of the classifier also induce less possibility for adaptation of the system to the user. It seems, however, that a careful selection of trials for the training data is more important for classification performance than a higher number of retrainings over the long course of the session. In addition, the initial amount of true-labeled training data is imperative.

PSD

The power spectral density describes the distribution of signal power in the frequency domain [51]. Power spectral density estimations for positions C3, Cz and C4 (Laplacian derivation) were calculated for every subject to investigate differences between the two classes hand and feet. Since event-related desynchronization and synchronization cause fluctuations of the brain signals in specific frequency ranges [8], power distributions that show more or less distinct peaks for frequencies of the MI regions (μ- and β-band) are to be expected. Results show that these peaks are linked with performance: Subjects exhibiting more pronounced peaks, such as B2 or B4 of the semi-supervised group, generally achieve a good classification performance (96.39% and 86.94% peak accuracies, respectively), which is in accordance with the findings of [52]. Moreover, a small-banded difference between power distributions for the two classes hand and feet allows the classifier to better distinguish between classes, resulting in a good performance.

ERD/ERS Maps

Time-frequency maps for both classes were calculated for every subject. In Figure 3.16 and 3.17, distinguishable brain patterns for classes hand and feet can be observed. For class hand, subject B2 shows strong narrow-banded event-related desynchronization in the μ-band frequency range particularly in position C3. This is conform with studies concerning ERD of right hand MI [8]. For class feet, it is evident that focal ERD is accompanied by ERS of the surrounding areas. Apparently, areas involved in the motor imagery are activated, while regions that are not involved in the task are deactivated simultaneously [8]. Comparing time-frequency maps of both classes for this subject shows large differences in patterns. Classification performance for subject B2 peaks at 96.39%, confirming findings that high classification performance is linked to good discriminability of brain patterns between the classes [28].
On the contrary, ERD/ERS maps of subject A4 in Figures 3.18 and 3.19 show no pronounced event-related desynchronization for any class. According to Blankertz et al. [52], BCI performance can be predicted by the distinctiveness of the power peak of SMRs. Motor imagery leads to a decrease of SMR (event-related desynchronization) and the strength of this ERD determines BCI performance. For subject A4, PSD estimations show no discernible peaks, and also event-related desynchronization is very weak, as can be seen in the ERD/ERS maps. This coincides

with a poor classification performance of 59.67% peak accuracy.

4.3 Possible Improvements

Several points for further improvements and enhancements of the current system can be addressed.
As a first step, testing the proposed semi-supervised BCI system with a smaller amount of initial training data would be of interest. Preliminary tests in the offline analysis showed a preferred initial true-labeled training data amount of 30 to 40 TPC. For this online study, 40 TPC were recorded in the calibration phase. A reduction of this amount to 30 TPC, or even less, would decrease the duration of the calibration phase considerably by at least one run. This would be an advantage for studies with shorter session times, as in most common BCI studies.
In this study, all recording sessions were conducted in a shielded laboratory environment. Any real-life BCI applications, however, would not be used in such an environment. Therefore, it is important to transfer BCI use out of the laboratory to test it under real-life circumstances. Due to the small setup and robust and simple system operation of the proposed BCI system, this could be done quite easily.
Furthermore, subjects reported a definite increase in motivation for the task when receiving feedback. The time period in which the subject does not receive any feedback yet was rather short, which was pointed out as advantageous and convenient by several subjects.
Contrary to this, many subjects described increasing monotony and dullness of the experiment over the course of the session. Some amount of fatigue and decreasing concentration was to be expected for such a long experiment. However, this phenomenon was amplified by the nature of the paradigm, as some subjects declared. A more interesting and engaging paradigm could help to keep the motivation and concentration of users high even in such a long session. Moreover, a broader variety of break activities, e.g. reading, playing a computer game or having a conversation, instead of only watching TV, could also provide more distraction and diversion for the users.
Another point is the selection of the semi-supervised exclusion criteria. Only a small amount of possible options were tested in this work. Different methods could still be found and evaluated for further performance boosts.
For real-life applications, a change of the two used classes in the system could be advantageous: Instead of two motor imagery classes, one motor imagery class and a rest condition could be introduced, making it possible for the system to distinguish between a default non-action condition and an action condition indicating activation of the application. In this regard, EEG data recorded during the breaks of the online study of this work could be analysed in a first step to evaluate classification of idle background brain patterns and possibly introduce a classification threshold that has to be exceeded in order to trigger some action.

5 Conclusion

The goal of this thesis was to find an approach to implement a semi-supervised learning algorithm in an adaptive online MI-based BCI system. The hypothesis to be tested was that such a system using semi-supervised learning works as good as a standard supervised BCI.
Several different methods for selecting training trials with a high information content for the classifier were tested offline. The boxplot peak method presented promising results and was therefore implemented in an online adaptive BCI. To evaluate online performance, two groups of subjects were tested in an online study: One group was examined using a standard supervised adaptive system, while the semi-supervised approach was analyzed with the other group. Results of the two groups show that the starting hypothesis can be accepted: There were no significant differences in mean or peak accuracies between the two groups in any stage of the experiment ($F_{3,54} = 2.058$, $p > .05$, $F_{3,54} = 1.22$, $p > .05$). This outcome confirms that employing semi-supervised learning in an adaptive BCI works efficiently, allowing for further testing of BCI systems where the user's intent is not known to the system.
Due to the long duration of the experiment, subjects reported increased exhaustion. However, no significant performance drops could be found. Still, options on how to keep the subject's motivation high throughout the experiment should be considered, such as starting feedback even earlier or changing activities during the breaks.
All in all, results of this thesis are a full success and can benefit further developments of real-life BCI applications for severely impaired users.

Bibliography

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, Jun. 2002.

[2] J. J. Vidal, "Toward direct brain-computer communication," *Annu. Rev. Biophys. Bioeng.*, vol. 2, pp. 157–180, 1973.

[3] J. R. Wolpaw, G. E. Loeb, B. Z. Allison, E. Donchin, O. F. do Nascimento, W. J. Heetderks, F. Nijboer, W. G. Shain, and J. N. Turner, "BCI meeting 2005–workshop on signals and recording methods," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 138–141, Jun. 2006.

[4] E. M. Maynard, C. T. Nordhausen, and R. A. Normann, "The utah intracortical electrode array: a recording structure for potential brain-computer interfaces," *Electroencephalogr. Clin. Neurophysiol.*, vol. 102, no. 3, pp. 228–239, Mar. 1997.

[5] P. L. Nunez and R. Srinivasan, *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, USA, Jan. 2006.

[6] E. Niedermeyer and F. H. L. da Silva, *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.

[7] C. Brunner, "Brain-computer interfaces: feature selection of spatially filtered data and phase synchronization," Ph.D. dissertation, Graz University of Technology, 2007.

[8] G. Pfurtscheller and F. H. Lopes da Silva, "Event-related EEG/MEG synchronization and desynchronization: basic principles," *Clin. Neurophysiol.*, vol. 110, no. 11, pp. 1842–1857, Nov. 1999.

[9] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proc. IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.

[10] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," *Neurosci. Lett.*, vol. 239, no. 2-3, pp. 65–68, 1997.

[11] "Somatic sensory and somatic motor maps in the cerebral cortex," http://higheredbcs.wiley.com/legacy/college/tortora/0470565101/hearthis_ill/pap13e_ch16_illustr_audio_mp3_am/simulations/hear/maps.html, 2011, accessed: 2017-4-10.

[12] M. Naeem, "Spatio-temporal decomposition of bioelectrical brain signals," Ph.D. dissertation, Graz University of Technology, 2008.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, 17 Aug. 2006.

[14] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[15] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "EEG-based discrimination between imagination of right and left hand movement," *Electroencephalogr. Clin. Neurophysiol.*, vol. 103, no. 6, pp. 642–651, 1997.

[16] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, pp. R1–R13, 2007.

[17] P. Shenoy, M. Krauledat, B. Blankertz, R. P. N. Rao, and K.-R. Müller, "Towards adaptive classification for BCI," *J. Neural Eng.*, vol. 3, no. 1, pp. R13–23, Mar. 2006.

[18] Isaksson, A., Wennberg, A., & Zetterberg, L.H., "Computer analysis of EEG signals with parametric models," *Proc. IEEE*, vol. 69, no. 4, pp. 451–461, 1981.

[19] C. Vidaurre, C. Sannelli, K.-R. Müller, and B. Blankertz, "Machine-learning-based coadaptive calibration for brain-computer interfaces," *Neural Comput.*, vol. 23, no. 3, pp. 791–816, Mar. 2011.

[20] C. Vidaurre, C. Sannelli, K.-R. Müller, and B. Blankertz, "Co-adaptive calibration to improve BCI efficiency," *J. Neural Eng.*, vol. 8, no. 2, p. 025009, Apr. 2011.

[21] J. Faller, C. Vidaurre, T. Solis-Escalante, C. Neuper, and R. Scherer, "Autocalibration and recurrent adaptation: towards a plug and play online ERD-BCI," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 3, pp. 313–319, May 2012.

[22] A. Schwarz, R. Scherer, D. Steyrl, J. Faller, and G. R. Muller-Putz, "A co-adaptive sensory motor rhythms Brain-Computer interface based on common spatial patterns and random forest," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2015, pp. 1049–1052, Aug. 2015.

[23] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning.* MIT Press (MA), 2010.

[24] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R.* Springer Science & Business Media, 24 Jun. 2013.

[25] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1285–1294, 2008.

[26] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) algorithm using online adaptive and semi-supervised learning," in *The 2011 International Joint Conference on Neural Networks*, 2011.

[27] L. F. Nicolas-Alonso, R. Corralejo, J. Gomez-Pilar, D. Álvarez, and R. Hornero, "Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces," *Neurocomputing*, vol. 159, pp. 186–196, 2015.

[28] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 441–446, Dec. 2000.

[29] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, 22 Oct. 2013.

[30] Yijun Wang, Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selelction in motor imagery based brain-computer interface," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005.

[31] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller, "Optimizing spatial filters for robust EEG Single-Trial analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, 2008.

[32] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K.-R. Muller, "Combined optimization of spatial and temporal filters for improving Brain-Computer interfacing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2274–2281, 2006.

[33] S. Lemm, B. Blankertz, G. Curio, and K.-R. Muller, "Spatio-Spectral filters for improving the classification of single trial EEG," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 9, pp. 1541–1548, 2005.

[34] Q. Novi, C. Guan, T. H. Dat, and P. Xue, "Sub-band common spatial pattern (SBCSP) for Brain-Computer interface," in *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, 2007.

[35] Kai Keng Ang, K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (FBCSP) in Brain-Computer interface," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008.

[36] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning.* Springer, Berlin: Springer series in statistics, 2001, vol. 1.

[37] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components–a tutorial," *Neuroimage*, vol. 56, no. 2, pp. 814–825, 15 May 2011.

[38] D. Steyrl, R. Scherer, J. Faller, and G. R. Müller-Putz, "Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier," *Biomed. Tech.*, vol. 61, no. 1, pp. 77–86, Feb. 2016.

[39] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "EMG and EOG artifacts in brain computer interface systems: A survey," *Clin. Neurophysiol.*, vol. 118, no. 3, pp. 480–494, Mar. 2007.

[40] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, 15 Mar. 2004.

[41] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *Neuroimage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[42] C. Vidaurre and B. Blankertz, "Towards a cure for BCI illiteracy," *Brain Topogr.*, vol. 23, no. 2, pp. 194–198, Jun. 2010.

[43] Rydesaeter, P, "TCP/UDP/IP toolbox 2.0.6, pnet," https://de.mathworks.com/matlabcentral/fileexchange/345-tcp-udp-ip-toolbox-2-0-6/content/tcp_udp_ip/pnet.m, 2008, accessed: 2017-4-10.

[44] A. L. Abhishek Jaiantilal, "Random forest matlab implementation," https://code.google.com/archive/p/randomforest-matlab/downloads, 2009, accessed: 2017-4-10.

[45] C. Kothe, "Fast Serialize/Deserialize," https://de.mathworks.com/matlabcentral/fileexchange/34564-fast-serialize-deserialize, 2012, accessed: 2017-4-10.

[46] C. Breitwieser, C. Neuper, and G. R. Muller-Putz, "A concept to standardize raw biosignal transmission for brain-computer interfaces," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.

[47] G. R. Mueller-Putz, R. Scherer, D. Steyrl and J. Faller, "GRAZ BCI libraries," https://www.tugraz.at/institute/ine/research/software/, 2014, accessed: 2017-4-10.

[48] H. H. Jasper, "Report of the committee on methods of clinical examination in electroencephalography: 1957," *Electroencephalogr. Clin. Neurophysiol.*, vol. 10, no. 2, pp. 370–375, May 1958.

[49] "International 10-20 electrode system setup," http://www.fieldtriptoolbox.org/_media/template/easycapm11.png, accessed: 2017-4-10.

[50] G. Pfurtscheller, R. Scherer, G. R. Müller-Putz, and F. H. Lopes da Silva, "Short-lived brain state after cued motor imagery in naive subjects," *Eur. J. Neurosci.*, vol. 28, no. 7, pp. 1419–1426, Oct. 2008.

[51] P. Herman, G. Prasad, T. M. McGinnity, and D. Coyle, "Comparative analysis of spectral approaches to feature extraction for EEG-based motor imagery classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 4, pp. 317–326, Aug. 2008.

[52] B. Blankertz, C. Sannelli, S. Halder, E. M. Hammer, A. Kübler, K.-R. Müller, G. Curio, and T. Dickhaus, "Neurophysiological predictor of SMR-based BCI performance," *Neuroimage*, vol. 51, no. 4, pp. 1303–1309, 15 Jul. 2010.

A Appendix

A.1 Online System - Implementation

In the following graphs, the implementation of the online system and some important components are displayed. Critical code parts are integrated as s-functions into the model. The interface to the optimizer is ddepicted in figure A.1.

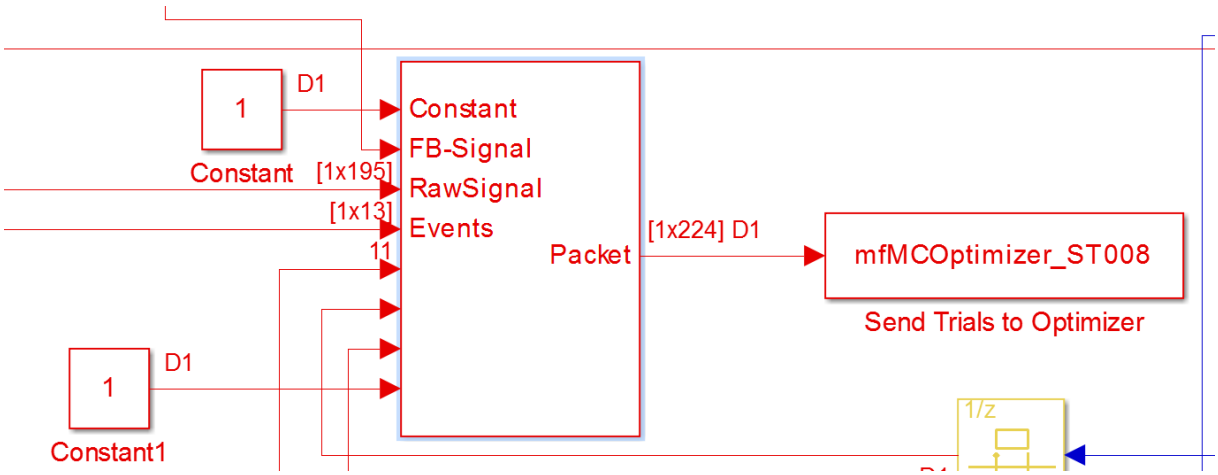
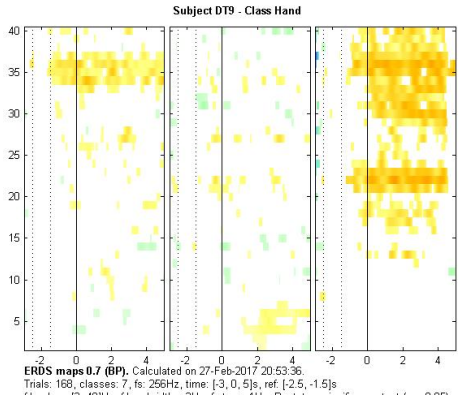


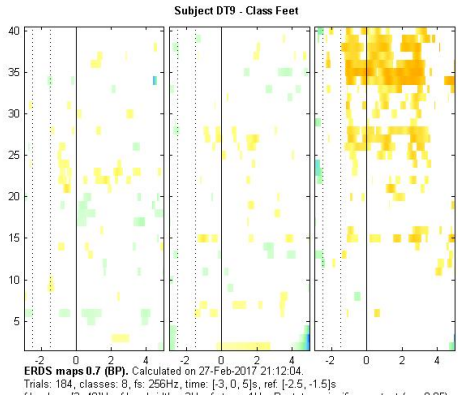
Figure A.1: Ths interface that connects the online system with the optimizer.

A.2 ERD/ERS Maps

The following figures show ERD/ERS maps for class hand and feet for all subjects apart from B2 and A4, which were already included in the Results section.

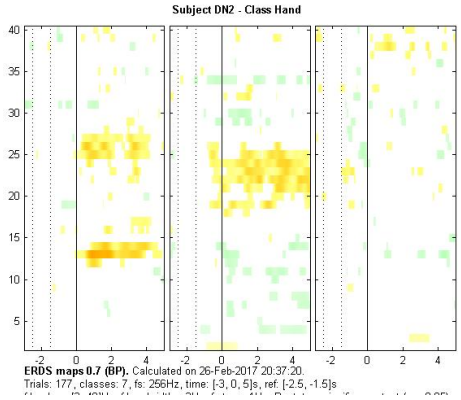


(a) ERD/ERS Hand

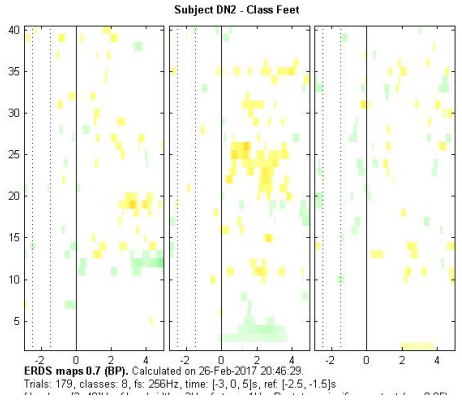


(b) ERD/ERS Feet

Figure A.2: ERD/ERS maps of subject A1.

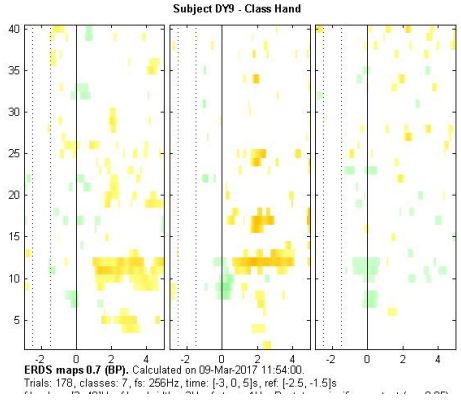


(a) ERD/ERS Hand

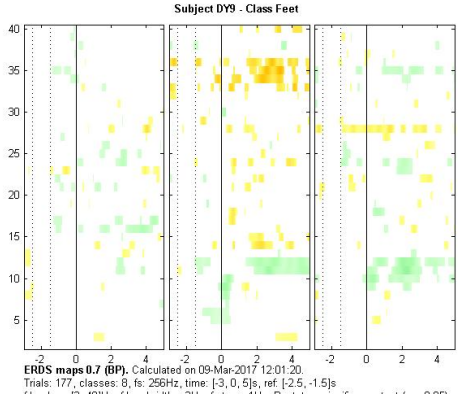


(b) ERD/ERS Feet

Figure A.3: ERD/ERS maps of subject A2.

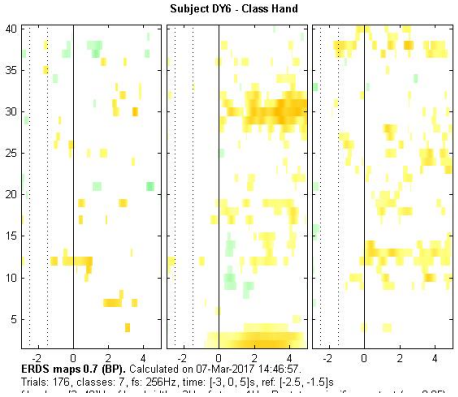


(a) ERD/ERS Hand

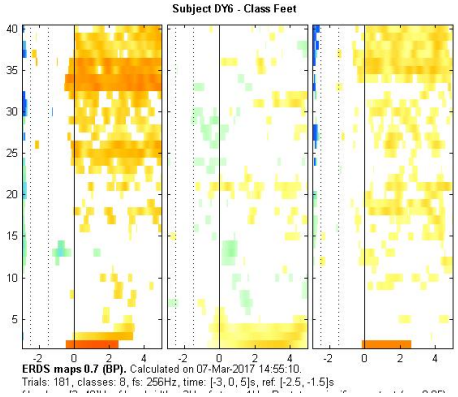


(b) ERD/ERS Feet

Figure A.4: ERD/ERS maps of subject A3.

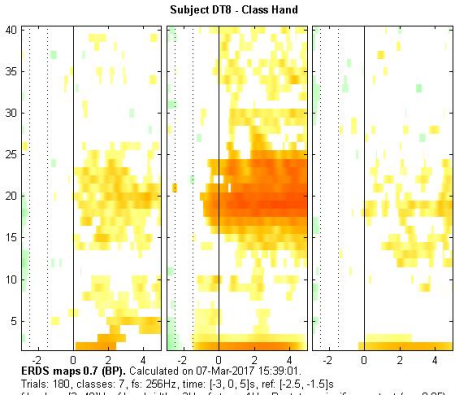


(a) ERD/ERS Hand

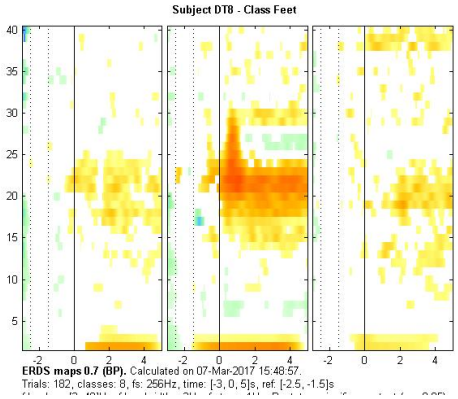


(b) ERD/ERS Feet

Figure A.5: ERD/ERS maps of subject A5.

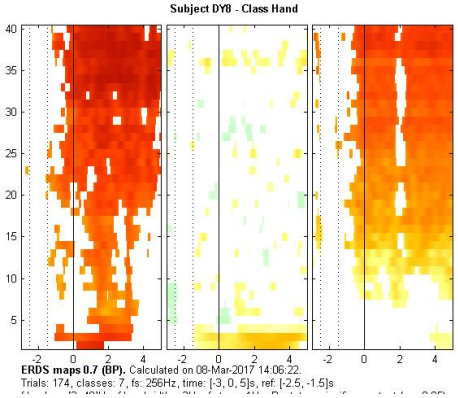


(a) ERD/ERS Hand

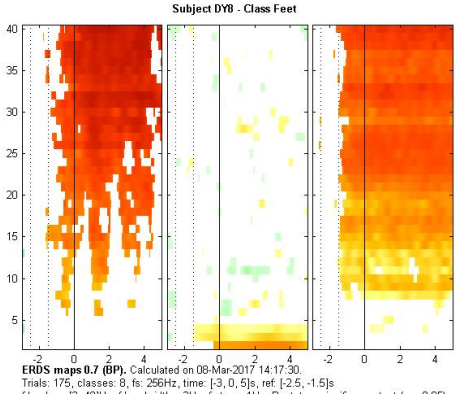


(b) ERD/ERS Feet

Figure A.6: ERD/ERS maps of subject A6.

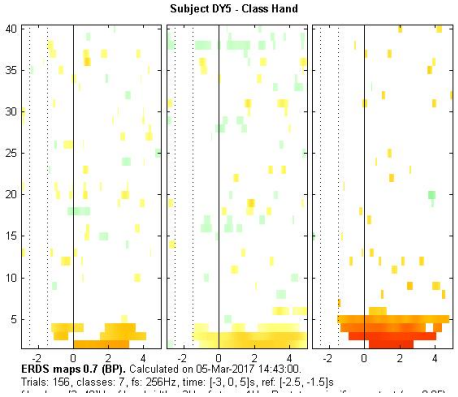


(a) ERD/ERS Hand

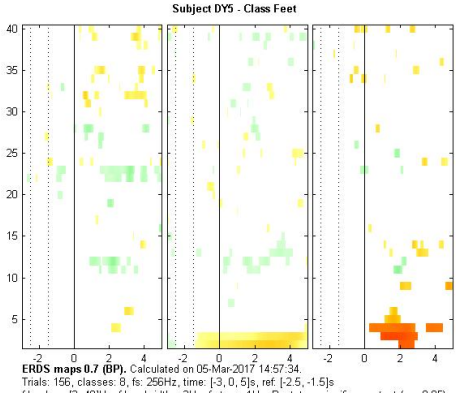


(b) ERD/ERS Feet

Figure A.7: ERD/ERS maps of subject A7.

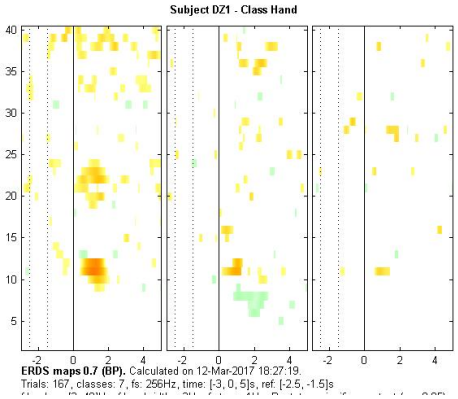


(a) ERD/ERS Hand

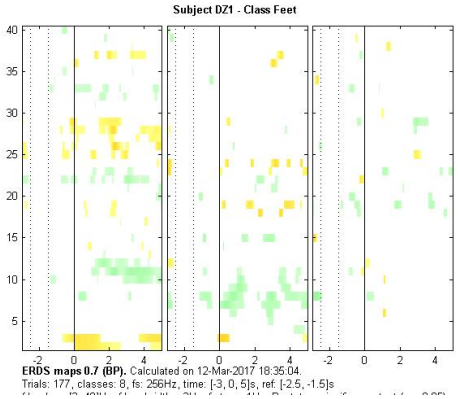


(b) ERD/ERS Feet

Figure A.8: ERD/ERS maps of subject A8.

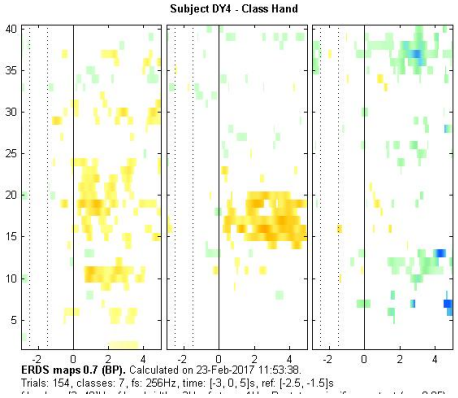


(a) ERD/ERS Hand

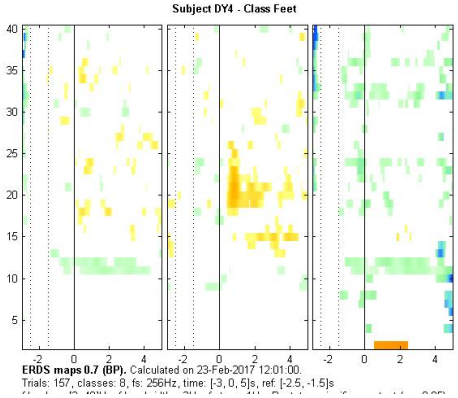


(b) ERD/ERS Feet

Figure A.9: ERD/ERS maps of subject A9.

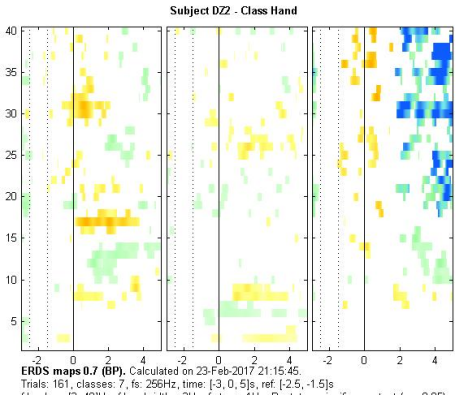


(a) ERD/ERS Hand

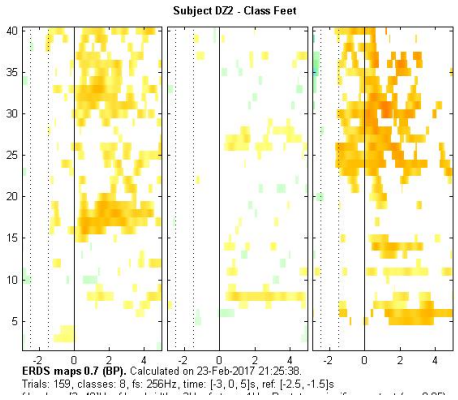


(b) ERD/ERS Feet

Figure A.10: ERD/ERS maps of subject A10.

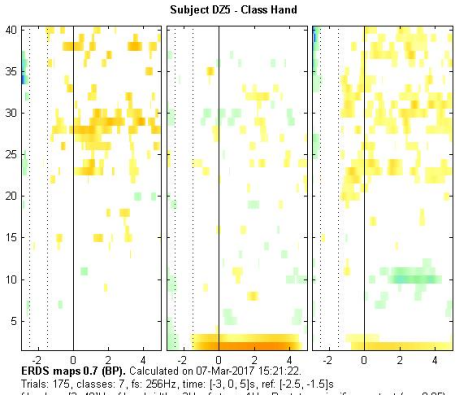


(a) ERD/ERS Hand

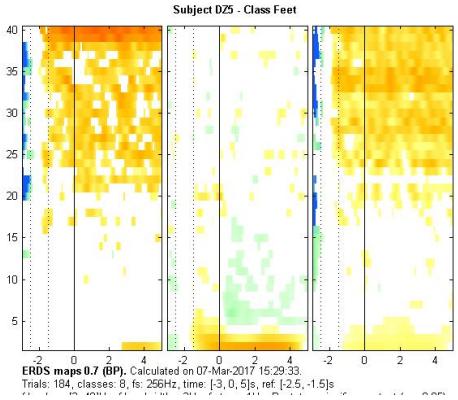


(b) ERD/ERS Feet

Figure A.11: ERD/ERS maps of subject B1.

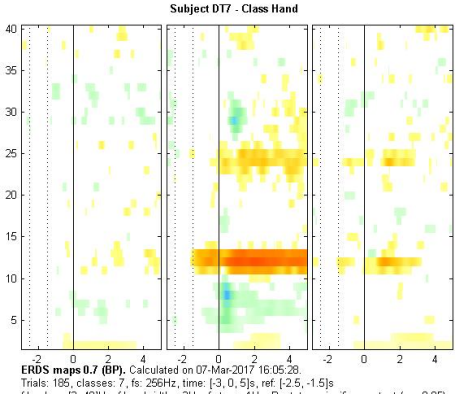


(a) ERD/ERS Hand

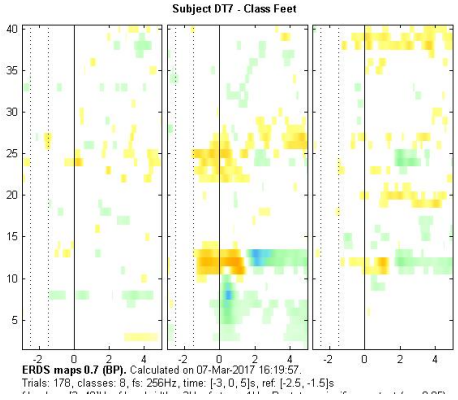


(b) ERD/ERS Feet

Figure A.12: ERD/ERS maps of subject B3.

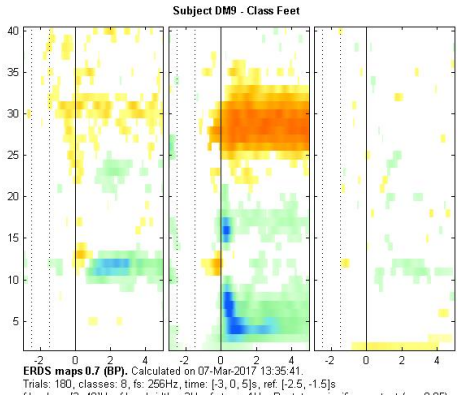


(a) ERD/ERS Hand

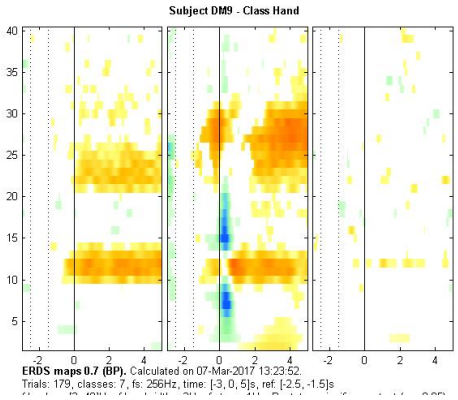


(b) ERD/ERS Feet

Figure A.13: ERD/ERS maps of subject B4.

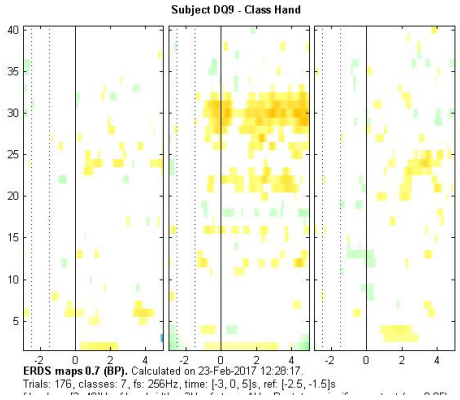


(a) ERD/ERS Hand

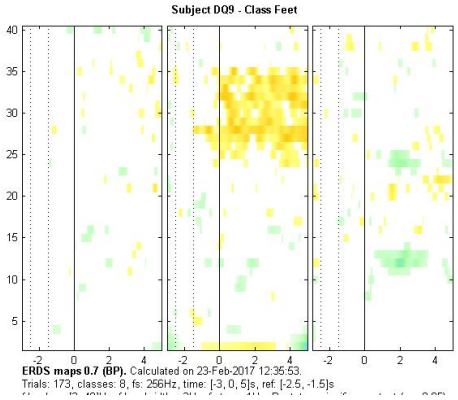


(b) ERD/ERS Feet

Figure A.14: ERD/ERS maps of subject B5.

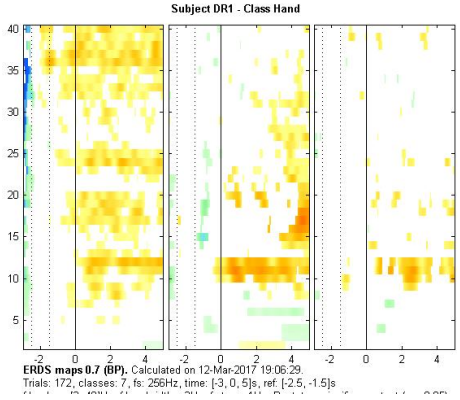


(a) ERD/ERS Hand

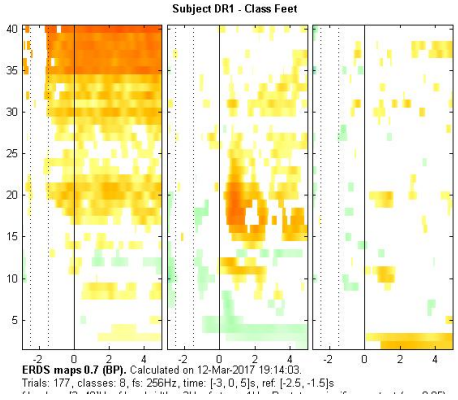


(b) ERD/ERS Feet

Figure A.15: ERD/ERS maps of subject B6.

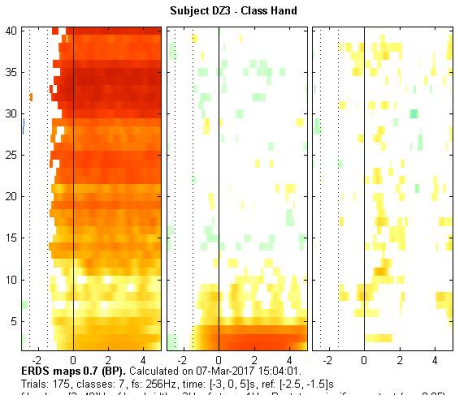


(a) ERD/ERS Hand

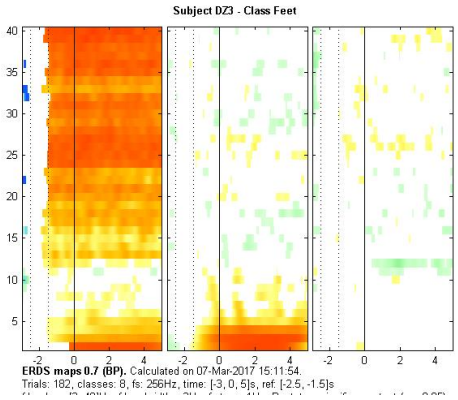


(b) ERD/ERS Feet

Figure A.16: ERD/ERS maps of subject B7.

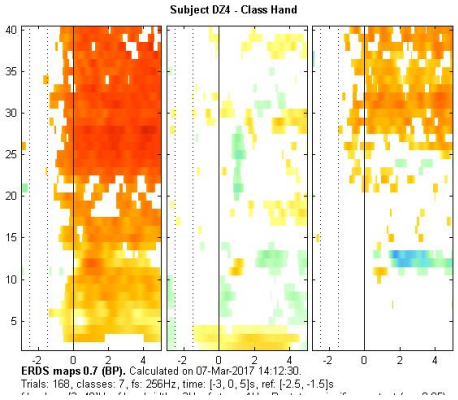


(a) ERD/ERS Hand

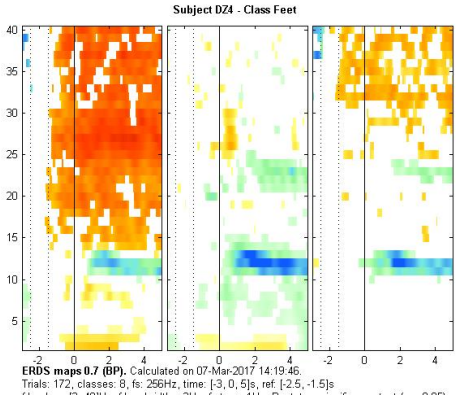


(b) ERD/ERS Feet

Figure A.17: ERD/ERS maps of subject B8.

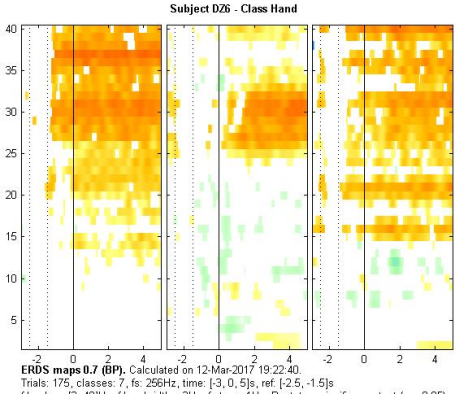


(a) ERD/ERS Hand

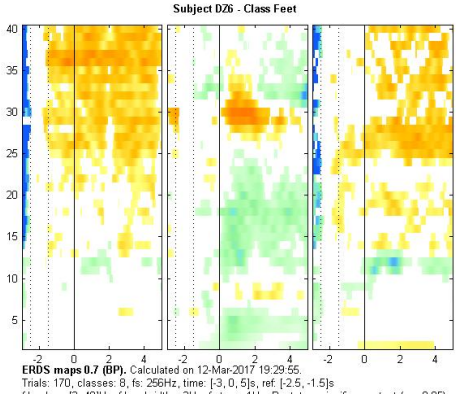


(b) ERD/ERS Feet

Figure A.18: ERD/ERS maps of subject B9.



(a) ERD/ERS Hand



(b) ERD/ERS Feet

Figure A.19: ERD/ERS maps of subject B10.