Nitikorn Srisrisawang, BEng.

Attenuating effects from intra-session non-stationarities in a multiclass oscillatory Brain-Computer Interface

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

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

Institute for Neural Engineering

Co-supervisor

Dipl.-Ing. Reinmar Kobler, BSc.

Graz, February 2020

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

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.

Datum / Date

Unterschrift / Signature

Acknowledgements

I would like to thank all of the members of the MIRAGE91 Graz BCI racing team for providing equipment, valuable advice, and moral support, especially our former team member, as well as my co-advisor, Reinmar Kobler, who guided me all the way to the end of this thesis. Furthermore, I would like to thank my supervisor, Dr. Gernot Müller-Putz for giving me an opportunity to do this thesis.

I also want to thank Andreas Schwarz and Julia Brandstetter for providing me with a dataset. Without the dataset, it would have taken me some more weeks to record a dataset.

I would like to thank the Royal Thai government for the financial support as well as the Royal Thai Embassy in Vienna and the Office of the Civil Service Commission.

Finally, I want to thank my family and all of my friends who give me mental support during the up, and down, of my Master's degree study.

Abstract

EEG signals that are typically utilized in a brain-computer interface (BCI), can be subject to non-stationary changes over time which degrade the performance of the BCI. This thesis focuses on oscillatory-based BCIs, and compares algorithms that can reduce the effect of intra-session (within the same session) non-stationarities. The thesis specifically considers online BCI operation where, for new data, no class label information is available. Hence, the calibration data recorded at the beginning of a session is the only labeled data.

In an offline analysis, four algorithms were compared with regard to their performance within the same session. One algorithm was a **standard** classification model, consisting of shrinkage common spatial pattern (sCSP), and shrinkage linear discriminant analysis (sLDA). This was the reference model which did not address non-stationarities. The other three algorithms, which address non-stationarities, were adaptive normalization LDA (**AdLDA**), Importance-weighted LDA (**IWLDA**), and within-session divergence-based CSP (**divCSP-WS**). Furthermore, their combinations, which were **AdLDA**+**divCSP-WS** and **IWLDA**+**divCSP-WS**, were also considered.

The offline simulation with a 2-class oscillatory-based BCI dataset (right hand and feet motor imagery) revealed that **AdLDA** showed a significant improvement in the accuracy over the **standard** algorithm (mean accuracy **AdLDA** 77.51%, **standard** 76.07%). Based on this finding, **AdLDA** was used in a 3-class oscillatory-based online BCI experiment (right hand motor imagery, mental subtraction, and mental rotation). In the online experiment, an additional factor was introduced to induce a change of the state of mind by competing with a computer-generated rival player. The analysis indicated that **AdLDA** still significantly outperformed the **standard** algorithm. (mean accuracy **AdLDA** 55.64%, **standard** 53.20%).

In conclusion, **AdLDA** exhibited improvement over the **standard** algorithm by 1.44% in 2-class BCI and 2.44% in the 3-class BCI which indicated that the rival factor marginally influenced the accuracy.

Keywords: brain-computer interface (BCI), intra-session, non-stationarities, electroencephalogram (EEG), common spatial pattern (CSP), linear discriminant analysis (LDA)

Kurzfassung

EEG-Signale, die typischerweise in einer Gehirn-Computer Schnittstelle – im Englischen Brain-Computer-Interface (BCI) – verwendet werden, können im Laufe der Zeit nichtstationären Änderungen unterliegen. Diese Nichtstationäritäten können die Leistung des BCIs verschlechtern. Diese Arbeit konzentriert sich auf oszillatorische BCIs und vergleicht Algorithmen, die den Effekt von intra-session (innerhalb der gleichen Sitzung) Nichtstationäritäten reduzieren können. Die Arbeit betrachtet insbesondere die Anwendung eines BCIs, bei dem für neue Daten keine Klassenlabel verfügbar sind. Daher sind die zu Beginn einer Sitzung aufgezeichneten Kalibrierungsdaten die einzigen Daten mit Labels.

In einer simulierten Experiment wurden vier Algorithmen hinsichtlich ihrer Leistungsfähigkeit innerhalb derselben Sitzung verglichen. Ein Algorithmus war ein **Standard** Klassifikationsmodell, bestehend aus shrinkage common spatial patterns (sCSP) und shrinkage Linear Discriminant Analysis (sLDA). Dieser Referenzalgorithmus ignorierte die Nichtstationäritäten. Die anderen drei Algorithmen, die Nichtstationäritäten berücksichtigten, waren adaptive normalized LDA (**AdLDA**), importance weighted LDA (**IWLDA**) und within-session divergence based CSP (**divCSP-WS**). Darüber hinaus wurden auch die Kombinationen, **AdLDA+divCSP-WS** und **IWLDA+divCSP-WS** berücksichtigt. Die Ergebnisse mit einem 2-Klassen Datensatz (Bewegungsvorstellung der rechten Hand oder beider Füße) ergab, eine signifikante Verbesserung der Klassifikationsgenauigkeit von AdLDA gegenüber dem Referenzalgorithmus (mittlere Genauigkeit **AdLDA** 77,51%, **Standard** 76,07%). Basierend auf diesem Ergebnis wurde AdLDA in einem 3-Klassen Online-Experiment (Bewegungsvorstellung der rechten Hand, mentale Subtraktion und mentale Rotation) verwendet. In dem Online-Experiment wurde ein zusätzlicher Faktor eingeführt, um eine Veränderung des mentalen Zustands zu induzieren. Dieser Faktor stellte einen simulierten Gegner dar. Die Ergebnisse zeigen, dass AdLDA den Referenzalgorithmus immer noch signifikant übertraf. (mittlere Genauigkeit AdLDA 55,64%, **Standard** 53,20%).

Zusammenfassend zeigte **AdLDA** eine geringe, aber signifikante Verbesserung gegenüber dem **Standard** Algorithmus um 1,44% im 2-Klassen-BCI und 2,44% im 3-Klassen-BCI, was darauf hinweist, dass der simulierte Gegner die Klassifikationsgenauigkeit nur geringfügig beeinflusst hat.

Schlüsselwörter: Gehirn-Computer Schnittstelle (BCI), intra-session, nichtstationären, elektroenzephalogram (EEG), CSP, LDA

Contents

Chapter 1

Introduction

1.1 Background

In this section, the background and scope of the thesis are established. Important concepts such as the term Electroencephalogram (EEG) as well as the Brain-Computer Interface (BCI) system are introduced and explained. Subsequently, non-stationarities affecting the EEG are described.

1.1.1 Electroencephalogram (EEG)

Back in 1929, Hans Berger was the first who was able to measure the electrical fluctuations from a human's brain at the scalp [1]. He found the alpha rhythm, which is an oscillation at around 12 Hz in the recorded EEG. Later on, the EEG was divided into different frequency bands, namely, (the exact definition of the frequency ranges varies in the literature)

- Delta band: 0.5 to 4 Hz
- Theta band: 4 to 8 Hz
- Alpha band: 8 to 13 Hz
- Beta band: 13 to 30 Hz

- Gamma band: from 30 Hz and higher

EEG has become an essential tool in many fields such diagnosis of epilepsy in medicine [2] or in behavioral studies in psychology [3], as well as in the BCI field. This is because it is a non-invasive method of measuring the electrical voltage and it is cheap compared to other methods. However, there are several other non-invasive options to measure brain activity, such as magnetoencephalography (MEG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), Near-infrared spectroscopy (NIRS) [4], [5]. There are also invasive options [5], such as multi unit activity which is a highly invasive method, and Electrocorticography (ECoG) which is a minimally invasive method. Both methods require a medical surgery to implant a sensing device underneath the skull of the user, which could be risky for the user. Despite the different methods of recording brain activity, the recording of these brain signals has to be handled carefully as the signal-to-noise ratio (SNR) is low in these signals.

However, each method has its trade-off between temporal resolution and spatial resolution [4], [5]. For example, fMRI might have an excellent spatial resolution (in millimeters) but its temporal resolution is low (seconds), not to mention the immobility of the measuring device compared to EEG. On the other hand, EEG has a good temporal resolution (milliseconds) but poor spatial resolution (centimeters), while the mobility of the user is good. Especially, when the technology allows a wireless connection between a computer and the EEG recording device [6].

1.1.2 Brain-Computer Interface (BCI)

The concept of using the EEG to communicate with a machine emerged around 40 years after Berger's discovery. The term BCI was introduced by Vidal [7] in 1973 in which he explained the possibility of using EEG for direct brain-computer communication.

A BCI can be described as a system that allows users to communicate with the external world without relying on the normal pathways of the brain through muscles or the peripheral nervous system [8]. Figure 1.1 visualizes the components of a

BCI. The BCI system works by first, signal acquisition, where the EEG signals are measured from the user, and then, signal processing is applied to the EEG signals. Feature extraction is done to the processed EEG signals and then used in the classification to get control signals that can be used in an application. Typical applications are word spelling [9], [10], gaming [11], or robotic arm control [12]. Finally, to close this loop, the applications present the feedback to the user.

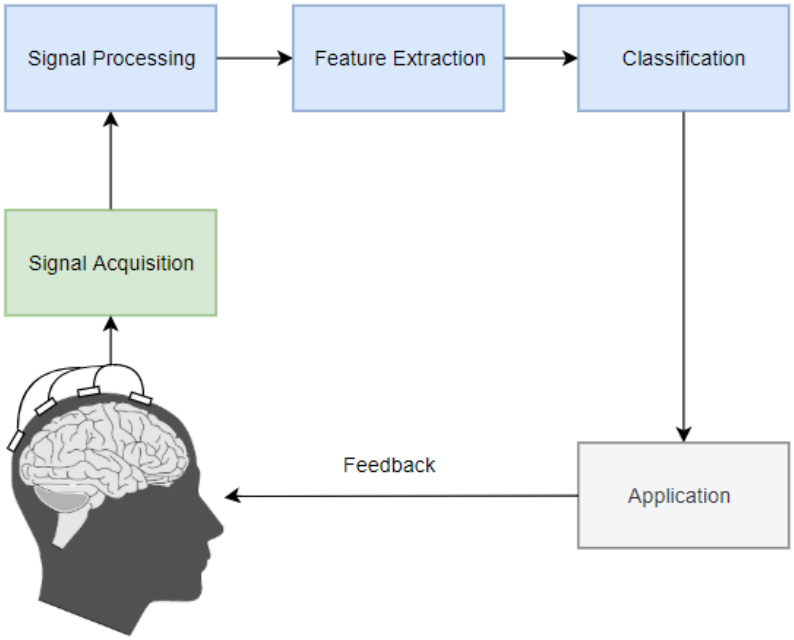


Figure 1.1: The overview of a closed-loop BCI. The signals recorded from the user are processed and the output signals are used to control an application that gives feedback to the user.

Several strategies can be used to control a BCI [5]. The two most common strategies are stimulus-evoked activity and mental imagery. The former usually begins with presenting stimuli to the user where one has to focus on stimuli that differ from the other. An example would be the scenario where several pictures of non-target stimuli are flashed in front of the user, but occasionally, there would be pictures of target in between, which the user has to count the number of occurrences of the

target stimulus. The BCI then processes and classifies the signal to distinguish the difference between the time when the target stimuli are presented and when the non-target stimuli are presented. This type of experimental paradigm is called an oddball paradigm [13].

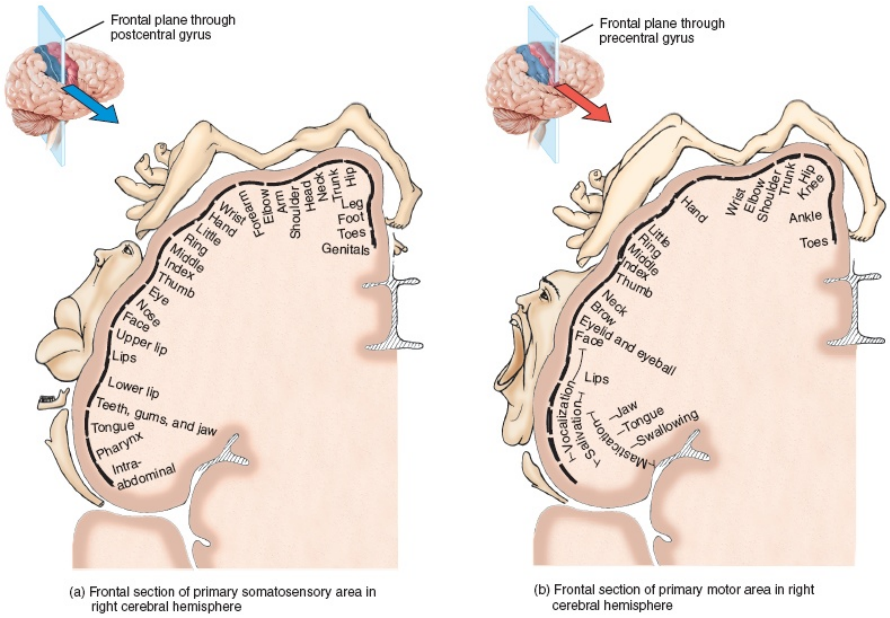


Figure 1.2: The homunculus map of the primary sensorimotor areas (left) the sensory mapping, (right) the motor mapping. Figure taken from [14].

A mental imagery BCI, however, relies on sensorimotor rhythms (SMRs) of the brain. The SMRs are oscillations of the brain signals of specific frequency bands that can be acquired over the sensorimotor areas of the brain [15]. The sketches in Figure 1.2 illustrate how different parts of the body are mapped to the sensorimotor areas. In the case when one performs a movement of a specific limb, the power of the oscillations change in the associated cortical patch which is specifically called motor imagery. Then, electrodes in the vicinity of the corresponding cortical patch can register the changes of the power of the oscillations. This was discovered by Pfurtscheller in 1977, where he called the SMR power changes as event-related de/synchronization (ERDS) [16]–[19]. ERDS effects are not only measurable in an actual movement but also in an imagined movement. Because of this, motor imagery became the main

interest in BCI research as it can also be used in people without an ability to move a limb. However, the key to successful motor imagery is to do a kinesthetic motor imagery, meaning to imagine the feeling when doing that movement, rather than a visual motor imagery, meaning to imagine observing the movement [20], [21].

There are also other mental tasks that can be differentiate by power changes of different brain rhythms but the location of the changes is non-specific and can vary between persons. The tasks that can be used in the experiment beside the hand and feet motor imagery are [22]

- word association: thinking of words that begin with a specific letter
- mental subtraction: subtracting two numbers consecutively
- mental rotation: visualizing an object rotating
- auditory imagery: singing a song in the head
- spatial navigation: picturing a walk in a familiar place

Friedrich et al. [22] chose 4 classes that indicated the best accuracy for the individual participant. It was shown that hand motor imagery was the most occurrence among the participants following by one "brain-teaser" task (either word association or mental subtraction), mental rotation task, and lastly, one of these imageries (auditory imagery, spatial navigation, and feet motor imagery). The tasks chosen in this thesis is based on the finding of this paper.

1.1.3 Non-stationarities in BCI

The stationarity of a time series can be described as when the probability distribution of the time series does not change over time [23]. In contrast, a time series is non-stationary when the probability distribution changes over time.

In the case of an EEG-based BCI, the EEG signal does not only have a low SNR but is also highly non-stationary [23]–[25]. The factors influencing the non-stationarities are unclear but it is typically divided into psychophysiological and

technical factors. The psychophysiological factors are changes in the user's attention, mood, motivation, and others [23]. The technical factor is the interface between the scalp and an electrode. For example, considering a scenario within the same session, the conductive gel, that is used to keep the impedance between the scalp and the electrodes low, can dry up causing the impedance to vary over time, and if one considers a scenario between sessions, the position of electrodes might also change which eventually cause the signals to vary between session. The combination of all these makes the usage of BCIs in the real world challenging, as currently, we do not fully understand the interaction between different factors.

1.2 Motivation and Aims

The non-stationarities can affect the BCI in general, but in this thesis, the use case within the MIRAGE91 Graz BCI racing team is considered. The MIRAGE91 Graz BCI racing team [26] is a student team that was founded to participate in the CYBATHLON event [27]. The CYBATHLON event is a competition between teams from different countries where people with disabilities (pilots) use a system provided by the team so they can perform daily tasks. There are in total six disciplinaries within this competition, namely; BCI Race, Functional Electrical Stimulation (FES) Bike Race, Powered Arm Prosthesis Race, Powered Leg Prosthesis Race, Powered Exoskeleton Race, and Powered Wheelchair Race.

In the BCI race, the pilot has to use a BCI to control a game with his/her brain signals only. The Graz BCI racing team participated in the CYBATHLON competition in 2016, and observed that the performance of the team's pilot worsened during the competition with a run time of the race of 196 seconds, compared to run times of 143 to 178 seconds during the training session. In an offline analysis [11], the authors identified that the degradation of the system's performance was due to non-stationarities affecting the EEG signal.

The main interest of this thesis concerns a scenario the BCI racing competition in the event. Figure 1.3 compares between the classical BCI scenario, where the temporal gap between a calibration block and operation blocks is short, while in

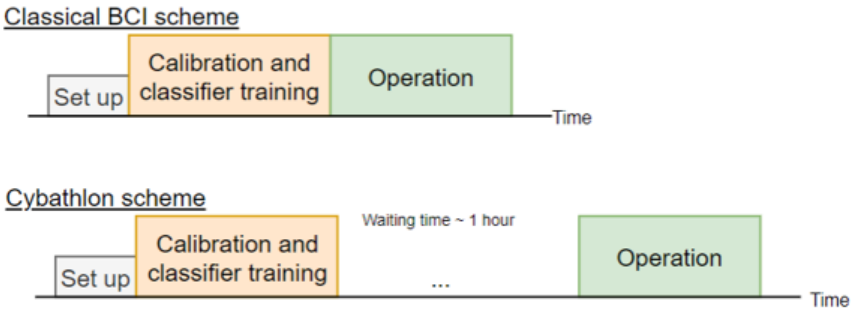


Figure 1.3: Comparison between the classical BCI usage to the usage. The difference is the temporal gap between calibration and operation blocks.

the scenario, the team and the pilot have to wait some time until the competition begins. The temporal gap and a mental state change in the pilot could cause non-stationarities in the EEG and degrade the BCI performance.

The specific goal of this thesis is to first build a new system for the Graz BCI racing team, since the team was offered new recording hardware that was not compatible with the predecessor system. The BCI uses a standard classification model, that consists of shrinkage linear discriminant analysis (sLDA) and shrinkage common spatial pattern (sCSP). The other goals are to investigate algorithms that reduce the effect of within-session non-stationarities and to implement and evaluate an improved BCI in an online experiment.

To do this, three algorithms were investigated, namely, Adaptive normalized LDA (**AdLDA**), Importance-weighted LDA (**IWLDA**), and within-session divergence-based CSP (**divCSP-WS**). The dataset from [28] was used to determine the best algorithm in terms of classification accuracy. The best algorithm was then used to test the new system in an online experiment, where we tried to replicate the scenario.

Chapter 2

Related Work

2.1 Graz BCI Racing team predecessor system

In this section, the description of the predecessor system that was used in the VYBATHLON event in 2016, is given. It is essential to mention the predecessor system since parts of the current system are based on them. The explanation of the system was taken partly from the Master's Thesis of a Graz BCI Racing team member, Brandstetter [28].

The central part of the system is the software called tools for BCI (TOBI) SignalServer [29], which is a data acquisition software that provides biosignals in a standardized protocol in real time. The BCI was implemented in a simulink model with Matlab (Mathworks Inc., Natick, USA), where a TOBI client block was used to get the signals in real time.

For the offline training, the recorded files were loaded, and two bandpass filters were applied to the signals, one between 8 to 16 Hz and the other one between 16 to 30 Hz. The signals were cut according to each trial. The signals were normalized and used to train all of the combinations of sCSP. Then, log band power (LBP) features were calculated. Finally, the LBP features were used to train the sLDA model.

For the online operation, the same pipeline was applied to the signal in real time as well without the cutting of the signal, since the EEG signals come in as

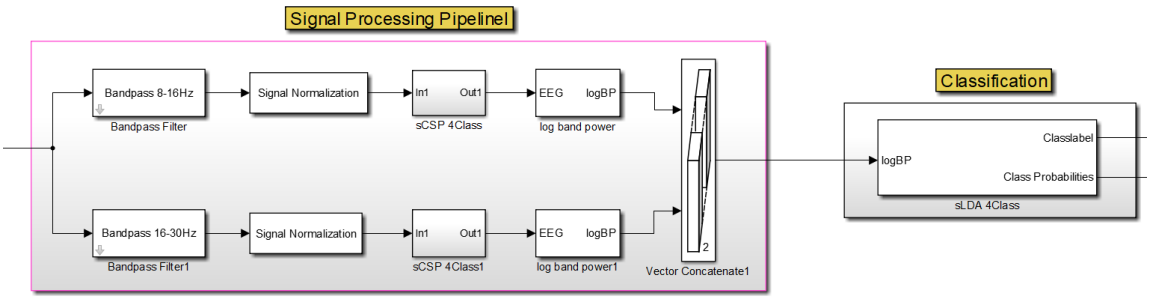


Figure 2.1: Overview of the online processing pipeline. For the offline training, the EEG signals needed to be cut by trial. Taken from the Simulink model [28].

a continuous stream. Figure 2.1 visualizes the processing pipeline of the online operation as described before.

2.2 Basic classification model

In this section, the shrinkage covariance estimation is described, following by the descriptions of two standard algorithms, CSP and LDA. Furthermore, the extension of the two algorithms by utilizing the shrinkage covariance estimation is stated.

2.2.1 Shrinkage Covariance Estimation

With high dimensional data and few observations to estimate the parameters correctly, it can lead to a problem known as curse of dimensionality, that cause a poor performance of the BCI [30], this is because large eigenvalues are estimated too large and small eigenvalues are estimated too small. This problem can be overcome by a method called shrinkage coviariance estimation, which can prevent an error from overestimating or underestimating variance [31].

The empirical way of calculating the covariance is

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T \tag{2.1}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{2.2}$$

where N is a number of observations, $\mathbf{x}_i$ is a vector of the i th observation , $\hat{\boldsymbol{\Sigma}}$ is the covariance, $\hat{\boldsymbol{\mu}}$ is the mean.

To remedy this problem, $\hat{\boldsymbol{\Sigma}}$ is replaced by the shrinkage covariance, $\tilde{\boldsymbol{\Sigma}}$,

$$\tilde{\boldsymbol{\Sigma}}(\gamma) = (1-\gamma)\hat{\boldsymbol{\Sigma}} + \gamma\nu\mathbf{I} \tag{2.3}$$

where ν is the average eigenvalue of $\hat{\boldsymbol{\Sigma}}$

$$\nu = \frac{trace(\hat{\boldsymbol{\Sigma}})}{d} \tag{2.4}$$

where d is the dimension of the feature space, and γ is a regularizing parameter, ranged between 0 and 1.

As the matrix $\hat{\boldsymbol{\Sigma}}$ is positive semidefinite, it can be factorized with eigendecomposition as

$$\hat{\boldsymbol{\Sigma}} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{2.5}$$

where $\mathbf{V}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix. From equation 2.3 and 2.5, the equation becomes

$$\tilde{\boldsymbol{\Sigma}}(\gamma) = \mathbf{V}((1-\gamma)\mathbf{D} + \gamma\nu\mathbf{I})\mathbf{V}^T \tag{2.6}$$

It can be interpreted that eigenvalues are altered toward the average eigenvalue. While both $\hat{\boldsymbol{\Sigma}}$ and $\tilde{\boldsymbol{\Sigma}}$ still share the same eigenvectors, only the eigenvalues are pulled into average eigenvalue, ν. It was shown that finding the optimum value for the shrinkage parameter, γ, can be done analytically [32].

2.2.2 (shrinkage) Common Spatial Pattern ((s)CSP)

The problem with EEG signals is that they suffer from the effect of the volume conduction [33], [34]. This means that an electrode records a combination of the

activity from several sources (or spatial components) in the brain. CSP is a spatial filter which was first introduced by Ramosar et al. [34]. The idea of CSP is to decompose the EEG signals into spatial components such that these components will maximize the variance of the signal from one condition and, at the same time, minimize the variance of the signal from the other condition [35]. This spatial filter is great for an oscillatory-based BCI that utilizes the changes in the SMR power, which is equivalent to the variance that the filter tries to optimize. These changes are corresponding to the ERDS [17], [18]. Several works utilized CSP or its variations in the classification of the BCI [34]–[37], not only for the features from ERDS but also the event-related potential (ERP) as well [30].

Let us define the normalized spatial covariance as

$$\boldsymbol{\Sigma} = \frac{\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}}{trace(\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}})} \tag{2.7}$$

where the $\boldsymbol{X}^T$ is the transpose of $\boldsymbol{X}$.

In this case, $\boldsymbol{X}$ is the matrix containing EEG signals with the dimension $N_{CH} \times N$, where N_{CH} is the number of channels, and N is the number of observations.

The average covariance of both conditions, $\bar{\boldsymbol{\Sigma}}_{\mathbf{1}}$ and $\bar{\boldsymbol{\Sigma}}_{\mathbf{2}}$, can be computed by averaging over trials of each condition separately. Summing the average covariance of 2 classes, we have a composite spatial covariance

$$\boldsymbol{\Sigma}_c = \bar{\boldsymbol{\Sigma}}_{\mathbf{1}} + \bar{\boldsymbol{\Sigma}}_{\mathbf{2}} \tag{2.8}$$

which can be decomposed into

$$\boldsymbol{\Sigma}_c = \boldsymbol{U}_c \boldsymbol{\lambda}_c \boldsymbol{U}_c^T \tag{2.9}$$

where $\boldsymbol{U}_c$ is a matrix of eigenvectors, and $\boldsymbol{\lambda}_c$ is a diagonal matrix of corresponding eigenvalues. Note that the eigenvalues and their corresponding eigenvectors will be assumed to be ordered in a descending manner.

From this, the whitening transformation is defined as

$$\boldsymbol{P} = \boldsymbol{\lambda}_c^{-\frac{1}{2}} \boldsymbol{U}_c^T \tag{2.10}$$

where all eigenvalues of the whitened $\boldsymbol{\Sigma}_c$, computed as $\boldsymbol{P}\boldsymbol{\Sigma}_c\boldsymbol{P}^T$, will be one.

With the transformation, this produces

$$\boldsymbol{S}_1 = \boldsymbol{P}\bar{\boldsymbol{\Sigma}}_1\boldsymbol{P}^T \tag{2.11}$$

$$\boldsymbol{S}_2 = \boldsymbol{P}\bar{\boldsymbol{\Sigma}}_2\boldsymbol{P}^T \tag{2.12}$$

which can be further decomposed by eigendecomposition into

$$\boldsymbol{S}_1 = \boldsymbol{U}\bar{\boldsymbol{\lambda}}_1\boldsymbol{U}^T \tag{2.13}$$

$$\boldsymbol{S}_2 = \boldsymbol{U}\bar{\boldsymbol{\lambda}}_2\boldsymbol{U}^T \tag{2.14}$$

where $\boldsymbol{U}$ is a matrix of shared eigenvectors between two conditions, and $\boldsymbol{\lambda}_i$ is a diagonal matrix of the eigenvalues of $\boldsymbol{S}_i$, with the property of $\boldsymbol{\lambda}_1 + \boldsymbol{\lambda_2} = \boldsymbol{I}$, where $\boldsymbol{I}$ is an identity matrix.

This is optimal to discriminate between two classes since this property means that if an eigenvalue of one class is large, a corresponding eigenvalue of the other class is small.

Finally, the spatial filter is defined as

$$\boldsymbol{W} = (\boldsymbol{U}^T\boldsymbol{P})^T \tag{2.15}$$

Applying $\boldsymbol{W}$ to $\boldsymbol{X}$, we obtain a components matrix that is,

$$\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{X} \tag{2.16}$$

The components $\boldsymbol{Z}$ are sorted such that the explained variance of one condition is at the maximum in the first component and is descending. However, the explained variance for the other condition is at the maximum in the last component. The

components in the middle have little discriminative information and can be discarded, leaving only some number of the first components and the last components to be used [34], [35], [37], [38]. This can be seen as the tool for dimensionality reduction [38]. For example, if only the first two and last two components are selected, the dimension is reduced from N channels to just 4 components. The improvement in the separability between two conditions was demonstrated in several works [33]–[35].

One restriction for CSP is that it is defined for binary problems. However, it can be extended for multi-class problems by determining the projection matrix for each of a combination between 2 classes, i.e. considering three classes, we have six combinations. This leads to 6 projection matrices.

However, the curse of dimensionality arises if the dimension of the feature is large, as well as when the number of training data is too small [31], this can cause an overfitting problem [39]. One of the ways to overcome this is to apply a shrinkage estimation of the covariance [40], [41], as mentioned in section 2.2.1.

2.2.3 (shrinkage) Linear Discriminant Analysis ((s)LDA)

LDA [42] can be used as a classification model by making use of a linear discriminant function, meaning that the decision plane is a linear function defined as

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \tag{2.17}$$

where $\mathbf{w}$ is defined as a weight vector and w_0 is defined as a bias term. In the classification problem, if $y(\boldsymbol{x}) > 0$, x would be classified as class 1 and if $y(\boldsymbol{x}) \leq 0$, $\boldsymbol{x}$ would be classified as class 2 instead.

These parameters can be varied infinitely, but the goal of the classification problem is to be able to classify each data point correctly, which means that choosing the optimum parameters is essential. The model parameters, $\boldsymbol{w}$, w_0, are usually fitted to calibration data.

Let us first define the mean vector of class i as

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{n \epsilon C_i} \mathbf{x}_n \tag{2.18}$$

where N_i is the number of sample that belong to class i.

One simple way of maximizing the class separability is to maximize the difference between the projected class mean as

$$(\tilde{m}_2 - \tilde{m}_1) = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) \tag{2.19}$$

where $\tilde{m}_i = \mathbf{w}^T\mathbf{m}_i$. Here, the bias term, w_0 is merged into $\boldsymbol{w}^T$.

The within-class variance without the normalized term of the projected data from class C_i is defined as

$$s_i^2 = \sum_{n \epsilon C_i} (y_n - \tilde{m}_i)^2 \tag{2.20}$$

where $y_n = \mathbf{w}^T\mathbf{x}_n$.

The ratio of between-class variance to the within-class is defined as

$$J(\mathbf{w}) = \frac{(\tilde{m}_2 - \tilde{m}_1)^2}{s_1^2 + s_2^2} \tag{2.21}$$

or in the form of dependency of $\mathbf{w}$

$$J(\mathbf{w}) = \frac{\mathbf{w^T S_B w}}{\mathbf{w^T S_W w}} \tag{2.22}$$

where $\mathbf{S_B}$ is called between-class covariance matrix defined as

$$\mathbf{S}_B = (\mathbf{m_2} - \mathbf{m_1})(\mathbf{m_2} - \mathbf{m_1})^T \tag{2.23}$$

and $\mathbf{S_W}$ is called within-class corvariance matrix defined as

$$\mathbf{S}_W = \sum_{n \epsilon C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \epsilon C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \tag{2.24}$$

Taking the derivative of $J(w)$ and setting it to zero, we have,

$$(\mathbf{w}^T\mathbf{S}_B\mathbf{w})\mathbf{S}_W\mathbf{w} = (\mathbf{w}^T\mathbf{S}_W\mathbf{w})\mathbf{S}_B\mathbf{w} \tag{2.25}$$

Maximizing the term $J(\boldsymbol{w})$ means that we try to maximize between-class covariance while minimizing within-class covariance at the same time.

From equation 2.19, 2.23, it can be seen that $\mathbf{S}_B\boldsymbol{w}$ is in the same direction as $(m_2 - m_1)$ and that the scaling of $\mathbf{w}$ is not important, hence, we can ignore the scalar terms, $(\mathbf{w}^T\mathbf{S}_B\mathbf{w})$ and $(\mathbf{w}^T\mathbf{S}_W\mathbf{w})$, which results in

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{2.26}$$

Furthermore, the same trick of shrinkage covariance estimation in section 2.2.1 can be applied with LDA in the calculation of the within-class variance, $\mathbf{S}_W$, [43], [44]. Moreover, Peck R. et al. [43] also reported that the shrinkage covariance estimation indicated a better performance than the standard LDA, if the distribution of the data from two conditions are Gaussian and share the same covariance matrix.

2.3 Candidate Algorithms

In this section, the candidate algorithms to address intra-session non-stationarities are discussed in more detail. The candidate algorithms were chosen to be some variations of the basic classification model mentioned before (sCSP and sLDA), so as not to change the overall structure of the system.

2.3.1 Adaptive normalization LDA (AdLDA)

Adaptive normalization LDA (**AdLDA**) is an extension of LDA. It was shown that the means of the features could change over time [45] because of the non-stationarities. Due to these changes of the means, Vidaurre et al. [24] introduced the idea of adapting the means.

The linear discriminant function can be written in a vector form as

$$y(x) = \begin{bmatrix} w_0 & \mathbf{w}^T \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \tag{2.27}$$

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{2.28}$$

$$w_0 = -\mathbf{w}^T\mathbf{m} \tag{2.29}$$

where $\mathbf{m} = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)$ is the pooled mean of two classes

One can think of the bias term, w_0, as a function of time instead of just a constant. This leads to

$$w_0(t) = -\mathbf{w}^T\mathbf{m}(t) \tag{2.30}$$

where $\mathbf{m}(t)$ can be updated with the data. With this approach, only the bias term, w_0, will be updated, but the weight vector $\mathbf{w}$ will not. In the hyperspace of the features, this can be interpreted as moving the discriminant plane according to the changes in the features' means.

The approach taken in this work was slightly different from [24]. Instead of moving the discriminant plane around, the features were adaptively normalized. Both the mean $\tilde{\mathbf{m}}(t)$ and the standard deviation (SD) $\tilde{\mathbf{s}}(t)$ of the feature $\mathbf{f}(t)$ are estimated with an exponential moving average (EMA) estimation.

$$\tilde{\mathbf{m}}(t) = (1-\eta)\tilde{\mathbf{m}}(t-1) + \eta\mathbf{f}(t) \tag{2.31}$$

$$\tilde{\mathbf{s}}(t) = [(1-\eta)\tilde{\mathbf{s}}(t-1)^2 + \eta(\mathbf{f}(t) - \tilde{\mathbf{m}}(t))^2]^{\frac{1}{2}} \tag{2.32}$$

and

$$\mathbf{f}_{normalized}(t) = (\mathbf{f}(t) - \tilde{\mathbf{m}}(t)) \times diag(\tilde{\mathbf{s}}(t))^{-1} \tag{2.33}$$

where $\mathbf{f}(t)$ is the features vector, $\tilde{\mathbf{m}}(t)$ is the mean of the features that was filtered with EMA, and $\tilde{\mathbf{s}}(t)$ is the standard deviation of the features that was filtered with EMA, and η is the smoothing parameter. The speed of adaptaion can be changed

by varying the smoothing parameter, η. From equation 2.31, it can be expanded as an infinite sum as

$$\begin{aligned}\tilde{\mathbf{m}}(t) =&\eta[\mathbf{f}(t) + (1-\eta)\mathbf{f}(t-1) + ... + (1-\eta)^k\mathbf{f}(t-k)] \\ &+ (1-\eta)^{k+1}\tilde{\mathbf{m}}(t-(k+1))\end{aligned} \tag{2.34}$$

where $k\epsilon\{0, 1, 2, ...\}$. The weight of the term $\mathbf{f}_{t-i}$ is $\eta(1-\eta)^i$. The percentage of the weight represented in the k terms, p, can be expressed as

$$(1-p) = \frac{\textit{weight limited after k term}}{\textit{total weight}} \tag{2.35}$$

$$= \frac{\eta[(1-\eta)^k + (1-\eta)^{k+1} + ...]}{\eta[1 + (1-\eta) + (1-\eta)^2 + ...]} \tag{2.36}$$

using Maclaurin Series

$$(1-p) = \frac{\eta(1-\eta)^k \frac{1}{1-(1-\eta)}}{\eta\frac{1}{1-(1-\eta)}} \tag{2.37}$$

$$= (1-\eta)^k \tag{2.38}$$

Finally, the smoothing term can be computed as

$$\eta = 1 - (1-p)^{\frac{1}{k}} \tag{2.39}$$

where k is the number of terms before truncation and p is the percentage of their EMA weights.

The same EMA is also applied to the standard deviation $\tilde{\mathbf{s}}(t)$ in equation 2.32.

In this work, the parameter was set to $p = 0.9$ and $k = F_s \times 600$. So that the weights of the samples within the last 10 minutes have 90% of the weights.

2.3.2 Importance-weighted LDA (IWLDA)

One of the assumptions of supervised learning algorithms, such as LDA, is that the distribution of the training and testing data is the same. Practically, this is not always true and under such conditions, the algorithms might fail [46]–[48].

We consider the case where the distribution of the data, $P(x)$, between training and testing data is not the same but the conditional distribution of output labels, $P(y|x)$, is unchanged. This situation is termed covariate shift [48].

Shimodaira et al. [48] also proposed that the effects caused by covariate shift could be improved by weighting the log-likelihood term during the training with the *importance*, $d(x)$, defined as

$$d(\mathbf{x}) = \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} \tag{2.40}$$

where $p_{te}(\mathbf{x})$ is the density function of testing data and $p_{tr}(\mathbf{x})$ is the density function of training data.

However, in order to use this ratio in weighting, one can relate the learning of LDA parameters in a least-squares manner. First, the target value, indicating the class that each observation belongs to, has to be set to $\frac{N}{N_1}$ and $\frac{-N}{N_2}$ where N is the total number of samples, and N_i is the number of samples that belong to class i. The objective function is

$$\min_{\mathbf{W}} ||\mathbf{D}^{\frac{1}{2}}(\mathbf{Y} - \mathbf{X}\mathbf{W})||^2 + \lambda||\mathbf{W}||^2 \tag{2.41}$$

where $\mathbf{D}$ is a diagonal matrix containing importance of each data point and λ is a regularization term. From this minimization problem, the solution can be derived as [49]

$$\hat{\mathbf{W}}_{IWLDA} = (\mathbf{X}^T\mathbf{D}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{D}\mathbf{Y} \tag{2.42}$$

Despite the derived solution, finding the importance of the data remains a difficult task. The simplest way is to determine the densities of both testing and training data individually, then directly compute the importance by finding the ratio from both densities. However, estimating density from the data with high dimensional is

not an easy task.

One way of estimating the importance without actually determine the data densities was proposed by Sugiyama et al. [46]. This method is called Kullback–Leibler importance estimation procedure (KLIEP), which utilizes the Kullback-Leibler divergence [50], that tells how different one distribution to another.

The importance is, modeled as a summation of Gaussian kernels

$$\hat{w}(\mathbf{x}) = \sum_{l=1}^{b} \alpha_l exp(-\frac{||\mathbf{x} - c_l||^2}{2\sigma^2}) \tag{2.43}$$

where α_l are the parameters needed to be learned, c_l is picked randomly from the training data, $\mathbf{x}^{te}$, the number of parameter, b, is chosen as $min(100, N_{te})$, where N_{te} is the number of testing data [49] and the width of Gaussian kernel, σ, can be selected by cross-validation [46]. Applying this model results in

$$\hat{p}_{te}(\mathbf{x}) = \hat{w}(\mathbf{x}) p_{tr}(\mathbf{x}) \tag{2.44}$$

From this, α_l must be learnt so that the Kullback-Leibler divergence between $p_{te}(\mathbf{x})$ and $\hat{p}_{te}(\mathbf{x})$ is minimized.

$$KL[p_{te}(\mathbf{x})||\hat{p}_{te}(\mathbf{x})] = \int p_{te}(\mathbf{x}) log \frac{p_{te}(\mathbf{x})}{\hat{w}(\mathbf{x}) p_{tr}(\mathbf{x})} dx \tag{2.45}$$

$$= \int p_{te}(\mathbf{x}) log \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} dx - \int p_{te}(\mathbf{x}) log(\hat{w}(\mathbf{x})) dx \tag{2.46}$$

The first term can be ignored since it does not depend on α_l, the last term can be approximated from the testing data as

$$\int p_{te}(\mathbf{x}) log(\hat{w}(\mathbf{x})) dx \approx \frac{1}{N_{te}} \sum_{j=1}^{N_{te}} log(\hat{w}(\mathbf{x}_j^{te})) \tag{2.47}$$

$$= \frac{1}{N_{te}} \sum_{j=1}^{N_{te}} log(\sum_{l=1}^{b} \alpha_l exp(-\frac{||\mathbf{x}_j^{te} - c_l||^2}{2\sigma^2})) \tag{2.48}$$

Because $\hat{p}_{te}(\mathbf{x})$ is a probability density function, it must be normalized to 1 which means

$$1 = \int \hat{p}_{te}(\mathbf{x})dx \tag{2.49}$$

$$= \int \hat{w}(\mathbf{x})p_{tr}(\mathbf{x})dx \tag{2.50}$$

$$\approx \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \hat{w}(\mathbf{x}_i^{tr}) \tag{2.51}$$

$$= \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} \sum_{l=1}^{b} \alpha_l exp(-\frac{||\mathbf{x}_i^{tr} - c_l||^2}{2\sigma^2}) \tag{2.52}$$

Again, the integral was approximated from the training data $\mathbf{x}_i^{tr}$.

Note that only testing data is used in Equation 2.48, however, the training data is utilized in the constraint in Equation 2.52.
Finally, the optimization problem becomes

$$\max_{\alpha_l} \left[\sum_{j=1}^{N_{te}} log(\sum_{l=1}^{b} \alpha_l exp(-\frac{||\mathbf{x}_j^{te} - c_l||^2}{2\sigma^2})) \right] \tag{2.53}$$

$$\text{subject to} \sum_{i=1}^{N_{tr}} \sum_{l=1}^{b} \alpha_l exp(-\frac{||\mathbf{x}_i^{tr} - c_l||^2}{2\sigma^2}) = N_{tr} \text{ and } \alpha_1, \alpha_2, ..., \alpha_b \geq 0 \tag{2.54}$$

A Matlab implementation of the KLIEP algorithm by Sugiyama, which is publicly available based on [46], [47], was used to find the importance of each class of the training data separately, and then they were combined by normalizing them to sum up one.

2.3.3 Within-session divergence-based CSP (divCSP-WS)

The idea of divCSP idea was first proposed by Samek et al. [51], where it was described that the problem of finding the optimal projection matrix of CSP could as well be represented in the maximization problem of the symmetric Kullback-Leibler

divergence, $\tilde{D}_{KL}$, [50]. The maximization problem can be written as

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}}\, \tilde{D}_{KL}(\mathbf{W}^T\mathbf{\Sigma}_1\mathbf{W}||\mathbf{W}^T\mathbf{\Sigma}_2\mathbf{W}) \tag{2.55}$$

where $\mathbf{W}$ is the CSP coefficient containing all of the components.

As pointed out by Samek et al. in their consecutive work [41], there are two ways to incorporate regularization into CSP. One way is to regularize during the estimation of the covariance matrix as in section 2.2.1 and 2.2.2. The other way is to regularize the objective function of the optimization problem directly. They further extended their work by not only generalizing the framework for CSP with the divergence-based framework but also including the second approach of regularizing the objective function directly at the same time. The proposed framework along with the regularization is formalized into

$$\mathcal{L}(\mathbf{W}) = (1-\lambda)\tilde{D}_{KL}(\mathbf{W}^T\mathbf{\Sigma}_1\mathbf{W}||\mathbf{W}^T\mathbf{\Sigma}_2\mathbf{W}) - \lambda\mathbf{\Delta} \tag{2.56}$$

where $\mathcal{L}(.)$ is the loss function, the first term is the term from optimizing the problem of divCSP, and the second term is the regularization term. The authors proposed different regularization terms to tackle 4 cases, which were

1. Within-session stationarity (divCSP-WS)
2. Between-session Stationarity (divCSP-BS)
3. Across Subject stationarity (divCSP-AS)
4. Multisubject CSP (divCSP-MS)

Here within-session stationarity is relevant since the goal of this thesis is to improve the performance of the BCI within the same session.

The equation described the regularization term for divCSP-WS is

$$\mathbf{\Delta} = \frac{1}{2N}\sum_{c=1}^{2}\sum_{i=1}^{N_c} D_{KL}(\mathbf{W}^T\mathbf{\Sigma}_c^i\mathbf{W}||\mathbf{W}^T\mathbf{\Sigma}_c\mathbf{W}) \tag{2.57}$$

where N_c is the number of epochs of class c, $\mathbf{\Sigma}_c^i$ is the estimated covariance matrix of epoch i of class c.

The idea of this regularization term is to separate the training data into several epochs of the same class. Then, the average divergence between each epoch and the overall data is estimated for each class separately, where the divergence represents the shift of the signals due to the non-stationarities. Please be aware that the term D_{KL} is not the symmetrical Kullback–Leibler divergence but the regular version instead. The reason for doing this is that the non-symmetric version of Kullback–Leibler divergence will shrink the effect of the matrix $\mathbf{W}^T\mathbf{\Sigma}_c^i\mathbf{W}$ in case if it is ill-conditioned. The optimization problem was solved using a Matlab implementation of divCSP by Samek based on [51], [52].

Chapter 3

Methods

This chapter consists of an explaination of the offline simulation and then the online experiment. The candidates algorithm are **AdLDA**, **IWLDA**, **divCSP-WS**, **AdLDA+divCSP-WS**, and **IWLDA+divCSP-WS**.

3.1 Offline Simulation

In this section, the details of the offline simulation is explained. It begins with a description of the dataset and an experimental paradigm. Afterwards, the calibration pipeline and the model operation are explained. The simulation scenario of each algorithm is given and then the evaluation of the offline simulation.

3.1.1 Dataset Description

The dataset used in the offline simulation was provided by Julia Brandstetter [53]. In her study, EEG data was collected from 20 participants. They were divided into two groups to compare between supervised adaptation and semi-supervised adaptation in a 2-class BCI (hand motor imagery and feet motor imagery). However, only the semi-supervised group was used in the simulation.

3.1.1.1 Participants

The dataset came from the semi-supervised group consisted of 10 healthy participants. There were five male and five female participants. The median age of the participant was 24 years and ranged from 20 to 35 years. One participant was excluded since more than half of the training data were rejected.

3.1.1.2 Paradigm

The experimental paradigm was the standard Graz-BCI paradigm [54] as depicted in Figure 3.1.

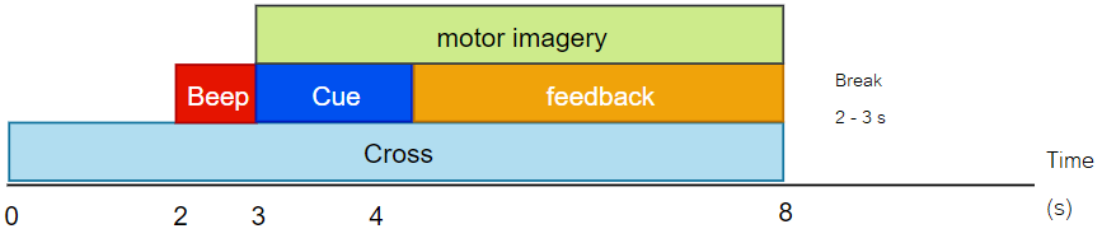


Figure 3.1: The standard Graz-BCI paradigm of each trial. Each colored block represents the timing of each element used in the experiment, which are cross, beep sound, cue image, feedback, and motor imagery.

A trial began with the emergence of a white cross in the center of a black screen. After 2 seconds, a beep was played. Then, the cue indicating the task that the participant has to perform (either hand motor imagery or feet motor imagery), was shown. The cue also indicated the participant to begin performing motor imagery. Feedback was shown at 4.25 seconds, and continuously updated. After 8 seconds, the screen turned black again, which indicated the end of the trial. After each trial, there was a period of the break, which lasted between 2 to 3 seconds.

The EEG signals were recorded with g.tec Gamma box and g.tec USBamp (g.tec, Graz, Austria) biosignal amplifiers with 13 channels measured with g.tec Ladybird active electrodes with a sampling rate of 256 Hz. A bandpass filter between 0.1 and 100 Hz was applied. The electrodes were placed to cover the sensorimotor area of

the brain, which are FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz, and CP4.

The experiment started with a calibration period, which consisted of four runs, each including ten trials per class. After the first run, the data was used to calibrate the classifier, and this classifier was then used to give feedback in the second run of the calibration period. The calibration period was followed by a break that lasted 20 minutes. Then, five runs were recorded. These processes were repeated three times after the initial calibration.

Figure 3.2 shows the overview of the whole simulation based on this dataset. The difference was that all of the data from the first block was used in a calibration instead of just the first run.

Figure 3.2: The simulation paradigm. The data from the calibration block was used to train the model that was used in the three operation blocks.

3.1.2 Calibration pipeline

In order to calibrate a BCI, the classification models were fitted to the calibration data. The processing pipeline was based on the predecessor system with some revision taken from the semester project of John Bosco Uroko. Figure 3.3 visualizes the calibration pipeline.

The files containing signals collected from the calibration runs were loaded into EEGLAB format, which is a Matlab toolbox for EEG data analysis [55]. The signals were filtered with an eighth-order butterworth low-pass filter with a cut-off frequency of 80 Hz. Afterward, they were downsampled from 500 Hz to 250 Hz. To get rid of a low-frequency drift, they were filtered with a second-order butterworth high-pass filter with a cut-off frequency of 1 Hz. Finally, a forth-order butterworth notch filter was applied to remove the power line noise at 50 Hz. The signals were checked for

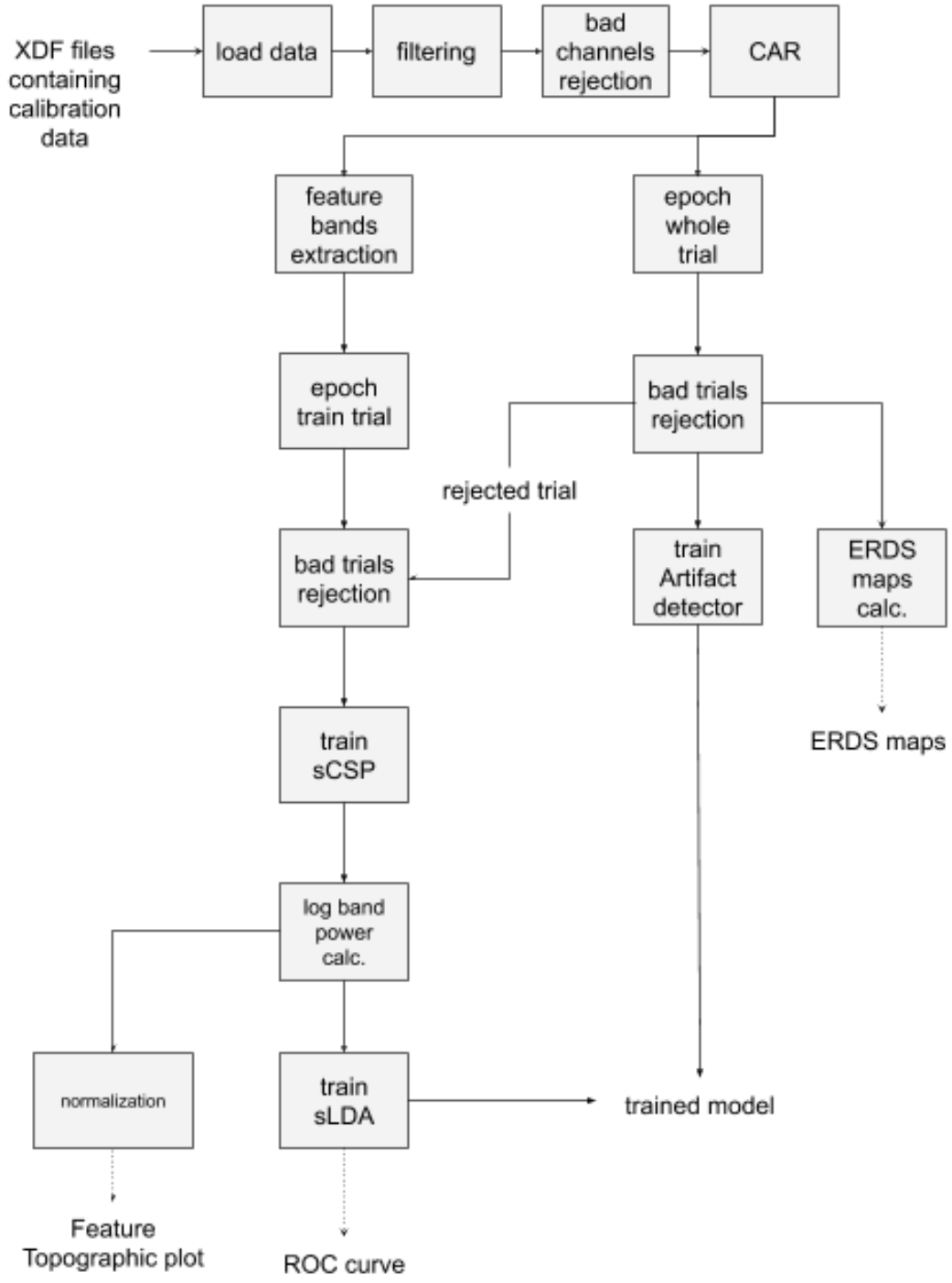


Figure 3.3: The overview of processing pipeline. Each block represents a function where the arrows indicates an input-output relationship between blocks.

any bad channels which should be removed. After the rejection of bad channels, a Common Average Reference (CAR) filter [56] was applied to the signal.

Subsequently, the pipeline was separated into two branches. The first one was used to find bad epochs to be removed. Here, the signals from the last step were cut for the whole period of trial, which was between 2 seconds before and 5 seconds after the cue for each trial. Statistical properties of the signals were used to determine which trial should be rejected, according to [53]. The clean trials were used to compute the ERDS maps [18], [19]. Furthermore, the remaining trials were used to train an artifact detector, according to [57]. This artifact detector was used to give feedback to the participant of how likely the current trial was contaminated.

In the second branch, the preprocessed signals were divided into four narrower band signals through bandpass filtering with third-order butterworth filters. These four bands consisted of 6-10 Hz, 8-12 Hz, 15-25 Hz, 25-35 Hz, which covered components of the alpha and beta frequency range. The 3-second epochs starting from 1 second to 4 seconds after the cue for each trial were extracted. The list of bad trials was used to reject outlier trials. After the trial rejection, narrow band signals were used to train sCSP filters to reduce the dimension of the features and to optimize separability between 2 classes. Afterwards, LBP features were computed with a 1-second sliding window and the time point at 2, 3, and 4 seconds were picked from the CSP-filtered signals, separately for each frequency band [36]. These LBPs were normalized to compute a topographic plot of features and finally, sLDA was trained with LBP features.

The classification model was used for evaluation by cross-validation. A receiver operating characteristic (ROC) curve were calculated. Parameters for sCSP and sLDA were saved for the online classification.

3.1.3 Model operation

For model operation in the simulation, the model parameters from the training period were used for classification as visualized in Figure 3.4.

First, a chunk of EEG signals was obtained in real time. The signals were filtered

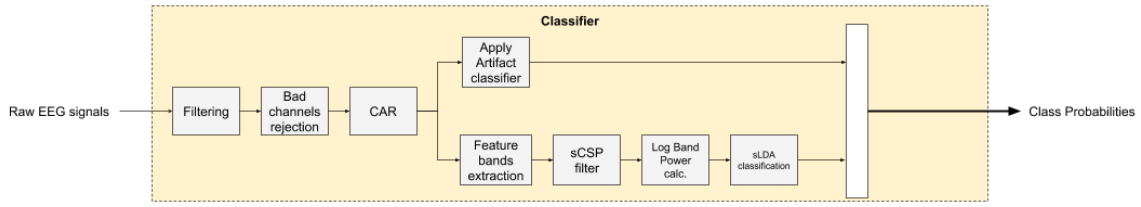


Figure 3.4: Overview of model application. The yellow block indicates the application of the model in the online case. Each of the grey block represents a function and the arrows visualizes the flow of the signals.

in the same way as in the calibration pipeline. All of the filters applied were causal filters. Then, the bad channels indicated from the calibration were rejected. In one branch, the artifact detector was used to determine the probability of the signals being an artifact. In the other branch, feature extraction was carried out, as mentioned in the calibration period. sCSP filters were applied followed LBP features extraction. Afterwards, the sLDA classifier was used to compute the probability for each class. Finally, the probabilities output from both artifact classifier and the sLDA model were concatenated and sent back. The information was received by the visualizer, that displayed the feedback at the end of a trial.

3.1.4 Simulation scenario

As pointed out in the introduction, the candidate algorithms have to comply with the CYBATHLON scenario [27] so that

- the training dataset is only from the calibration period
- the labels of the testing dataset are not available

The simulation was carried out with these restrictions. The offline simulation was done with four algorithms and two combinations.

3.1.4.1 Standard algorithm

The **standard** algorithm is described in section 2.2. It utilized sCSP and sLDA. The BCI calibration follows the same processing line mentioned in section 3.1.2. In the operation blocks, the model was used as it was calibrated, without an adaptation. Figure 3.5 visualizes the simulation scenario of the **standard** algorithm.

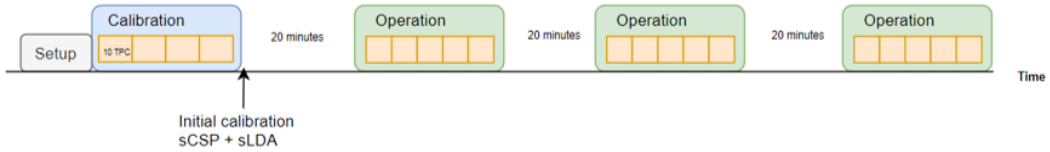


Figure 3.5: The simulation scenario of the **standard** algorithm. sCSP and sLDA were calibrated after the calibration block and were used in the operation blocks.

3.1.4.2 AdLDA

For the **AdLDA**, which was explained in section 2.3.1, the calibration was done with sCSP and sLDA similar to the standard algorithm, but the normalization was applied to the LBP features. Afterwards, the normalization parameters of the features were updated as the new samples were available, throughout the whole simulation. Figure 3.6 visualizes the simulation scenario of **AdLDA**.

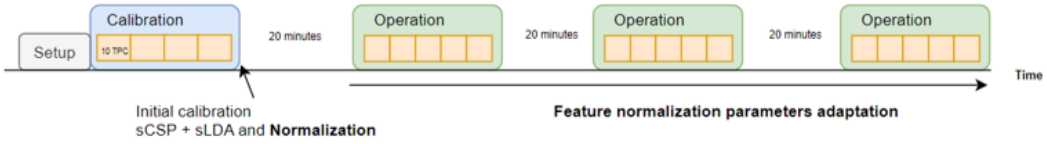


Figure 3.6: The simulation scenario of the **AdLDA** algorithm. sCSP and sLDA were calibrated with normalized features after the calibration block and were used in the operation blocks with the adapatation of normalization parameters.

3.1.4.3 IWLDA

The initial calibration of **IWLDA** was done in the same way as the **standard** algorithm. The difference was that after the end of each run, the data from that run was added into the testing dataset, which was then employed into the KLIEP algorithm to (re)determine the importance of the training data. This updated importance was used to retrain the **IWLDA** model. Note that the testing data will grow larger, but the training data will be the same. The testing data was used only to determine the importance, as it was not included in the training data. Figure 3.7 visualizes the simulation scenario of **IWLDA**.

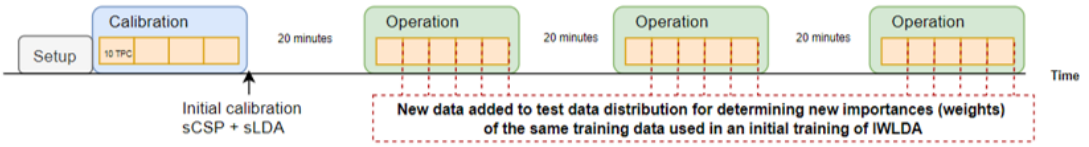


Figure 3.7: The simulation scenario of **IWLDA**. The calibration was done exactly as in the **standard** algorithm. At the end of each run, the new data were included to compute the importance term of the training data.

3.1.4.4 divCSP-WS

The details of this algorithm can be found at 2.3.3. The simulation was done similarly to the standard algorithm with the change that **divCSP-WS** was learnt instead of normal sCSP. After that, the model was used without any adaptation or retraining. Figure 3.8 visualizes the simulation scenario of **divCSP-WS**.

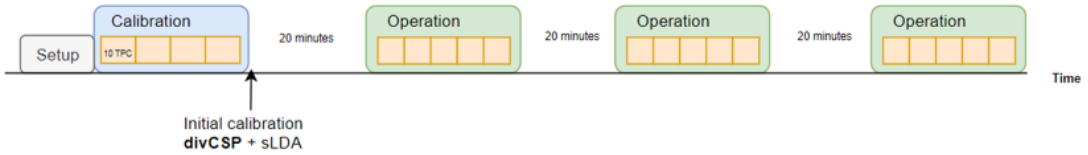


Figure 3.8: The simulation scenario of **divCSP-WS**. The calibration was done with **divCSP-WS** and sLDA. The model was used without adaptation in the operations blocks.

The following parameters were used in the simulation

- **max_iter = 1000** the maximum iteration is 1,000 iterations
- **nreps = 1** no repetition in the calculation of CSP
- **lam = 0.5** the regularization parameter
- **beta = 0** apply Kullback-Leibler divergence
- **mode = 0** for choosing **divCSP-WS**
- **deflation = 0** applying subspace algorithm
- **pca = 1** PCA will be applied to the filter
- **csp_init = 1** initialize the repetition with CSP solution
- **sym = 0** apply the non-symmetric Kullback-Leibler divergence

The model was trained once with the training data, no information from the testing data used.

3.1.4.5 AdLDA + divCSP-WS

In this scenario, the data were simulated with both **AdLDA** and **divCSP-WS**. The initial calibration was done with **divCSP-WS** and sLDA with normalization. Figure 3.9 visualizes the simulation scenario of **AdLDA + divCSP-WS**.

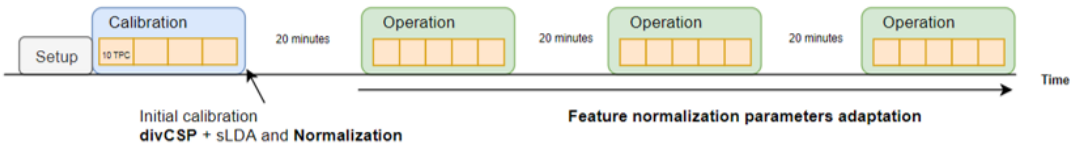


Figure 3.9: The simulation scenario of **AdLDA + divCSP-WS**. The calibration was done with divCSP-WS, sLDA, and features normalization. In the operation blocks, the normalization parameters were adapted.

3.1.4.6 IWLDA + divCSP-WS

In the initial calibration, **divCSP-WS** and **IWLDA** were trained, and at the end of each run, the new data points were included in the testing data pool, which were used in the (re)calculation of the importance and **IWLDA** was relearnt after that. The training data was the same for the whole simulation only that the importance was updated. Figure 3.10 visualizes the simulation scenario of **IWLDA + divCSP-WS**.

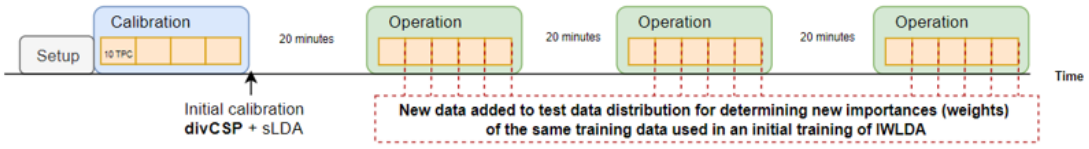


Figure 3.10: The simulation scenario of **IWLDA + divCSP-WS**. The calibration was done with **divCSP-WS** and sLDA. The importance terms of the training were recomputed.

3.1.5 Evaluation

For the evaluation of the different algorithms, we computed classification accuracy curve for each participant and section of the three operation blocks. The chance level was also calculated according to [58] with 150 trials per class for a 2-class problem. The median of the accuracy values between 1.5 to 4.5 seconds after the cue marker was extracted for each participant and for each operation block. These were used to test with two-way repeated measures analysis of variance (ANOVA) if the effects of 2 factors, namely; algorithm factor and section factor, were significant.

3.2 Online Experiment

Only one algorithm was chosen to be implemented in the experiment. The offline analysis revealed that **AdLDA** achieved the best performance

3.2.1 Participant characteristics

In total, 10 (1 female and 9 males) volunteers participated in the experiment. The age ranged from 20 to 30 years with a mean age of 24.8± SD 3.15 years. All of them were right-handed. After the participants were briefed about the procedure and goals of the experiment, a written consent form was obtained.

3.2.2 Hardware and electrode montage

The system used in the measurement was the BrainVision LiveAmp 32 (Brain Products GmbH, Gilching, Germany) biosignal amplifier. It is a 32-channel wireless amplifier. The system uses gel-based active electrodes (actiCAP slim). The impedance of the electrodes was kept below $10k\Omega$ by applying electrolyte gel between electrode and the scalp during the setup.

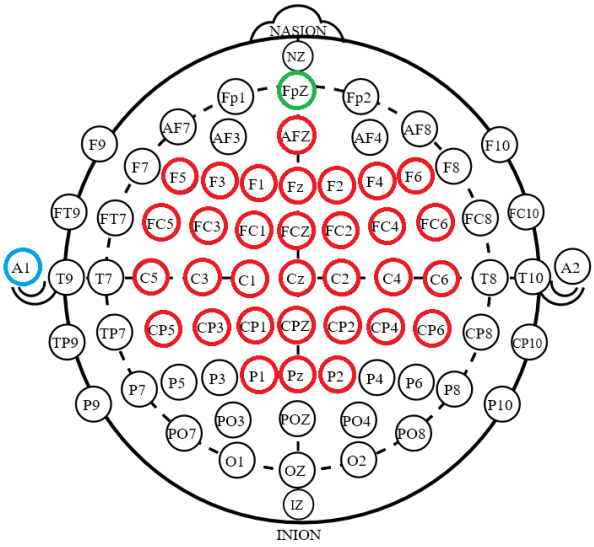


Figure 3.11: The topography plot of the electrode montage according to the extended 10-20 system. The blue ring and the green ring represent the reference and ground electrode, respectively. Red rings indicate normal electrodes.

As illustrated in Figure 3.11, the colored ring depicted the placement of electrodes where green indicates the ground electrode, red the measuring electrodes, and blue

the reference electrode. In this montage, the reference electrode was placed at the left mastoid, the ground electrode was placed at FPz and the other electrodes were placed to cover the central part of the scalp.

3.2.3 Graz BCI Racing Team system

As already mentioned, the new recording hardware was not compatible with the predecessor system. In this section the new BCI of the Graz BCI racing team is presented. In order to understand how the system works, we present the key changes with a focus on the new structure, processing pipeline, and classification process. This BCI was used in the online experiment.

3.2.3.1 Structure

Parts of the system design are based on the predecessor system. The parts of the system involved in the experimental paradigm are described below. The interface with the CYBATHLON game is not within the scope of this thesis. Therefore, it is omitted.

The structure of the system is illustrated in Figure 3.12. The backbone of the system is the Lab Streaming Layer (LSL) system, which manages the communication across platforms or different computers. Different clients can send and receive signals in real time with the LSL protocol. Two scenarios were considered in Figure 3.12, offline experiment, and online experiment.

In the first scenario, a custom Matlab client (a visualizer) was responsible for displaying an experimental paradigm as well as submitting markers, indicating what was presented at which time, to LSL. The LSL packages comes with several clients which is used in the system, namely, BrainVision Remote Data Access (RDA) client was used to acquire EEG signals, and Lab Recorder was used to collect every stream and save it into an XDF file format. The second scenario shared a similar design with the first one. However, one more Matlab instance, a classifier, was used for classification and provided the probabilities needed for feedback visualization to the first Matlab client, the visualizer.

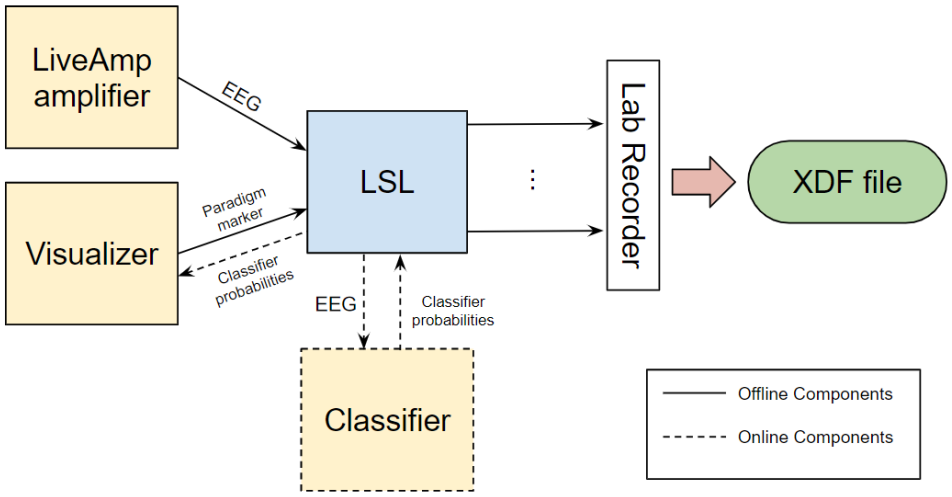


Figure 3.12: The overall structure of the system. Each block represents a software instance and the arrow indicates the dependencies of each block. Dashed-line visualizes components in the online scenario.

3.2.3.2 Calibration processing pipeline

The processing pipeline of the calibration is exactly as in 3.1.2.

3.2.3.3 Online classification

The classification processes in the online experiment was done as in 3.1.3.

3.2.4 Experimental paradigm

The experiment consisted of 4 blocks in total, 1 calibration block, and 3 operation blocks. The operation blocks were interleaved with 30-minute breaks. However, the first operation block started immediately after calibration block. During the breaks, the participants watched a nature documentary video. The whole experiment took around 3.5 to 4.25 hours depending on the setup and montaging time. Figure 3.13 illustrates the experimental paradigm.

Each block consisted of 4 runs represented in yellow boxes in Figure 3.13. There

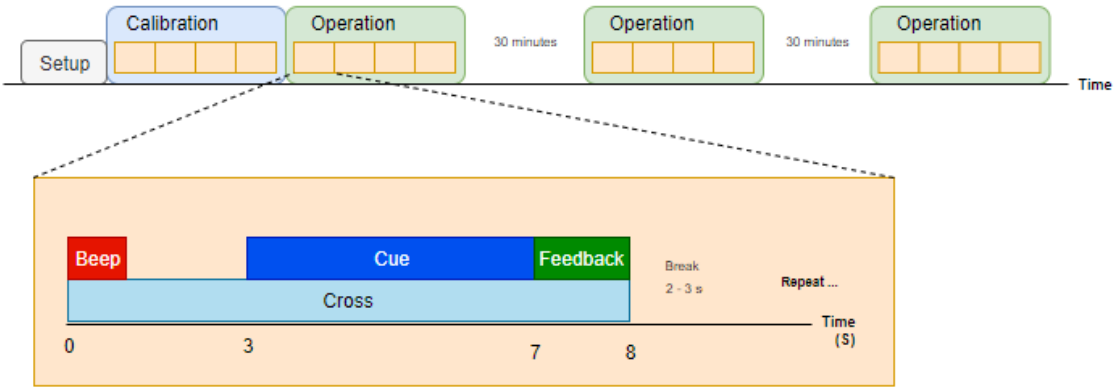


Figure 3.13: The overview of experimental paradigm. The blue and green box represent the calibration and operation blocks. Each block consists of 4 runs, indicating by small yellow blocks. The experimental paradigm was adapted from the Graz-BCI paradigm [54]. The figure below indicates the timing of each component in the experiment, namely, cross, beep sound, cue image, and feedback.

were 30 trials per run (10 trials per class). Based on the finding of [22], 3 classes were chosen for the BCI. They were

- **Hand motor imagery**: imagine the feeling of squeezing a ball with the right hand in the speed of around 2 Hz
- **Mental rotation**: visualize a solid cube rotating in a space e.g. rubics cube
- **Mental subtraction**: subtract number shown on the screen with 7, consecutively e.g. 315 - 7 = 308, 308 - 7 = 301, and so on

The experimental paradigm was adapted from the standard Graz-BCI paradigm [54]. Each trial began with an appearance of a white cross and the beep sound to remind the participant. After 3 seconds, a cue indicated the corresponding task for that trial. It disappeared after 4 seconds, then the feedback was presented to the participant.

Pictures that were used as cues can be found in Figure 3.14. An image of a hand indicated the hand motor imagery, while a dice indicated the mental rotation, and a

number that was randomly generated indicated the mental subtraction.

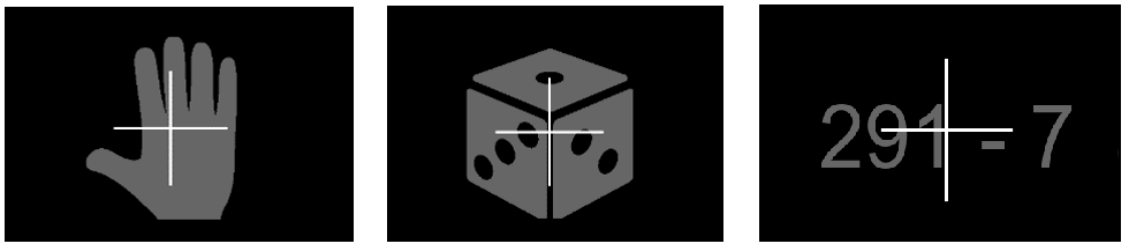


Figure 3.14: The cues indicating , hand motor imagery (left), mental rotation (middle), and mental subtraction (right).

3.2.5 Feedback

Feedback was given to the participant at the end of each trial in both calibration and operation blocks. The feedback bars consisted of a white bar and a red bar. The white bar represented the class probabilities output from the classification model, while the red bar denoted the probabilities of being contaminated with artifacts, that were output from the artifact detector, according to [57]. Figure 3.15 visualizes the feedback presented at the end of trial.

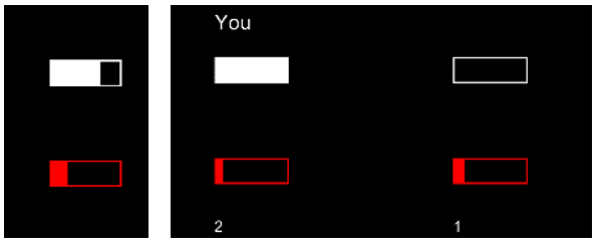


Figure 3.15: The feedback that the participant received in the feedback period, (left) in the calibration block and (right) in the operation blocks. The white bar indicates the probability of being the correct class, while the red bar indicates the probability that the signal contained artifacts. The numbers below the red bars are the score of the participant and a rival player.

Since the BCI was not yet calibrated, the feedback in the calibration block was pseudo-randomly generated. The participants were told that the feedback bars re-

sulted from a general classifier. It was done to keep the same experimental paradigm in both the calibration block and the operation blocks. In contrast to the feedback given in the calibration block, the one given in the operation blocks consisted of 2 sets of the feedback bars. The left set belonged to the participant, and the right set belonged to a "rival", which was pseudo-randomly generated. The probability of being correct, as well as the length of the bars of the rival can be adjusted freely. The inclusion of the rival was done to induce the competitive mind during the competition as in a CYBATHLON scenario. The participants were instead informed that the rival was a collection of other players, to give an idea that the participants were competing with real people. The classification bar ranged from 33%, since this is a 3-class classification, to 100%, while the artifact bar ranged from 0% to 100%. The BCI decided the output from the class with the highest probability.

The probability of the rival for being correct, p_{rival}, was determined from the average AUC score of each class for individual participant from the training period. In order to induce the participants into the competitive mind, the probability was set to 10% more than the average AUC score. However, the maximum probability was set to 90 % to give the idea that the rival did not always perform perfectly. This can be summarized into

$$p_{rival} = max(\frac{AUC_{Hand} + AUC_{Rot} + AUC_{Sub}}{3}, 90) \tag{3.1}$$

This equation determine rival's accuracy in the first operation block, for the second and the third, p_{rival} was chosen to be increased by 2.5% and 5% to account for a potential learning effect.

Numbers under the artifact bar stated the score for the participants and the rival. The player received a point only if the correct class was detected, which was indicated by a non-empty white bar, and the artifact bar was below 50%. These scores served as another element to persuade the participants into the competitive mind.

3.2.6 Evaluation

Similar to the evaluation in the offline simulation, the chance level was calculated

to check whether AdLDA showed the result above the chance level, with 120 trials per class for a 3-class problem. Subsequently, the simulation was done with the **standard** algorithm. The accuracy between 1.5 to 4.5 seconds was extracted from both the online experiment and the simulation to find the median for each participant. The median was then used in the two-way repeated measures ANOVA to check whether the results were significant.

Chapter 4

Result

In this chapter, the results of the offline simulation are provided, followed by the results of the online experiment.

4.1 Offline Simulation

4.1.1 Accuracy curve over trial from the simulation

Figure 4.1 and 4.2 illustrate the grand average of accuracy curve over participants. The first one compares the accuracy curve between algorithms of each section of operation block while the second one compares the accuracy curve between sections of each algorithm. The chance level was determined to be 55.62%, which is plotted as a dash-line in the figures. The calculation was based on the 95% confidence level with 150 trials per class. The shaded area indicates the standard error (SE) over participants. From the accuracy curve of each participant, the median of the accuracy between 1.5 to 4.5 seconds was extracted. The median accuracies are summarized in Table 4.1 for different algorithms and different sections.

In every section, the grand average accuracy of **AdLDA** was higher than the accuracy of the other algorithms. Considered the average over sections in Table 4.1, the average accuracy of **AdLDA** was 77.51 % and SD 0.69, which were higher than the **standard** algorithm at 76.07 % and SD 0.78. For **IWLDA, divCSP-WS** and

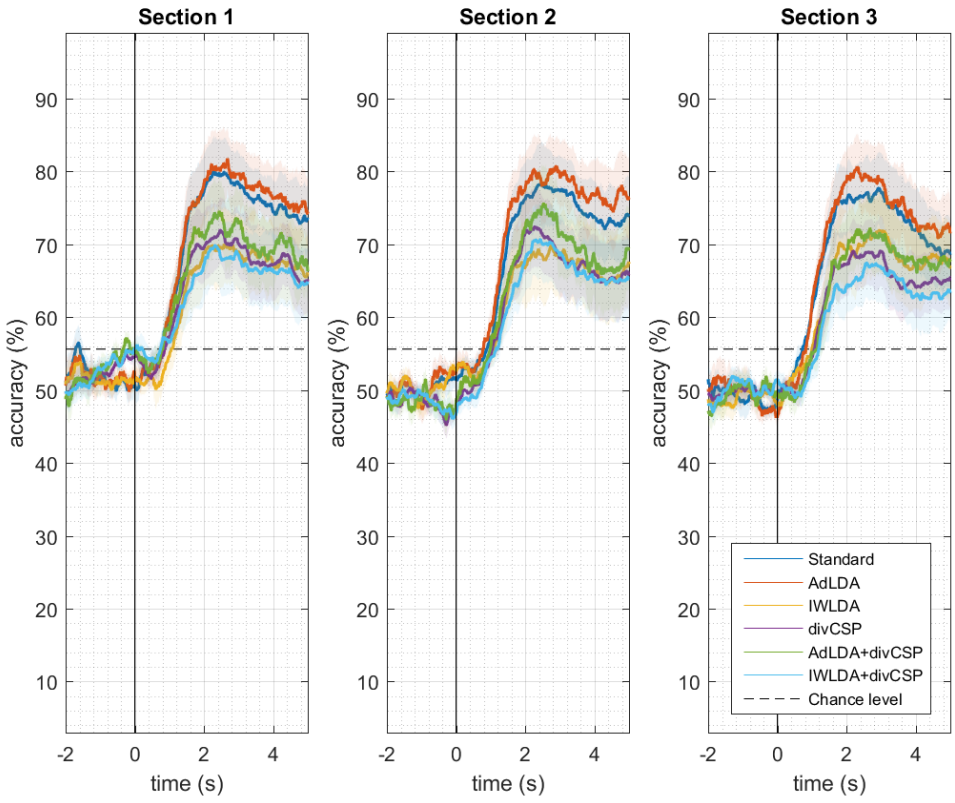


Figure 4.1: The grand average accuracy plot over participants of different algorithms for each section of operation blocks. The shaded area indicate the standard error. Each color represents each algorithm. Dark blue for the **standard** algorithm, red for **AdLDA**, yellow for **IWLDA**, purple for **divCSP-WS**, green for **AdLDA+divCSP-WS**, and light blue for **IWLDA+divCSP-WS**. The black dashed-line represents chance level of 55.62%.

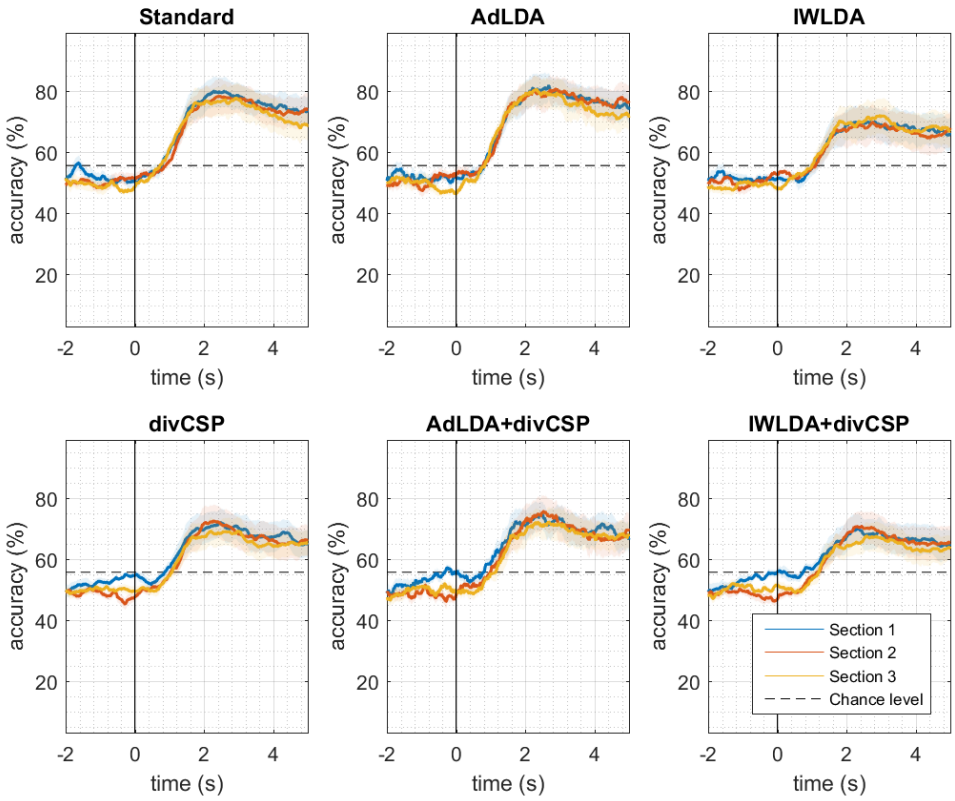


Figure 4.2: The grand average accuracy plot over participant of each section for different algorithms. Each color represents each algorithm. Dark blue for the **standard** algorithm, red for **AdLDA**, yellow for **IWLDA**, purple for **divCSP-WS**, green for **AdLDA+divCSP-WS**, and light blue for **IWLDA+divCSP-WS**. The black dashed-line represents chance level of 55.62%.

Table 4.1: The grand averaged accuracy of the median accuracy of each section and each algorithm.

Algorithms	Section 1	Section 2	Section 3	Average ± SD
Standard	76.98	76.16	75.07	76.07 ± 0.78
AdLDA	77.83	78.15	76.55	77.51 ± 0.69
IWLDA	68.70	67.64	68.92	68.42 ± 0.56
divCSP-WS	69.06	68.20	66.54	67.93 ± 1.04
AdLDA + divCSP-WS	71.36	70.89	69.54	70.59 ± 0.77
IWLDA + divCSP-WS	66.93	67.17	64.93	66.34 ± 1.01

their combinations, they showed lower accuracy than the **standard** algorithm. The average accuracy of **IWLDA** was 67.42 % and SD 0.56, **divCSP-WS** was 67.93 % and SD 1.04, **AdLDA + divCSP-WS** was 70.59 % and SD 0.77, and **IWLDA + divCSP-WS** was 66.34 % and SD 1.01). This effect can be seen in both Table 4.1 and Figure 4.1.

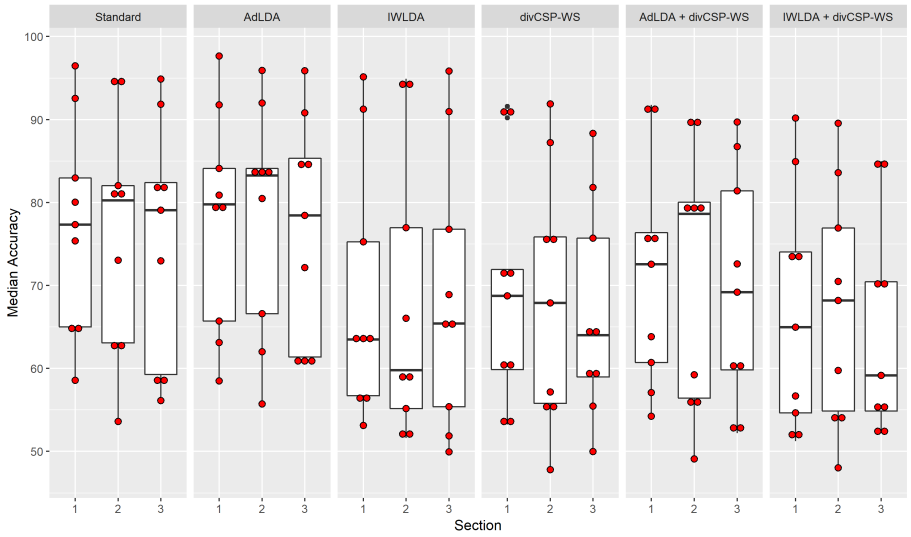


Figure 4.3: The boxplot of the extracted median of every participant over both algorithm and section factor. The red dots are the median extracted from each participant.

Compared the accuracy curve between sections as shown in Figure 4.2 and Table

4.1, the differences were small. In case of the **standard** algorithm, there were a decreasing trend (76.98%, 76.16%, and 75.07%). The same decreasing trend was observable in **divCSP-WS** and **AdLDA+divCSP-WS**, however, the differences were also small.

The boxplots in Figure 4.3 show that distribution of algorithm performance across participants for three sections, with the red dot as the median accuracy of each participant. The same observation can be made that, **AdLDA** showed higher performace than the others, **IWLDA** and **divCSP-WS** and its variations showed lower performance than the **standard** algorithm.

4.1.2 Statistical tests

A two-way repeated-measures ANOVA (ANalysis Of VAriance) was performed to test whether the differences between algorithms and between sections and the interaction between algorithm factor and section factor were significant. Before that, Mauchly's test was done to check if the sphericity assumption was violated. The results from the Mauchly's test were that

- **Algorithm factor**: $W = 6 \times 10^{-5}, p = 9.31 \times 10^{-5}$
- **Section factor**: $W = 0.61, p = 0.17$

which suggested that only the algorithms factor violated the sphericity assumption at the significant level of $\alpha = 0.05$.

To compensate for the violation, the p-value and the degree of freedom from the ANOVA of the algorithm factor were corrected according to Greenhouse-Geisser correction with $\epsilon = 0.37$.

The results from ANOVA were

- **Algorithm factor**: $F(1.85, 14.8) = 8.26, p = 4.4 \times 10^{-3}$
- **Section factor**: $F(2, 16) = 0.94, p = 0.39$
- **Interaction**: $F(10, 80) = 0.78, p = 0.62$

It indicated that only the differences for the algorithm factor were statistically significant at $\alpha = 0.05$. The section factor and the interaction, did not show significant results.

Subsequently, post-hoc tests were done to identify the significant difference for the algorithm factor. Pairwise Wilcoxon signed-rank tests were performed with multiple comparison correction by controlling the false discovery rate (FDR).

Table 4.2 presents the p-values of the tests. The results indicated that the paired differences between **IWLDA** and **divCSP-WS**, **AdLDA** + **divCSP-WS**, and **IWLDA** + **divCSP-WS** were not significant while the other differences were all significant.

Table 4.2: P-value from post-hoc test to identify the significant paired results at $\alpha = 0.05$. These p-values were corrected for multiple comparison by controlling the FDR.

	Standard	AdLDA	IWLDA	divCSP-WS	Ad + div
AdLDA	$6.3 \times 10^{-3}*$	-	-	-	-
IWLDA	$3.1 \times 10^{-7}*$	$1.8 \times 10^{-6}*$	-	-	-
divCSP-WS	$3.7 \times 10^{-7}*$	$7.5 \times 10^{-8}*$	0.93	-	-
Ad + div	$1.6 \times 10^{-5}*$	$1.1 \times 10^{-7}*$	0.48	$1.6 \times 10^{-4}*$	-
IW + div	$7.5 \times 10^{-8}*$	$7.5 \times 10^{-8}*$	0.37	0.03*	$1.0 \times 10^{-5}*$

In summary, the accuracy curve in Figure 4.1 indicated that **AdLDA** exhibited better performance than the other algorithms. The two-way repeated-measures ANOVA further revealed that the effect of the algorithms was significant. Then, the post-hoc tests demonstrated that the differences between **AdLDA** and the **standard** algorithm was significance at $\alpha = 0.05$. This led to the conclusion that **AdLDA** was chosen as the algorithm to be utilized in the online experiment.

4.2 Online Experiment

4.2.1 Accuracy curve over trial from the online experiment

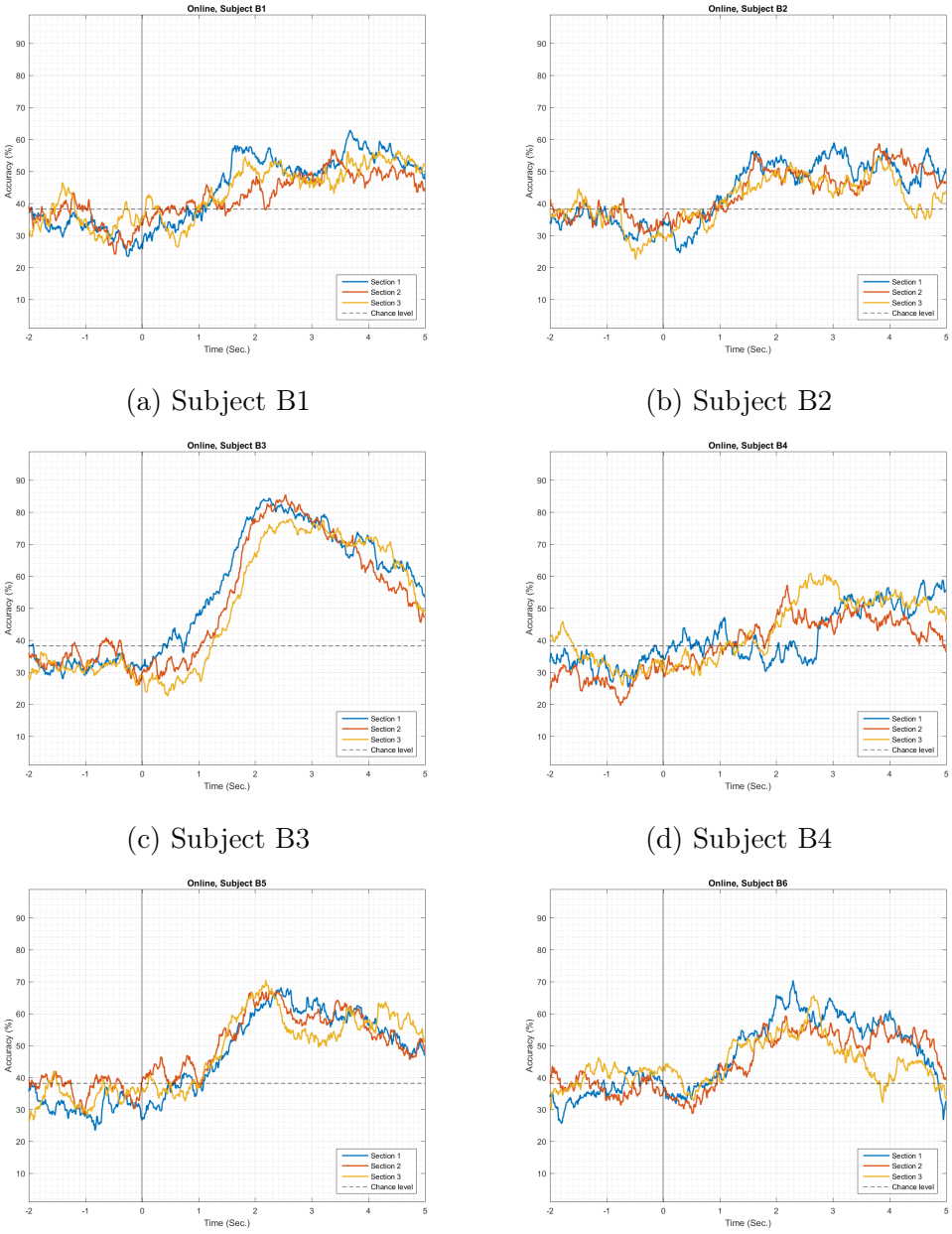
The accuracy curves of the online experiment for each participant are visualized in Figure 4.4 and 4.5. From the plots, participants can be roughly divided into three groups; the low performance group, the moderate performance group, and the high performance group. The high performance group consisted of participants B3, B5, B6, B7, B8, B9. The moderate performance group consisted of B1, B2, B4 and the low performance group consisted of B10.

The low performance group showed accuracy curves that were at around the chance level at 38.18% for 3-class classification at 150 trials per class, without a rise after the cue at time 0. The moderate performance group showed accuracy curves that developed an increase about 10% higher than the chance level after the cue at time 0. The high performance group showed accuracy curves that had a bump more than 10 % higher than the chance level.

Table 4.3: The median value extracted from the accuracy plot between 1.5 to 4.5 seconds along with the initial rival accuracy used in the first operation block for each participant.

Participant	Sect. 1	Sect. 2	Sect.3	Rival accuracy	Group
B1	53.98	47.65	49.67	73.33	Moderate
B2	52.12	49.62	47.25	79.33	Moderate
B3	75.24	72.27	71.99	90	High
B4	47.28	45.92	52.21	71	Moderate
B5	60.04	59.14	57.46	81.3	High
B6	57.97	53.09	48.98	83.6	High
B7	69.68	64.35	75.23	90	High
B8	61.34	60.97	56.61	88.3	High
B9	51.61	48.25	60.94	90	High
B10	39.68	39.99	38.78	66	Low

The accuracies between time 1.5 to 4.5 seconds are summarized in Table 4.3 for each participant, as well as the initial rival accuracy in the first section of operation



(a) Subject B1

(b) Subject B2

(c) Subject B3

(d) Subject B4

(e) Subject B5

(f) Subject B6

Figure 4.4: Accuracy plot from online experiment of each section for participant B1-B6. The blue, red, and yellow lines represent section 1, 2, and 3, respectively. The dashed line means chance level which calculated to be 38.18%.

(a) Subject B7

(b) Subject B8

(c) Subject B9

(d) Subject B10

Figure 4.5: Accuracy plot from online experiment of each section for participant B7-B10. The blue, red, and yellow lines represent section 1, 2, and 3, respectively. The dashed line means chance level which calculated to be 38.18%.

block. There were 3 participants B3, B7, and B9, from the high performance group that competed with rival at 90% accuracy. The lowest rival accuracy was from B10 at 66 %. By comparing the median values in table 4.3 to the chance level, only one participant showed accuracy curves lower than the chance level.

4.2.2 Comparison to the standard algorithm

An offline simulation was done to see how the **standard** algorithms would have performed, compared to **AdLDA** in the online experiment.

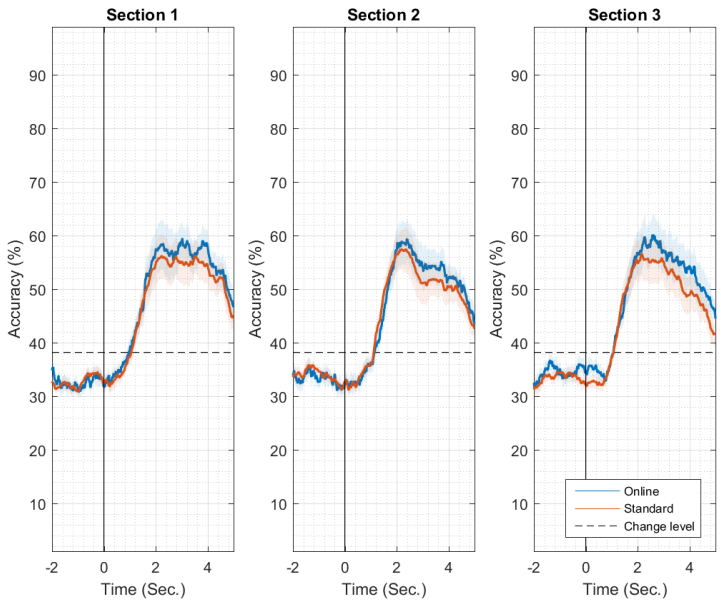


Figure 4.6: The grand average accuracy plots comparing between the results from the online experiment with **AdLDA** and the simulation with the **standard** algorithm for each section. The blue line denotes the result from the online experiment, while the orange line represents the result of the offline simulation with the **standard** algorithm. The shaded area represents the standard error (SE) of each algorithm. The dashed black line indicates the chance level at 38.18%.

The results are shown in Figure 4.6 and 4.7. Figure 4.6 visualizes the accuracy curves of both algorithms. Figure 4.7 visualizes the accuracy curves that compare the

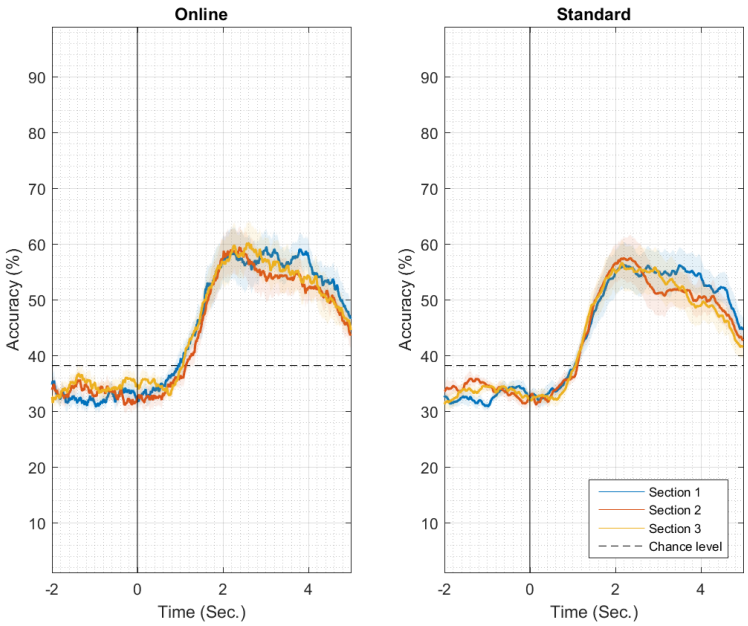


Figure 4.7: The grand average accuracy plots comparing between sections for both the online experiment and the **standard** algorithm. Each color denotes each section. The shaded area represents the standard error (SE) of each algorithm. The dashed black line indicates the chance level at 38.18%.

Table 4.4: The average over participant of the median extracted from both online experiment and offline simulation.

Algorithms	Section 1	Section 2	Section 3	Mean ± SD
Online	56.89	54.13	55.91	55.64 ± 1.14
Standard	54.26	52.26	53.09	53.20 ± 0.82

accuracy curve for each section. The results from figure 4.6 and 4.7 were summarized in table 4.4. It was done by computing the median of the accuracy between 1.5 to 4.5 seconds. From the table, the average of median accuracy over sections of **AdLDA** was 55.64% with SD 1.14, which was 2.44 % higher than the accuracy of the **standard** algorithm, which was 53.20 with SD 0.82.

Both algorithms exhibited similar behavior with **AdLDA** above the **standard**

algorithm. The accuracy curve was flater in section 1 than section 2, and 3. For section 2, There were small bumps occurred at time point 2, 3, and 4 seconds for **AdLDA**, which were the three time points that were chosen in the training of the classification model. In section 2, the bump characteristic was more pronounced than section 2 and 3, in both algorithms.

4.2.3 Statistical tests

Similar to the offline simulation, the two-way repeated-measures ANOVA was performed with algorithm and section factors. Mauchly's test was done beforehand to examine if the sphericity assumption was violated. Since the algorithm factor contained only two algorithms, there was no need to check for sphericity assumption. The result from the Mauchly's test was that the sphericity assumption was violated in the section factor ($W = 0.42, p = 0.03$). Hence, the result from the ANOVA test of the section factor must be corrected. The corrected result from ANOVA was

- **Algorithm factor**: F(1, 9) = 10.72, p = 9.6 $\times 10^{-8}$
- **Section factor**: F(0.6, 5.4) = 1.24, p = 0.30
- **Interaction**: F(2, 18) = 0.67, p = 0.52

where the correction was done according to the Greenhouse-Geisser correction with $\epsilon = 0.30$ as a compensation of the violation from the section factor. The algorithm factor was the only factor with a significant result. Because there were only two algorithms, the post-hoc test was not required in this case.

That is, the accuracy of **AdLDA** was significantly higher than the **standard** algorithm. The differences between sections, however, was not significant.

4.2.4 Topography of the LBP features

The LBP features from every participants were used to compute topography plots as visualized in figure 4.8.

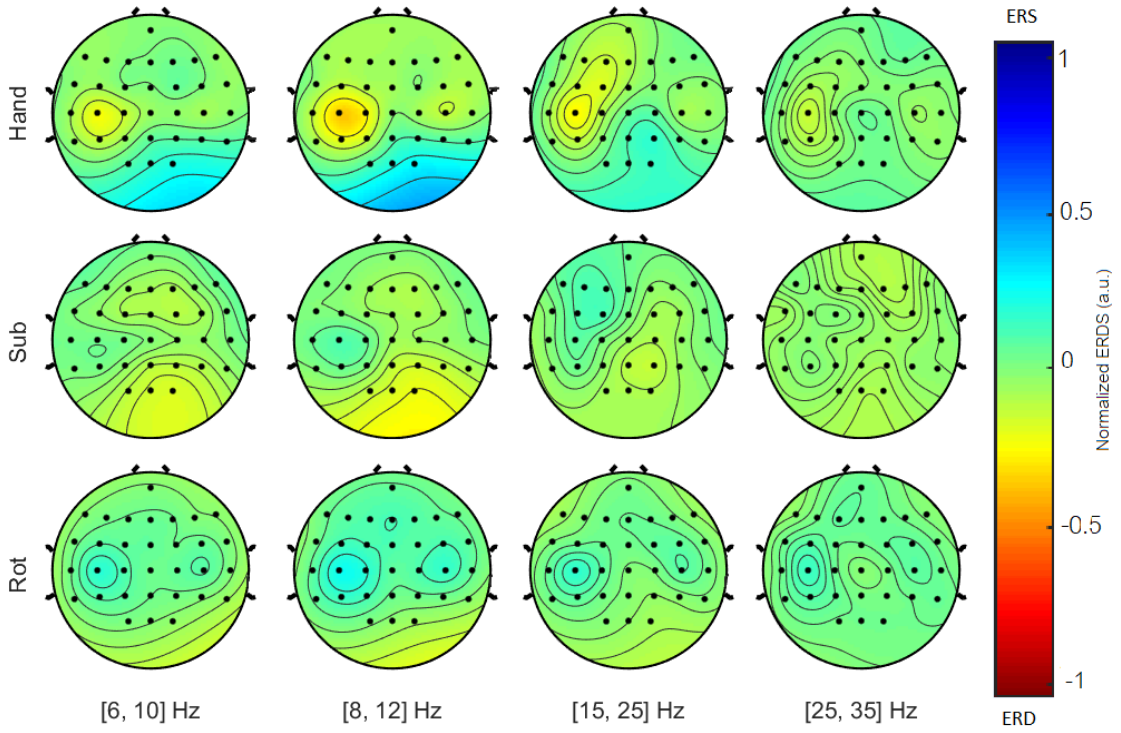


Figure 4.8: The grand average topological features plot over participants. Red represents ERD and blue represents ERS.

It shows a common pattern among participants of each class. The right hand motor imagery showed ERD pattern narrowed at channel C3 for the LBP features from 8 to 12 Hz which was the most distinctive features among the other classes. For mental subtraction, slight ERS were visible around C3 from 8 - 12 Hz and in the frontal part at 15 - 25 Hz. The common pattern for mental rotations were small ERS at around C3, and C4. The topography plot of individual participant can be found in A3

Chapter 5

Discussion

Offline simulation

The results from the offline simulation showed that the median accuracy of **AdLDA** was higher than the **standard** algorithm by 1.44 %, while the other algorithms did not improve over the **standard** algorithm. Two-way repeated measures ANOVA confirmed that the differences between algorithms were significant. Then, Pairwise Wilcoxon signed-rank tests showed that the differences between **AdLDA** and the **standard** algorithm were significant. Finally, **AdLDA** was chosen for the online experiment.

Besides the comparison between algorithms, the median accuracy from the offline simulation with **AdLDA**, which is an unsupervised adaptation, was compared with the original results of Brandstetter [28], which is a semi-supervised adaptation. The mean of **AdLDA** was 77.51 %, SD 13.24 and the mean of the semi-supervised adaptation was 77.81 %, SD 14.19. This indicated that **AdLDA** is comparable with the semi-supervised adaptation.

Why did IWLDA and divCSP-WS fail?

First, for **IWLDA**, it was demonstrated in appendix A1 that the importance estimated with KLIEP could lead to a less effective data point, that was used in the training period, and in the extreme case where the distribution of the training and the testing data were too far apart, even fewer samples were effective in the training process of the **IWLDA** model. This might cause a problem that the classifier from **IWLDA** was biased, which resulted in the classifier that could not compensate to the changes from the non-stationarities. This problem can be avoided if the training data was to be updated with the new data so that the distributions were not too far away, but this was not possible considering the restrictions in the simulation.

Second, the problem with **divCSP-WS** could be due to the restrictions in the simulation scenario. For **divCSP-WS** to work the best, the training data has to be updated so that the training data covered the new change from the new data. In the simulation scenario, however, the training data only covered the data from the calibration. This resulted to a model of **divCSP-WS** that cannot compensate for large changes from the longer period. The update of the training data was not possible because the label is needed in the recalculation of the CSP filters. Another factor could be that the parameters in the training of the CSP filter were not optimal for every participant, since the same set of parameters was used for every participant. The improvement could be made if an additional cross-validation was done to determine the optimum parameters. However, the time required for finding the optimum parameter might be tedious which was not practical with current usage of the system.

Online experiment

The results from the statistical tests from the offline simulation revealed that only the algorithm factor was significantly different, so other factors were added that might induce more of the non-stationarity effects by introducing a rival player that the participants competed with. Furthermore, the scoring system was also introduced

as a way to force participants to compare their performance to those from the rival player.

The results from the online experiment were that 9 out of 10 participants were able to perform the task above the chance level. After that, the simulation with the **standard** algorithm was done. The comparison was that **AdLDA** showed mean accuracy higher than the **standard** algorithm by 2.44 %. Then, two-way repeated measures ANOVA was used to test the significant results. It revealed that the differences between algorithms were significant but the differences between sections were not. By comparing the differences between the **standard** algorithm to **AdLDA**, in both the offline simulation (1.44%) and the online experiment (2.44%), the effect from the rival can be negligible. This could be due to an error in the calculation of the rival accuracy, which caused changes of the state of mind which was not enough to induce noticeable effects of the psychophysiological factor of the non-stationarities.

Problems with the online experiment

One of the problem with the rival accuracy was that it was calculated too high, in comparison to the accuracy of the participant as seen in Table 4.3. This is because the rival accuracy was mistakenly calculated based on individual average AUC score, which is not the same quantity as the accuracy.

The other problem with the experiment was that the accuracy curves of **AdLDA** were noisier than the **standard** algorithm. To identify the cause of this noisy behavior, the simulation was done with **AdLDA**. Figure A.2 showed a comparison between the online experiment with **AdLDA**, the offline simulation with the **standard** algorithm, and the offline simulation with **AdLDA**. The offline simulation with **AdLDA** showed less noisy curve than the online experiment, which indicated that the noisy curve was due to the differences between online and offline scenarios. This could be because of the chunk size of the EEG signal was fixed in the offline simulation but the chunk size was not constant during the online experiment.

Topography plot of the LBP features

The grand average of LBP features was calculated to see a common pattern among participants as mentioned in section 4.2.4. The most distinctive pattern was the ERS at channel C3 of the frequency band 8 to 12 Hz. This agreed with ERDS changes of the right hand motor imagery discovered by Pfurtscheller [17]. The pattern of mental subtraction and mental rotation were not as strong as the right hand motor imagery because the brain area associated with these two classes are not specific and might vary between person.

Limitation and improvements

In the end, three algorithms were considered in this thesis with restriction as in the CYBATHLON scenario. However, there could be a better algorithm beside these three algorithms, that could improve the performance of the system even more. Interesting algorithms are, for example, applying the importance weighting to CSP [59], or replacing LDA with an adaptive support vector machine (SVM) [60]. The change of the system was not limited to just the algorithm, but some parts of the system were implemented in a way that changes can be made without much effort. For example, the number of frequency band features as well as the cutoff frequency. Furthermore, if the implementation needed an interface with an additional hardware, this can be done easily due to the flexibility of the LSL system.

Chapter 6

Conclusion

The goal of the thesis was to find the best algorithm that addresses the within-session non-stationarities to be implemented into the Graz BCI racing team system while considering the CYBATHLON scenario. The new BCI was implemented based on the Graz BCI racing team predecessor system with sCSP and sLDA as the **standard** algorithm.

The results from both the offline simulation and the online experiment indicated that **AdLDA** improved the accuracy over the **standard** algorithm by 1.44 % and 2.44 %, respectively. Finally, **AdLDA** was implemented in the Graz BCI racing team system.

The investigations of different algorithms are not only beneficial to the Graz BCI racing team but also to the BCI research in general. This is because the within-session non-stationarities are common in BCI application and are needed to be handled eventually if BCI were to be used outside the laboratory environment. The conclusion that **AdLDA** was the final algorithm is also interesting since an unsupervised adaptation will be an interesting way of development, especially toward an active BCI which the user can controll BCI at will, without the cues.

Bibliography

[1] H. Berger, "Über das elektrenkephalogramm des menschen," *Archiv für Psychiatrie und Nervenkrankheiten*, vol. 87, no. 1, pp. 527–570, Dec. 1929, ISSN: 1433-8491. DOI: 10.1007/BF01797193.

[2] S. J. M. Smith, "Eeg in the diagnosis, classification, and management of patients with epilepsy," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, no. suppl 2, pp. ii2–ii7, 2005, ISSN: 0022-3050. DOI: 10.1136/jnnp.2005.069245.

[3] F. Vogel and E. Schalt, "The electroencephalogram (eeg) as a research tool in human behavior genetics: Psychological examinations in healthy males with various inherited eeg variants," *Human Genetics*, vol. 47, no. 1, pp. 81–111, Jan. 1979, ISSN: 1432-1203. DOI: 10.1007/BF00295571.

[4] T. Morita, M. Asada, and E. Naito, "Contribution of neuroimaging studies to understanding development of human cognitive brain functions," *Frontiers in Human Neuroscience*, vol. 10, p. 464, 2016, ISSN: 1662-5161. DOI: 10.3389/fnhum.2016.00464.

[5] R. P. N. Rao, *Brain-Computer Interfacing: An Introduction*. USA: Cambridge University Press, 2013, ISBN: 1139032801.

[6] E. Jungnickel and K. Gramann, "Mobile brain/body imaging (mobi) of physical interaction with dynamically moving objects," *Frontiers in Human Neuroscience*, vol. 10, p. 306, 2016, ISSN: 1662-5161. DOI: 10.3389/fnhum.2016.00306.

[7] J. J. Vidal, "Toward direct brain-computer communication," *Annual Review of Biophysics and Bioengineering*, vol. 2, no. 1, pp. 157–180, 1973, PMID: 4583653. DOI: 10.1146/annurev.bb.02.060173.001105.

[8] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002, ISSN: 1388-2457. DOI: https://doi.org/10.1016/S1388-2457(02)00057-3.

[9] L. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988, ISSN: 0013-4694. DOI: https://doi.org/10.1016/0013-4694(88)90149-6.

[10] T. Kaufmann, S. M. Schulz, C. Grünzinger, and A. Kübler, "Flashing characters with famous faces improves ERP-based brain–computer interface performance," *Journal of Neural Engineering*, vol. 8, no. 5, p. 056 016, Sep. 2011. DOI: 10.1088/1741-2560/8/5/056016.

[11] K. Statthaler, A. Schwarz, D. Steyrl, R. Kobler, M. Höller, J. Brandstetter, L. Hehenberger, M. Bigga, and G. Müller-Putz, "Cybathlon experiences of the graz bci racing team mirage91 in the brain-computer interface discipline," *Journal of NeuroEngineering and Rehabilitation*, vol. 14, no. 129, Dec. 2017, ISSN: 1743-0003. DOI: 10.1186/s12984-017-0344-9.

[12] L. Hochberg, D. Bacher, B. Jarosiewicz, N. Masse, J. Simeral, J. Vogel, S. Haddadin, J. Liu, S. Cash, P. van der Smagt, and J. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, pp. 372–5, May 2012. DOI: 10.1038/nature11076.

[13] S. A. Huettel and G. McCarthy, "What is odd in the oddball task?: Prefrontal cortex is activated by dynamic changes in response strategy," *Neuropsychologia*, vol. 42, no. 3, pp. 379–386, 2004, ISSN: 0028-3932. DOI: https://doi.org/10.1016/j.neuropsychologia.2003.07.009.

[14] *Somatic sensory and somatic motor maps in the cerebral cortex*, http://highheredbcs.wiley.com/legacy/college/tortora/0470565101/hearthis_ill/pap13e_ch16_illustr_audio_mp3_am/simulations/hear/maps.html Last accessed : 2019-12-20.

[15] J. Ward, *The student's guide to cognitive neuroscience*, 3rd ed. Psychology Press, 2015, ISBN: 978-1-84872-272-9.

[16] G. Pfurtscheller and A. Aranibar, "Event-related cortical desynchronization detected by power measurements of scalp eeg," *Electroencephalography and Clinical Neurophysiology*, vol. 42, no. 6, pp. 817–826, 1977, ISSN: 0013-4694. DOI: 10.1016/0013-4694(77)90235-8.

[17] G. Pfurtscheller, "Event-related synchronization (ers): An electrophysiological correlate of cortical areas at rest," *Electroencephalography and Clinical Neurophysiology*, vol. 83, pp. 62–69, 1992.

[18] G. Pfurtscheller and F. L. da Silva, "Event-related eeg/meg synchronization and desynchronization: Basic principles," *Clinical Neurophysiology*, vol. 110, pp. 1842–1857, 1999.

[19] B. Graimann, J. Huggins, S. Levine, and G. Pfurtscheller, "Visualization of significant erd/ers patterns in multichannel eeg and ecog data.," *Clinical Neurophysiology*, vol. 1(41), pp. 43–47, 2002, ISSN: 1388-2457.

[20] C. M. Stinear, W. D. Byblow, M. Steyvers, O. Levin, and S. P. Swinnen, "Kinesthetic, but not visual, motor imagery modulates corticomotor excitability," *Experimental Brain Research*, vol. 168, pp. 157–164, 2006, ISSN: 1432-1106. DOI: 10.1007/s00221-005-0078-y.

[21] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller, "Imagery of motor actions: Differential effects of kinesthetic and visual-motor mode of imagery in single-trial eeg," English, *Brain Research / Cognitive Brain Research*, vol. 25, no. 3, pp. 668–677, 2005, ISSN: 0926-6410. DOI: 10.1016/j.cogbrainres.2005.08.014.

[22] E. Friedrich, C. Neuper, and R. Scherer, "Whatever works: A systematic user-centered training protocol to optimize brain-computer interfacing individually," *PLoS ONE*, pp. 1–11, 2013, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0076214.

[23] M. Krauledat, "Analysis of nonstationarities in eeg signals for improving brain-computer interface performance," PhD thesis, Jan. 2008. DOI: 10.14279/depositonce-1817.

[24] C. Vidaurre, M. Kawanabe, P. von Bünau, B. Blankertz, and K. R. Müller, "Toward unsupervised adaptation of lda for brain–computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 587–597, 2011, ISSN: 1558-2531. DOI: 10.1109/TBME.2010.2093133.

[25] B. Blankertz, M. Kawanabe, R. Tomioka, F. Hohlefeld, K.-R. Müller, and V. V. Nikulin, "Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., Curran Associates, Inc., 2008, pp. 113–120.

[26] G. Müller-Putz, A. Schwarz, and D. Steyrl, "Mirage91: The graz bci-racing team – making students familiar with bci research," 6th International Brain-Computer Interface Meeting 2016 ; Conference date: 30-05-2016 Through 03-06-2016, 2016.

[27] R. Riener, "The cybathlon promotes the development of assistive technology for people with physical disabilities," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 49, 2016. DOI: 10.1186/s12984-016-0157-2.

[28] J. Brandstetter, "Online adaptation of a brain computer interface using semi-supervised learning techniques," Master's thesis, Graz University of Technology, Graz, Austria, 2017.

[29] C. Breitwieser, C. Neuper, and G. Müller-Putz, "A concept to standardize raw biosignal transmission for brain-computer interfaces," in *Annual International*

Conference of the IEEE Engineering in Medicine and Biology Society, ., 2011, pp. 6377–6380, ISBN: 978-1-4244-4122-8.

[30] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K. Muller, "Single-trial analysis and classification of erp components – a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, May 2011, ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2010.06.048.

[31] B. Reuderink and M. Poel, *Robustness of the Common Spatial Patterns algorithm in the BCI-pipeline*, Undefined, ser. CTIT Technical Report Series DTR08-9/TR-CTIT-08-52. Netherlands: Centre for Telematics and Information Technology (CTIT), Jul. 2008.

[32] M. Wolf and O. Ledoit, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of Multivariate Analysis*, vol. 88, no. 2, pp. 365–411, 2004, ISSN: 0047-259X. DOI: 10.1016/S0047-259X(03)00096-4.

[33] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-r. Muller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008. DOI: 10.1109/msp.2008.4408441.

[34] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial eeg during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, pp. 441–446, 2000, ISSN: 1063-6528.

[35] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek, "Optimizing the channel selection and classification accuracy in eeg-based bci," *IEEE Transactions on Biomedical Engineering*, vol. 58, pp. 1865–1873, 2011.

[36] K. Ang, Z. 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, pp. 2390–2397. DOI: 10.1109/IJCNN.2008.4634130.

[37] F. Popescu, S. Fazli, Y. Badower, B. Blankertz, and K.-R. Müller, "Single trial classification of motor imagination using 6 dry eeg electrodes," *PLOS ONE*, vol. 2, no. 7, pp. 1–5, Jul. 2007. DOI: 10.1371/journal.pone.0000637.

[38] M. Naeem, C. Brunner, and G. Pfurtscheller, "Dimensionality reduction and channel selection of motor imagery electroencephalographic data," *Computational intelligence and neuroscience*, vol. 2009, pp. 1–8, 2009, ISSN: 1687-5265. DOI: 10.1155/2009/537504.

[39] N. Hill, T. Lal, M. Tangermann, T. Hinterberger, G. Widman, C. Elger, B. Schölkopf, and N. Birbaumer, "Classifying event-related desynchronization in eeg, ecog and meg signals," Sep. 2006, pp. 404–413. DOI: 10.1007/11861898_41.

[40] M. E. Sharbaf, A. Fallah, and S. Rashidi, "Shrinkage estimator based common spatial pattern for multi-class motor imagery classification by hybrid classifier," in *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 2017, pp. 26–31. DOI: 10.1109/PRIA.2017.7983059.

[41] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve bci designs: Unified theory and new algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2011, ISSN: 1558-2531. DOI: 10.1109/TBME.2010.2082539.

[42] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387310738.

[43] R. Peck and J. Van Ness, "The use of shrinkage estimators in linear discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 5, pp. 530–537, 1982, ISSN: 1939-3539. DOI: 10.1109/TPAMI.1982.4767298.

[44] A. Mkhadri, "Shrinkage parameter for the modified linear discriminant analysis," *Pattern Recognition Letters*, vol. 16, no. 3, pp. 267–275, 1995, ISSN: 0167-8655. DOI: 10.1016/0167-8655(94)00100-H.

[45] M. Kawanabe, M. Krauledat, and B. Blankertz, "A bayesian approach for adaptive bci classification," in *In Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course 2006*, 2006.

[46] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, Dec. 2008, ISSN: 1572-9052. DOI: 10.1007/s10463-008-0197-x.

[47] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, 2007, ISSN: 1532-4435.

[48] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000, ISSN: 0378-3758. DOI: 10.1016/S0378-3758(00)00115-4.

[49] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, "Application of covariate shift adaptation techniques in brain–computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 6, pp. 1318–1324, 2010, ISSN: 1558-2531. DOI: 10.1109/TBME.2009.2039997.

[50] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951. DOI: 10.1214/aoms/1177729694.

[51] W. Samek, D. Blythe, K.-R. Müller, and M. Kawanabe, "Robust spatial filtering with beta divergence," in *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., 2013, pp. 1007–1015.

[52] W. Samek, M. Kawanabe, and K. Müller, "Divergence-based framework for common spatial patterns algorithms," *IEEE Reviews in Biomedical Engineering*, vol. 7, pp. 50–72, 2014, ISSN: 1941-1189. DOI: 10.1109/RBME.2013.2290621.

[53] A. Schwarz, J. Brandstetter, J. Pereira, and G. R. Müller-Putz, "Direct comparison of supervised and semi-supervised retraining approaches for co-adaptive bcis," *Medical & Biological Engineering & Computing*, vol. 57, no. 11, pp. 2347–2357, 2019, ISSN: 1741-0444. DOI: 10.1007/s11517-019-02047-1.

[54] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001, ISSN: 1558-2256. DOI: 10.1109/5.939829.

[55] A. Delorme and S. Makeig, "Eeglab: An open source toolbox for analysis of single-trial eeg dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, 2004, ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2003.10.009.

[56] K. A. Ludwig, R. M. Miriani, N. B. Langhals, M. D. Joseph, D. J. Anderson, and D. R. Kipke, "Using a common average reference to improve cortical neuron recordings from microelectrode arrays," *Journal of Neurophysiology*, vol. 101, no. 3, pp. 1679–1689, 2009, PMID: 19109453. DOI: 10.1152/jn.90989.2008.

[57] R. Kobler, A. Sburlea, V. Mondini, and G. Müller-Putz, "Hear to remove pops and drifts: The high-variance electrode artifact removal (hear) algorithm," in *Proceedings of the 41th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE Xplore, Jul. 2019.

[58] G. Müller-Putz, R. Scherer, C. Brunner, R. Leeb, and G. Pfurtscheller, "Better than random? a closer look on bci results," *International journal of bioelectromagnetism*, vol. 10, no. 1, pp. 52–55, 2008, ISSN: 1456-7857.

[59] A. Balzi, F. Yger, and M. Sugiyama, "Importance-weighted covariance estimation for robust common spatial pattern," *Pattern Recogn. Lett.*, vol. 68, no. P1, pp. 139–145, Dec. 2015, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2015.09.003.

[60] B. Yang, C. Fan, J. Jia, S. Chen, and J. Wang, "Adaptive kf-svm classification for single trial eeg in bci," in *Advanced Computational Methods in Life System Modeling and Simulation*, M. Fei, S. Ma, X. Li, X. Sun, L. Jia, and Z. Su, Eds., Singapore: Springer Singapore, 2017, pp. 35–45, ISBN: 978-981-10-6370-1.

Appendix A

Appendix

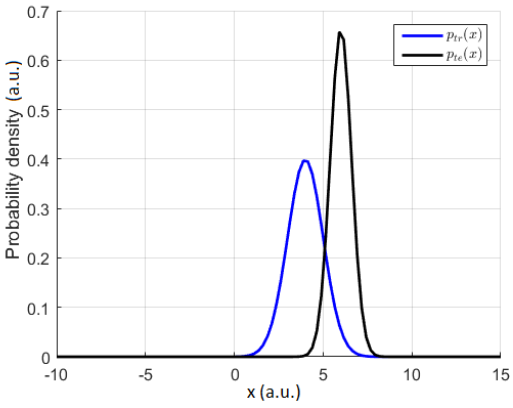
A1 Importance estimation

KLIEP algorithm was used with 100 points that were generated for the training data and 1,000 points for the testing data. Figure A.1 illustrates the results from this investigation. Here 2 cases were compared with the assumption that the data were distributed normally. The first case was when the mean of training and the testing data were chosen to be close together (4 for the training and 6 for the testing data) and the standard deviation of 1 and 0.6 for training and the testing data, respectively. The second case was when the mean of the training was changed to 1, which separated the training and the testing distribution apart while the other parameters were the same.

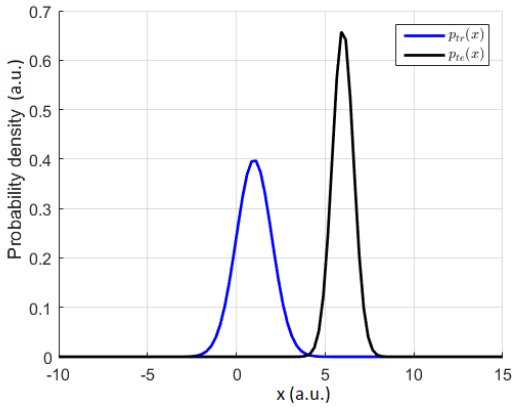
As mentioned before, the importance was modeled as the ratio between the testing data distribution to the training data distribution. Since the data was generated, the actual importance can be directly calculated, and it was shown as the red lines in figure A.1c for case 1 and figure A.1d for case 2. The green lines indicate the estimated importance from the KLIEP algorithm, and the blue circles mean the training data point but plotted with the estimated importance as the y-axis.

In the first case, as visualized in figure A.1c, the estimated importance of most of the point was closed to zero, while only the minority that the importance was not

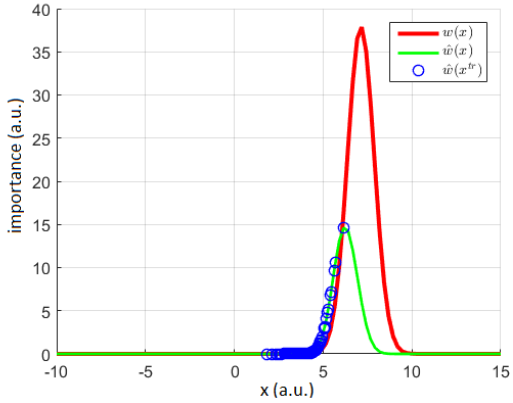
close to zero. In the second case, more data points were close to zero.



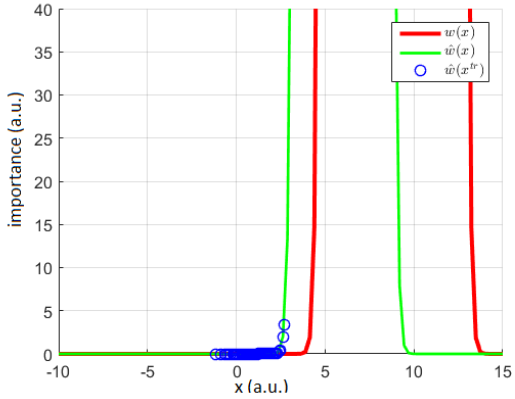
(a) The density plot of case 1



(b) The density plot of case 2



(c) The plot of importance of case 1



(d) The plot of importance of case 2

Figure A.1: The probability density plots and the importance of 2 cases. For the density plot, the blue and black lines indicate the probability density of the training and testing data estimated from 100 points and 1000 points, respectively. For the importance plot, the red line and green line mean the actual importance and the estimated importance from KLIEP algorithm, while the blue circles indicate the training data on the x-axis and the associating importance value on the y-axis. a) the density plot of case 1 where the means are 4, 6 and the standard deviation are 1, 0.6 for training and testing data, b) the density plot of case 2 where the mean are 1, 6 and the standard deviation are 1, 0.6 for training and testing data, c) the importance of case 1 and d) the importance of case 2.

A2 Inspection on the noisy accuracy curve of AdLDA

From the grand average accuracy curve in figure 4.6, the online experiment result exhibited a noisier accuracy curve comparing to the accuracy curve from the offline simulation with the **standard** algorithm. The offline simulation with **AdLDA** was done to see if the noisier behavior was due to the difference between offline and online or not. The results is displayed in figure A.2.

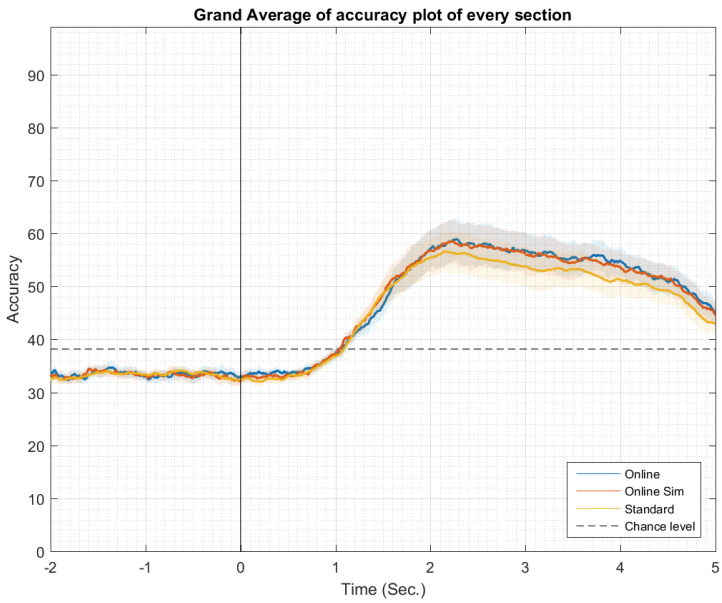
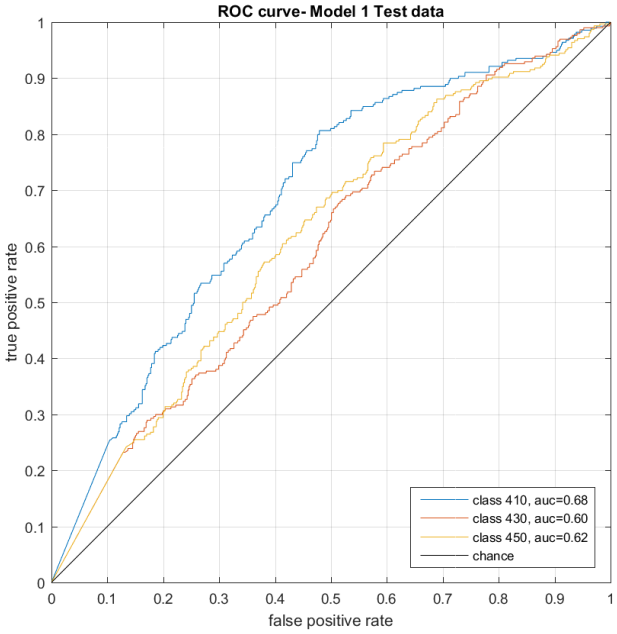


Figure A.2: The accuracy curve comparing between the results from online experiment (Blue), **AdLDA** with offline simulation (red), **standard** algorithm (yellow).

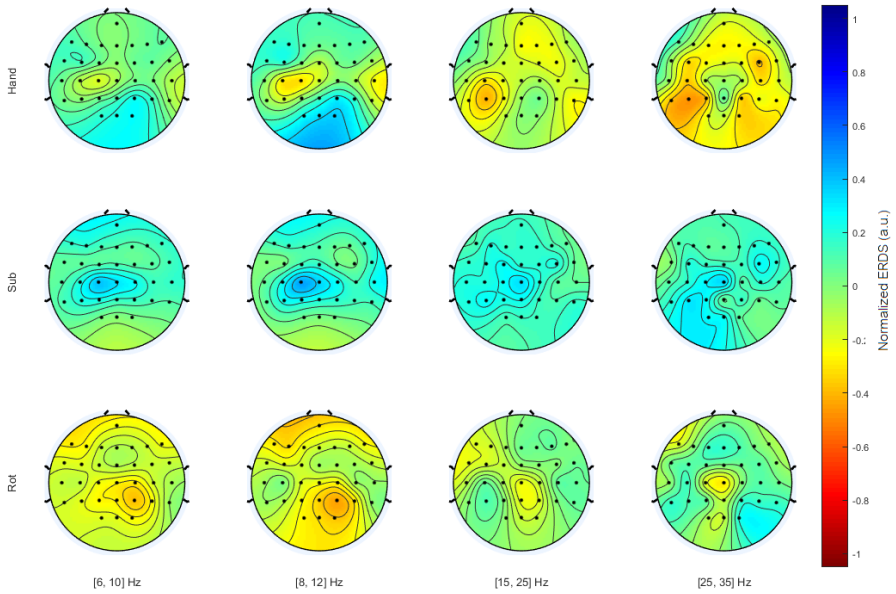
A3 ROC curve and Topography LBP features plot

The ROC curve and the topography generated from the training data of every participant were presented. The blue, yellow, and red lines indicate Hand motor imagery, Mental subtraction, and Mental rotation, respectively ,while the black line represents the chance level. The best case would be when the area under the curve (AUC) is one or that the curve looks similar to a stair step.

The topography plot represents the normalized LBP features of each class and different frequency bands. The blue color indicates an increase of the normalized LBP features associating with the ERS and the red color indicates a decrease of the normalized LBP features associating toe the ERD. The color bar was be shown only in the first participant but the scaling remained the same between 1 and -1.

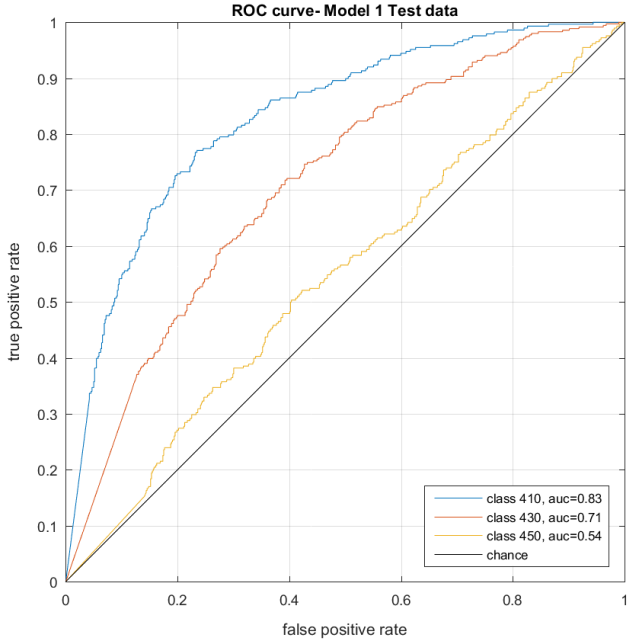


(a) ROC curve of participant B1.

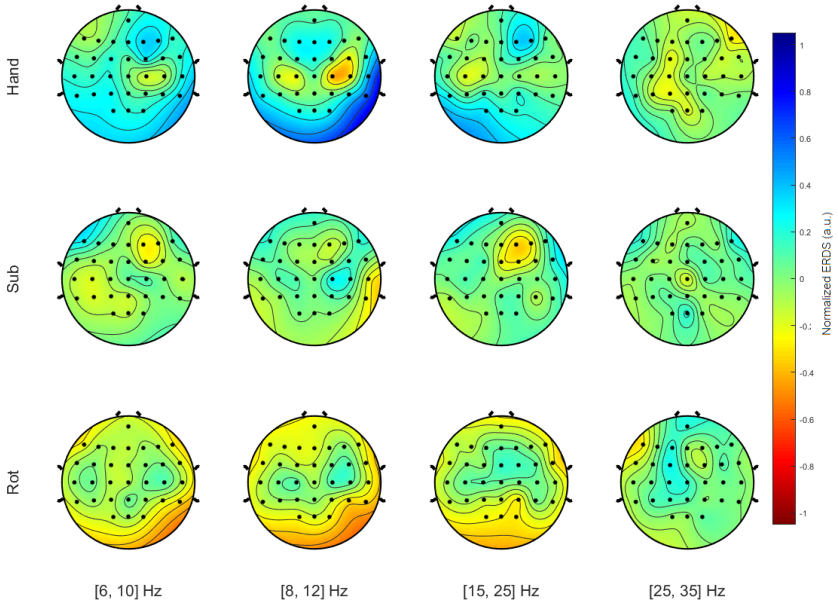


(b) Topography plot of participant B1.

Figure A.3: The plots generated from the data from participant B1.

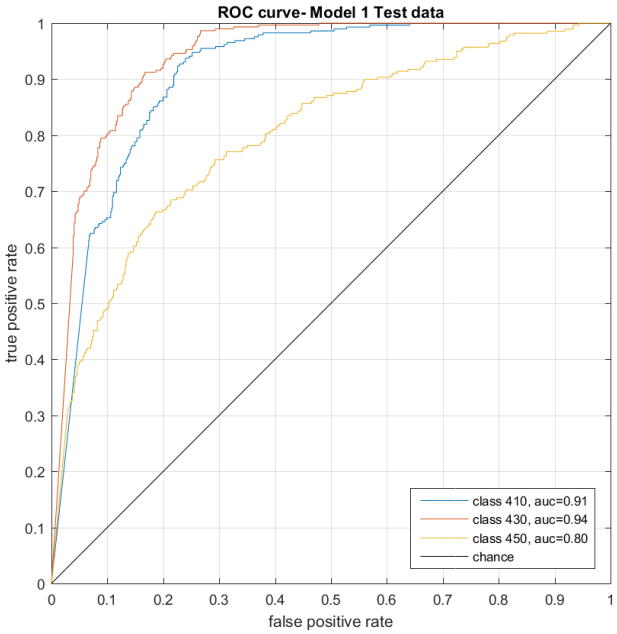


(a) ROC curve of participant B2.

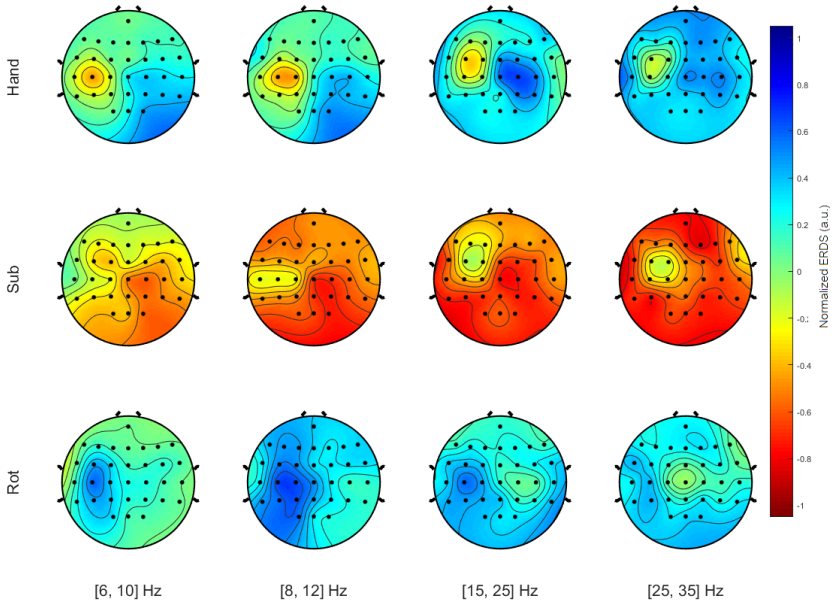


(b) Topography plot of participant B2.

Figure A.4: The plots generated from the data from participant B2.

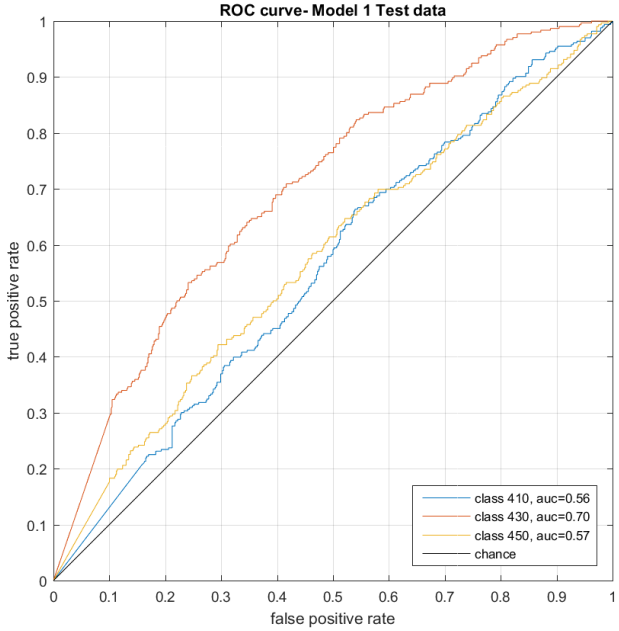


(a) ROC curve of participant B3.

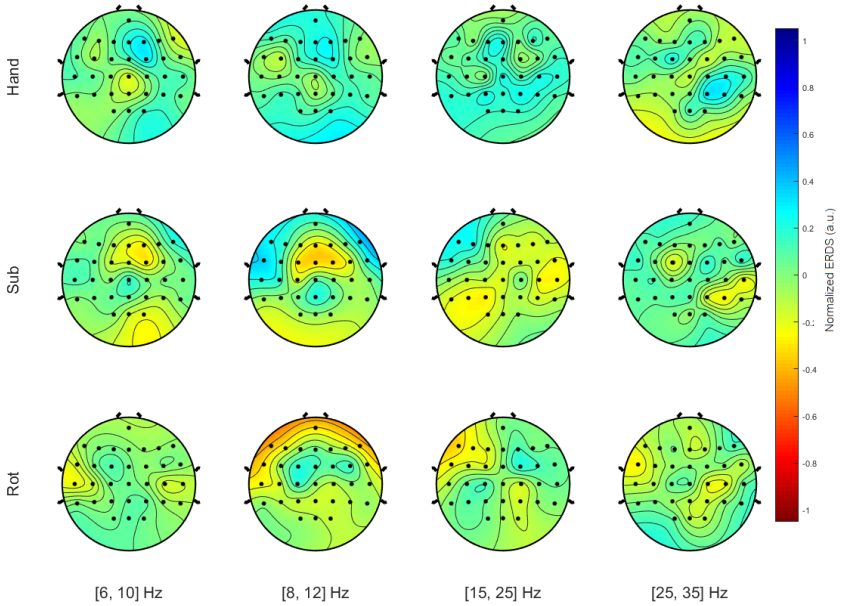


(b) Topography plot of participant B3.

Figure A.5: The plots generated from the data from participant B3.

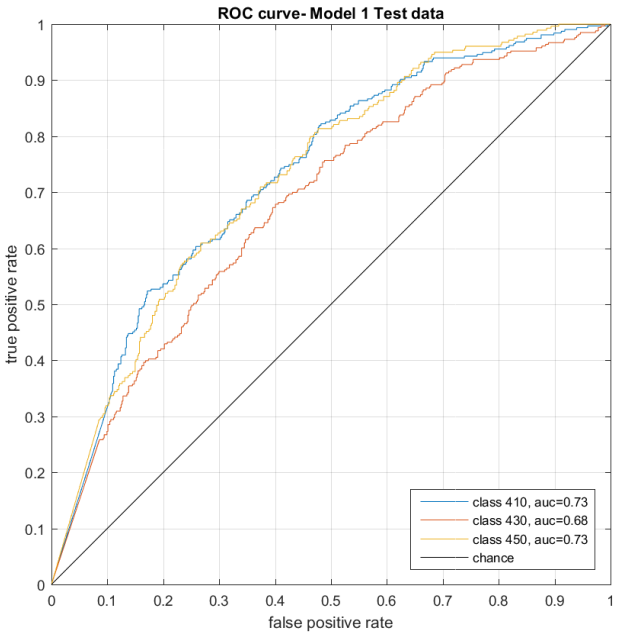


(a) ROC curve of participant B4.

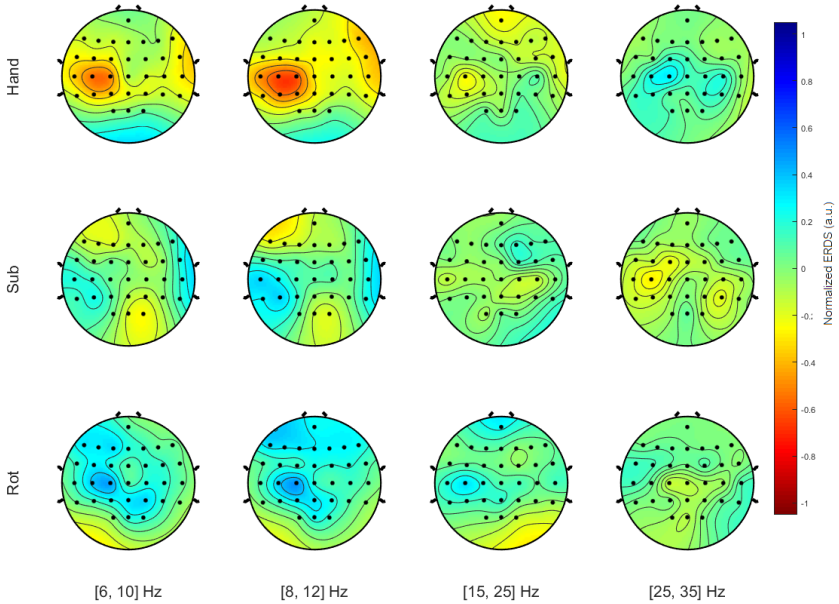


(b) Topography plot of participant B4.

Figure A.6: The plots generated from the data from participant B4.

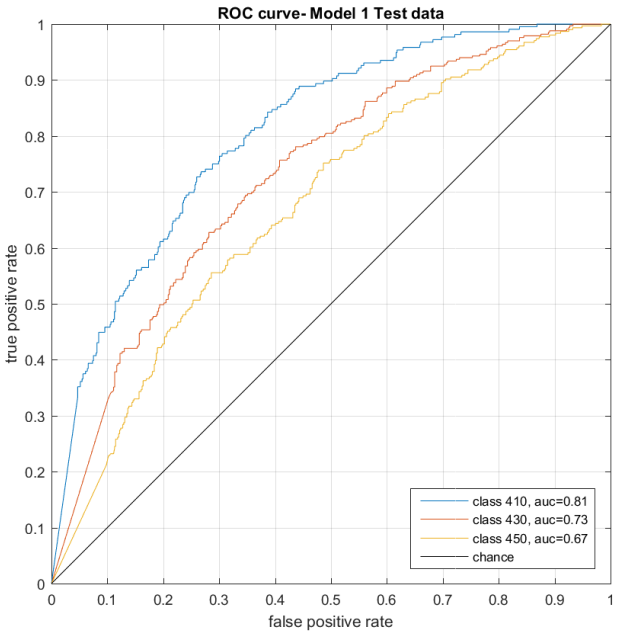


(a) ROC curve of participant B5.

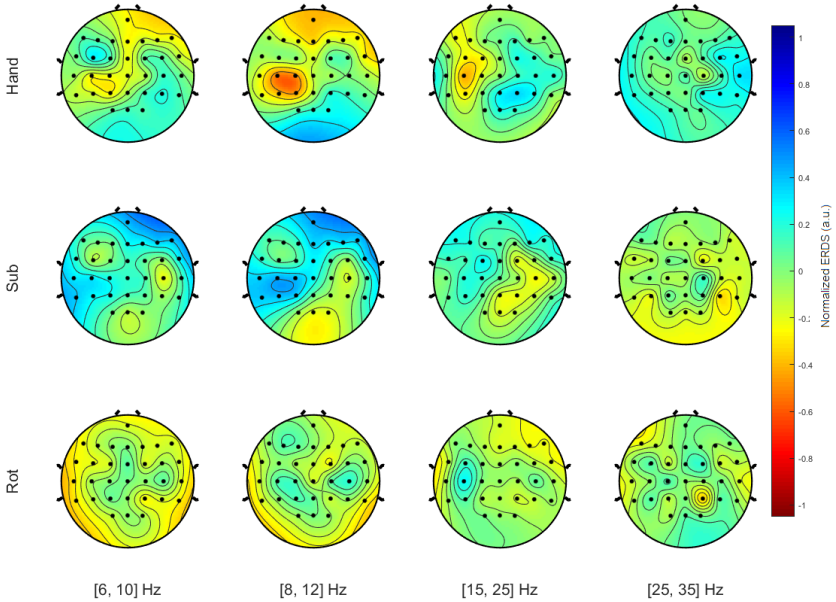


(b) Topography plot of participant B5.

Figure A.7: The plots generated from the data from participant B5.

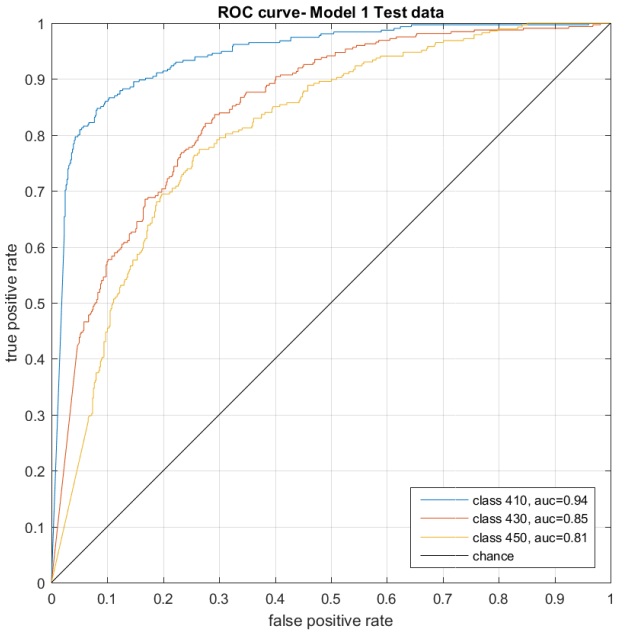


(a) ROC curve of participant B6.

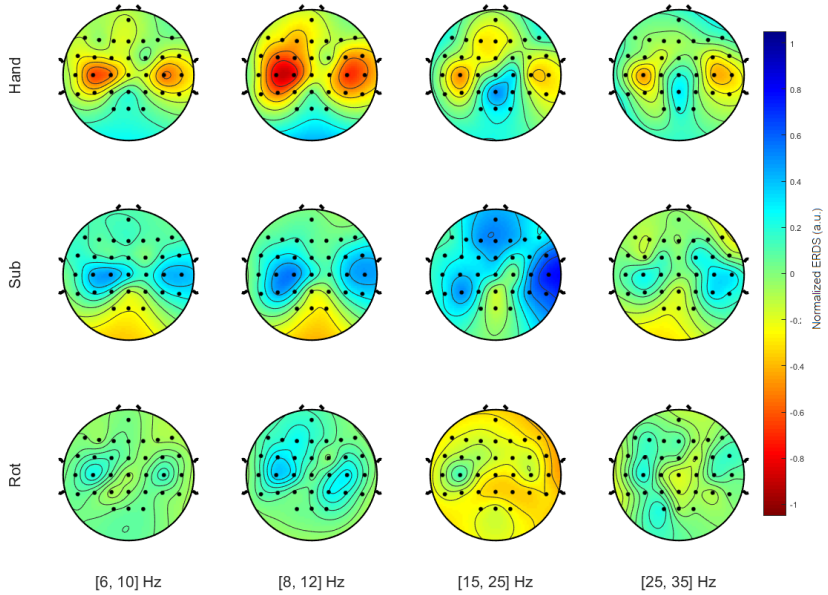


(b) Topography plot of participant B6.

Figure A.8: The plots generated from the data from participant B6.

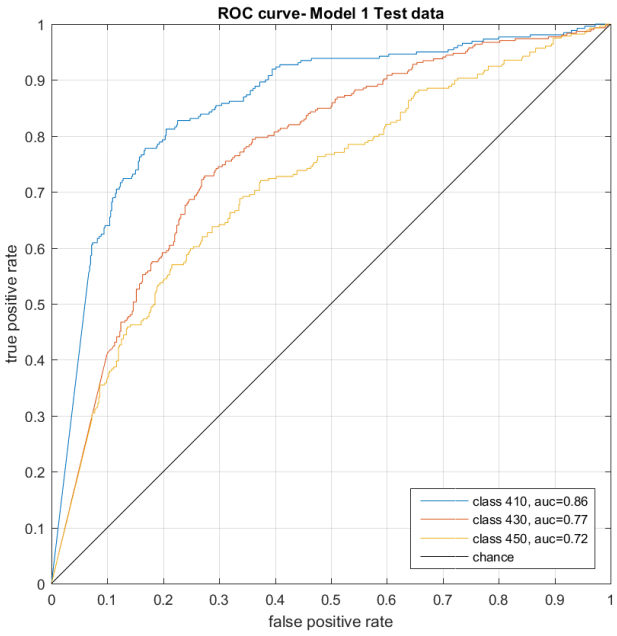


(a) ROC curve of participant B7.

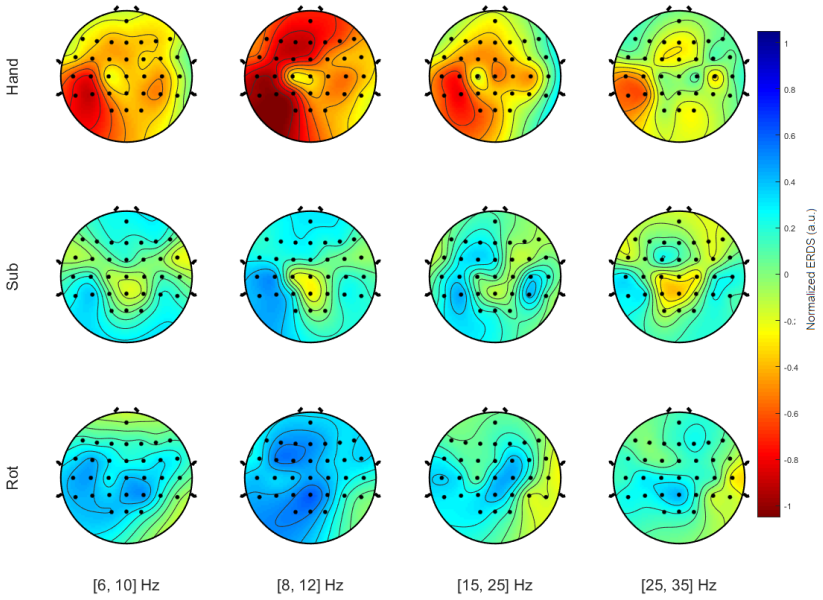


(b) Topography plot of participant B7.

Figure A.9: The plots generated from the data from participant B7.

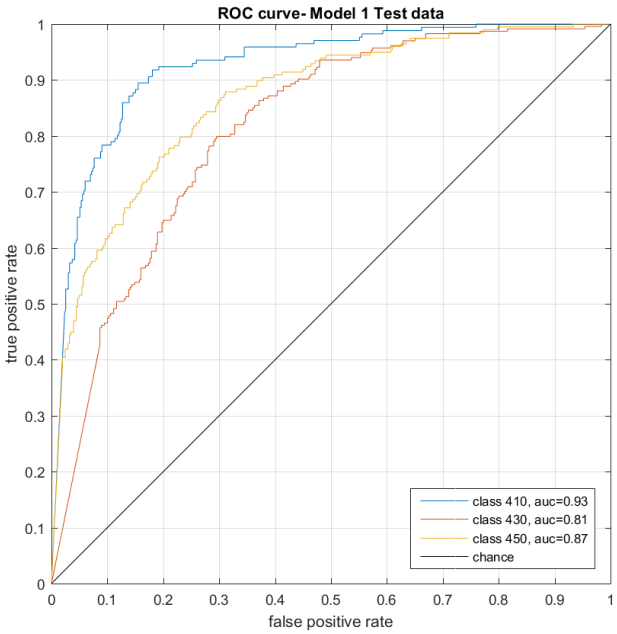


(a) ROC curve of participant B8.

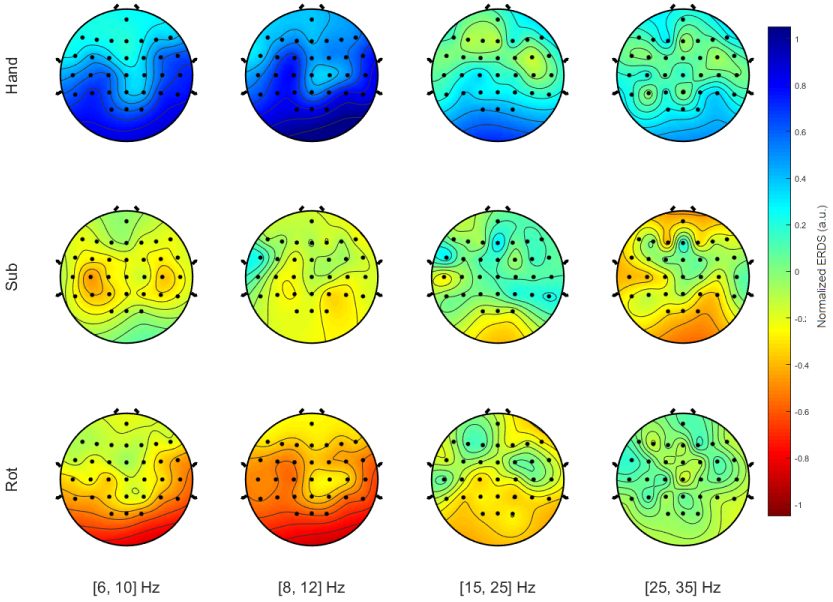


(b) Topography plot of participant B8.

Figure A.10: The plots generated from the data from participant B8.

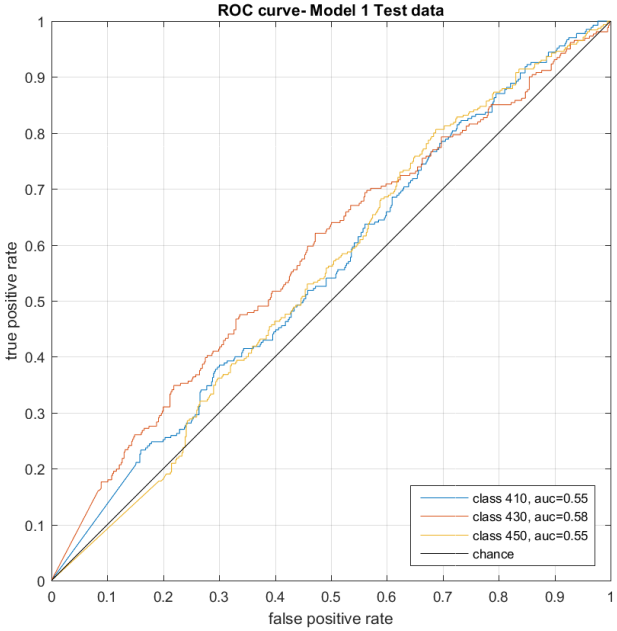


(a) ROC curve of participant B9.

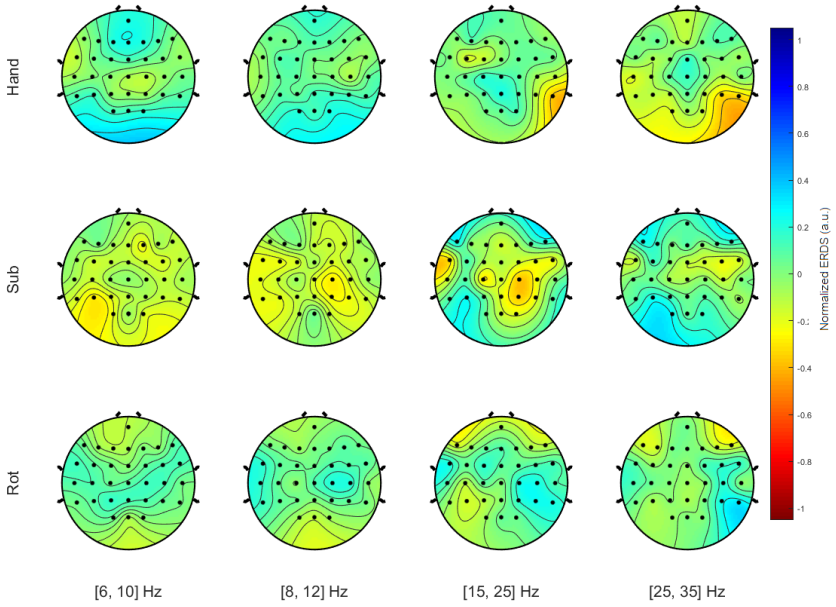


(b) Topography plot of participant B9.

Figure A.11: The plots generated from the data from participant B9.



(a) ROC curve of participant B10.



(b) Topography plot of participant B10.

Figure A.12: The plots generated from the data from participant B10.