Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Tobias Treider Moe

Towards a communication system for patients with locked-in syndrome based on EEG and visual perception

Master's thesis in Cybernetics and Robotics
Supervisor: Marta Molinas
June 2022

NTNU
Norwegian University of
Science and Technology

Tobias Treider Moe

Towards a communication system for patients with locked-in syndrome based on EEG and visual perception

Master's thesis in Cybernetics and Robotics
Supervisor: Marta Molinas
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Preface

This master thesis completes a Master of Technology at the Norwegian University of Science and Technology (NTNU) under the Department of Engineering Cybernetics in the spring of 2022. The project was proposed and supervised by Professor Marta Molinas and was completed in collaboration with Sunnaasstiftelsen. The work in this thesis is a continuation of my work in the specialization project [1] completed in December 2021.

Parts of chapter 2 and chapter 3 are updated and extended versions of what was presented in my specialization project [1]. This inclusion of the same topics was done to help the reader perceive the complete picture of my work. The software implementation [2, 3] and dataset recording were done by myself, and the experiments conducted are my original work. Dr. Luis Alfredo Moctezuma provided structural code for a recording protocol and the NSGA-II algorithm, while PhD candidate Andres Soler provided a code framework for artifact removal with ICA.

The datasets and code repositories used in this thesis can be shared upon request to me at `tobiast.moe@gmail.com`.

The main contribution of this thesis is the processing pipelines for different classification methods, the possibility of transfer learning with EEG signals, and showing the possibility of a BCI design based on visual perception. I hope this research will be of value when creating a communication system for persons with LIS.

I had no prior experience with EEG signals and BCI paradigms when I started working on these topics last fall. Hopefully, my master thesis reflects the new knowledge I have gained over the past year.

Trondheim, May 26, 2022

Tobias Treider Moe

Acknowledgement

I want to express my gratitude to Marta Molinas for letting me work on this project, introducing me to this field, and for supervision along the way. Special thanks to Anne and the rest in Sunnaasstiftelsen for showing deep interest in my research and always being ready to assist if needed. Sadly my work ended before I could get invaluable help from you.

A particular thanks to Luis Alfredo Moctezuma for regularly giving me guidance and feedback in the fall semester of 2021, and Andres Soler, who jumped in on my work during the last months of the spring semester of 2022 and gave me advice and valuable feedback on my experiments.

Particular gratitude to all who participated in the EEG recording, especially those who helped me in both datasets; without your help, the thesis would fall short. I am truly grateful for the time and enthusiasm you all set aside for me.

Finally, I want to thank my family and friends who supported and assisted me throughout my master's degree. Particularly to my mom, dad, and grandma for their never-ending support and interest in my study throughout the years.

Abstract

This thesis investigates the feasibility of a simple communication system for persons with Locked-in syndrome (LIS) by using a combination of the brain's color perception and the eye movement of the user. A person diagnosed with LIS is conscious and awake but trapped in his/her own body, unable to move and communicate. The communication system proposed here consists of a brain-computer interface (BCI) that uses recorded electroencephalography (EEG) signals generated after a dedicated visual stimulation protocol.

The BCI design needs a classification model, and this thesis explores different state-of-the-art processing and classification methods for the EEG signal. The classification task is split into two problems. The first problem consists of differentiating between a task state where the subject looks at a presented color and a resting state. The second problem consists of differentiating between the various task states, a subject looking at one of four different colors. An in-house experiment was designed and conducted to create a dataset that fits the designed BCIs specifications. The dataset includes recorded data from 22 healthy subjects, where everyone was exposed to two different protocols. The first protocol alternated between exposing the participants to one of four colors and a resting state. The second protocol displayed the color with a superimposed background icon indicative of a user-oriented need.

The results from the experiments showed that the proposed methods predicted similarly well on input data from both protocols. A random forest (RF) classifier proved to predict best on average when trained and tested on data from just one subject. The results calculated from the 22 individual RF models reached the average accuracies of 74.3 % and 61.4 % for differentiating between a task and resting state and between the four task states, respectively. RF reached these results by decomposing the input signal with variational mode decomposition (VMD), where the fractals, energies, and statistical features extracted from the modes were used.

Finally, a general model that could predict task-related information from new subjects was tested. The best performing model was a state-of-the-art convolutional neural network (CNN). The model was pre-trained on data from an optimized selection of subject data from a new dataset by the non-dominated sorting genetic algorithm II (NSGA-II). Then, the model performed a short calibration of its weights on 60 % of the data from the new subject the model was going to predict. The average accuracy for differentiating between a task and resting state and between the four task states was 69.8 % and 73.6 %, respectively. This demonstrates that a general model, only needing to calibrate on a few new samples from the user, can be used to create a BCI communication system.

Abstract - Norwegian

Denne oppgaven undersøker muligheten for et enkelt kommunikasjonssystem for personer med Locked-in syndrom (LIS) ved hjelp av en kombinasjon av personens øyebevegelse og hjernens fargeoppfatning. En person diagnostisert med LIS er bevisst og våken, men fanget i sin egen kropp og ute av stand til å bevege seg og kommunisere. Kommunikasjonssystemet som er foreslått her, består av et hjerne-datamaskin-grensesnitt (BCI) som bruker registrerte elektroencefalografi (EEG)-signaler, generert etter en bestemt visuell stimuleringsprotokoll.

BCI-designet benytter en klassifiseringsmodell, og denne oppgaven utforsker ulike
mulige state-of-the-art-prosessering og klassifiseringsmetoder for EEG-signalet. Klassifiseringsoppgaven er delt inn i to utfordringer som skal løses. Den første oppgaven består i å skille mellom når subjektet ser på en gitt farge og når subjektet hviler. Den andre består i å skille mellom de fire ulike fargene som subjektet ser på. Det ble designet og utført et eksperiment for å lage et datasett som passer til BCI-spesifikasjonene. Datasettet inkluderer EEG-signaler fra 22 friske forsøkspersoner, der alle ble utsatt for to forskjellige protokoller. Den første protokollen vekslet mellom å utsette deltakerne for en av fire farger og en hviletilstand, mens den andre protokollen viser fargen sammen med et ikon som indikerer et brukerorientert behov.

Resultatene fra eksperimentene viste at begge metodene predikerte data fra begge protokollene like godt. *Random forest* (RF) klassifisering viste seg å være best når den ble trent og testet på data fra bare ett subjekt. Resultatene beregnet fra de 22 individuelle RF-modellene nådde en gjennomsnittlig nøyaktighet på $74,3\%$, og $61,4\%$ når det gjald å skille mellom å se på en farge og hviletilstand og mellom de fire fargene respektivt. RF nådde disse resultatene ved å dekomponere inngangssignalet med *variational mode decomposition* (VMD), hvor fraktaler, energier og statistiske trekk ble hentet fra komponentene og brukt til å forutsi tilstanden.

Den siste testen i denne masteroppgaven var å lage en generell modell som kunne forutsi oppgave-relatert informasjon fra helt nye personer. Den beste modellen var et konvolusjonelt nevralt nettverk (CNN). Modellen ble forhåndstrent på data fra et optimert utvalg av subjektdata ved hjelp av *non-dominated sorting genetic algorithm II* (NSGA-II). Deretter utførte modellen en kort rekalibrering av sine filtre på 60 % av dataene til personen modellen skulle predikere på. Den gjennomsnittlige nøyaktigheten for å skille mellom å se på en farge og hviletilstand og mellom de fire aktivitetstilstandene var henholdsvis på $69,8\%$ og $73,6\%$. Dette viser at en generell modell som bare trenger å kalibrere på litt data fra brukeren, er en mulighet for det foreslåtte BCI-kommunikasjonssystemet.

Contents

List of Figures

List of Tables

List of Abbreviations

AUC	Area under curve
BCI	Brain-Computer Interface
CAR	Common Average Reference
CLIS	Complete Locked-in syndrome
CNN	Convolutional neural networks
CNS	Central nervous system
DE	Differential entropy
DL	Deep Learning
DT	Decision Tree
DWT	Discrete Wavelet Transform
EEG	Electroencephalography
ED	Euclidean distance
EMD	Empirical Mode Decomposition
GA	Genetic Algorithm
HFD	Higuchi fractal dimension
ICA	Independent Component Analysis
IMF	Intrinsic Mode Function
ITD	Intrinsic Time-scale Decomposition
LIS	Locked-in syndrome
LOOM	Leave one out model
NN	Neural network
NSGA-II	Non-dominated Sorting Genetic Algorithm II
MI	Motor Imagery
ML	Machine Learning

P1	Protocol 1
P2	Protocol 2
PFD	Petrosian fractal dimension
PRC	Proper rotation components
PSP	Parallel signal processing

QOL	Quality of life

PSR	Phase Space Reconstruction
RBF	Radial-basis function
RF	Random Forest
ROC	Receiver operating characteristic

SSVEP	Steady-state visually evoked potential
SVM	Support Vector Machine

V1	Primary Visual Cortex
V2	Visual Area 2
V4	Visual Area 4
VMD	Variational Mode Decomposition

Chapter 1

Introduction

The ability to express and communicate with other people and the environment is frequently taken for granted. People diagnosed with Locked-in syndrome (LIS) have lost this ability. They are conscious and awake but trapped within their bodies, with little or no possibility to communicate. A person diagnosed with LIS has *quadriplegia*, partial or total loss of control over limbs and torso, and *anarthria*, where the muscles used for speaking cannot be controlled [4].

Jean-Dominique Bauby was an editor in the french magazine Elle until he was diagnosed with LIS in 1995 after a massive stroke. In 1997, his book *The Diving Bell and the Butterfly* was published, which he had dictated by blinking, and eye movement [5]. He tells how it is to live with LIS, and the problem with not being able to communicate in a usual fashion. When his son asks him to play hangman, he thinks:

> *"I ache to tell him that I have enough on my plate playing quadriplegic. But my communication system disqualifies repartee: the keenest rapier grows dull and falls flat when it takes several minutes to thrust it home. By the time you strike, even you no longer understand what had seemed so witty before you started to dictate it, letter by letter. So the rule is to avoid impulsive sallies. It deprives conversation of its sparkle, all those gems you bat back and forth like a ball-and I count this forced lack of humor one of the great drawbacks of my condition."* [5]

In the classical and most widespread LIS version, vertical eye movement and blinking are intact, making it the only way to communicate with the outside world. Even though this allows for full expression, it takes time. The quote shows how locked a person with LIS is to express themselves as the time it takes to communicate excludes them from saying what they want. Furthermore, as Mathias, who has LIS, says in an interview [6], "I hate that delay. I want to answer right away because the sentence is clear in my head."

Persons with LIS also face social barriers through social exclusion, stigmatization, and frequently being underestimated. In other words, they are being 'locked out' from society and being 'locked in' in their bodies. When diagnosed with LIS, people with LIS do not always feel treated with dignity. As if muteness and paralysis mean being mentally impaired or devoid of feelings or preferences [7]. The consequence is that people rarely speak directly to a person with LIS, only around, over, or about them. There is also a generally poor understanding of the conditions, which has led to a failure to treat LIS persons as aware and conscious beings with interests and preferences worthy of consideration. This failure in treatment can be a violation of the United Nations Convention on the Rights of Persons with Disabilities [7].

It is critical and necessary to ensure that persons with LIS have a voice and may communicate as they please. The ability to communicate and do activities also directly influences their Quality of life (QOL) [3]. A way to communicate more straightforwardly with the environment is with a Brain-Computer Interface (BCI). A BCI establishes a direct communication pathway between the brain and a computer. Measuring electrical signals from the brain with Electroencephalography (EEG) and classifying them with pre-trained learning algorithms allows the user to control a computer with their brain [8]. Several paradigms are researched to control a BCI application, though many are challenging to use as they need to be learned or take significant cognitive effort to perform [9].

Both speed and comfort are essential when a person with LIS uses a BCI for communication—implying that a paradigm that can satisfy these conditions needs to be used in the BCI. This paradigm is how the brain perceives different colors when exposed to them while recording eye movement. This BCI is only feasible for persons with LIS who still have preserved control of eye movement and blinks. The persons who need a communication platform most may *not* use this design. However, it is an excellent first step to test if the paradigm works and satisfies the needs of a person with LIS.

1.1 Problem description

The overall purpose of this research is to investigate the possibility of creating a simple communication platform for a person with LIS by utilizing EEG signals. In order to do so, the designed BCI should be able to clearly differentiate between the brain's imprint of the colors looked upon, and eye movement. Furthermore, as this BCI is to be used in an online environment, the BCI design should focus on reducing model complexity, increasing classification accuracy while increasing speed, as well as focusing on comfort for the user.

1.1.1 Objectives

The main objectives for the research are:

O1: Explore different feature extractions methods and Machine Learning (ML) classifiers.

O2: Find the best method that can differentiate different between colors, and between a task- and a resting-state.

O3: Create a general model that can classify new EEG signals, both session and person, with minimal training.

1.2 Approach

The approach to the problem starts by identifying possible signal processing and learning algorithms for the designed BCI. The same BCI design and protocol frameworks were also used in [1]. A dataset was acquired from 22 volunteering participants moving their eyes to look at the presented colors on the screen. All subjects in the dataset were evaluated with several combinations of state-of-the-art feature extraction methods and classification techniques. The primarily tested methods decomposed the signal with Discrete Wavelet Transform (DWT) and Variational Mode Decomposition (VMD) and extracted fractals and energies from the component. While classification was performed by Convolutional neural networks (CNN), Support Vector Machine (SVM), and Random Forest (RF) classifiers. The best-performing methods were used to create a general model.

1.3 Limitation

The accuracies obtained during the initial experiments were too low for an online test of the BCI design in this thesis. Furthermore, problems in the dataset in [1] were uncovered, which meant that a new dataset had to be recorded. So the scope of the thesis is limited to improve the offline results by testing combinations of feature extraction and classification methods. Issues considering the hardware used for the data acquisition are outside the thesis's scope, as it was pre-determined. The BCI was only tested on data from healthy subjects, so the performance of the BCI for persons with LIS is still uncertain.

1.4 Outline

The thesis is divided into six chapters. Chapter 2 presents relevant background information about the brain, EEG signals, signal analysis, and classification methods needed for the experiments. A brief literature review on previously researched BCIs for persons with LIS and state-of-the-art EEG classification techniques is conducted in chapter 3. In chapter 4, the methods used for data acquisition and processing pipelines are explained. Chapter 5 presents the performed experiments and their corresponding results. In the last chapter, chapter 6, the results and BCI design are discussed, followed by a conclusion and suggestions for further work.

Chapter 2

Background

This chapter [1] *aims to provide the theoretical basis necessary to create models to be used in the BCI for communication. First, by explaining LIS, the human brain, and the generation of EEG signals. Then, introducing the data processing methods used to extract the most relevant features are introduced. Lastly, a description of classification models used in the BCI and an optimization technique are presented.*

2.1 Locked-in Syndrome

LIS is a rare neurological disorder resulting from a brainstem lesion, more precisely, an insult to the ventral pons. The disorder can have resulted from stroke, injury, hemorrhage, trauma, tumor, or ischemia [10, 11]. The outcome is nearly total paralysis of voluntary muscle control while cognitive functions are still operative. In addition, the following senses, vision, audition, gustation, and olfaction, are intact, as well as the experience of heat, cold, pain, and pleasure [7]. However, their eyesight can be obscured by diplopia, or blurry vision [4]. There is currently no cure available for persons diagnosed with LIS, only supportive and communicative care [12].

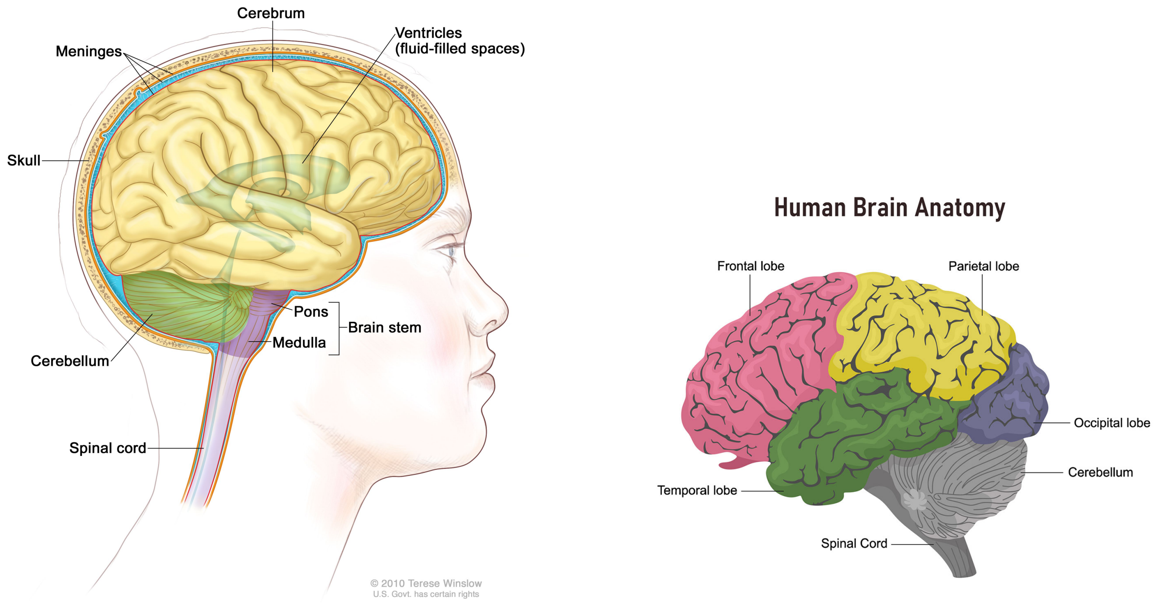
LIS are typically categorized into three subtypes [13]: classical, partial, and complete locked-in syndrome. A person with classical locked-in syndrome, called LIS in this work, has no mobility other than blinking or vertical eye movement. Partial locked-in syndrome in some literature called *incomplete* LIS is classified as persons who have preserved some voluntary motor functions other than eye movement. The most severe type is Complete Locked-in syndrome (CLIS). A person with this condition cannot voluntarily move muscles at all, including blinking and horizontal eye movement, making them unable to communicate [4]. A person with LIS may, over time, completely lose their voluntary control ending in the state of CLIS.

2.2 The human brain

The human brain is part of the Central nervous system (CNS) together with the spinal cord and consists of three parts: the cerebrum, brain stem, and cerebellum, where the pons are placed in the upper part of the brain stem, visualized in fig. 2.1a. The cerebrum divides into four lobes, where each lobe

[1] Note that this is an updated and extended version of the background in the author´s previous work [1].

controls different functions. The signals of interest in this project are mainly located in the occipital lobe, see fig. 2.1b, the part of the brain that is involved with vision.



(a) Anatomy of the brain. Reprinted from [14].

(b) The four brain lobes. Reprinted from [15].

Figure 2.1: Visualization of the anatomy of the brain and the four brain lobes.

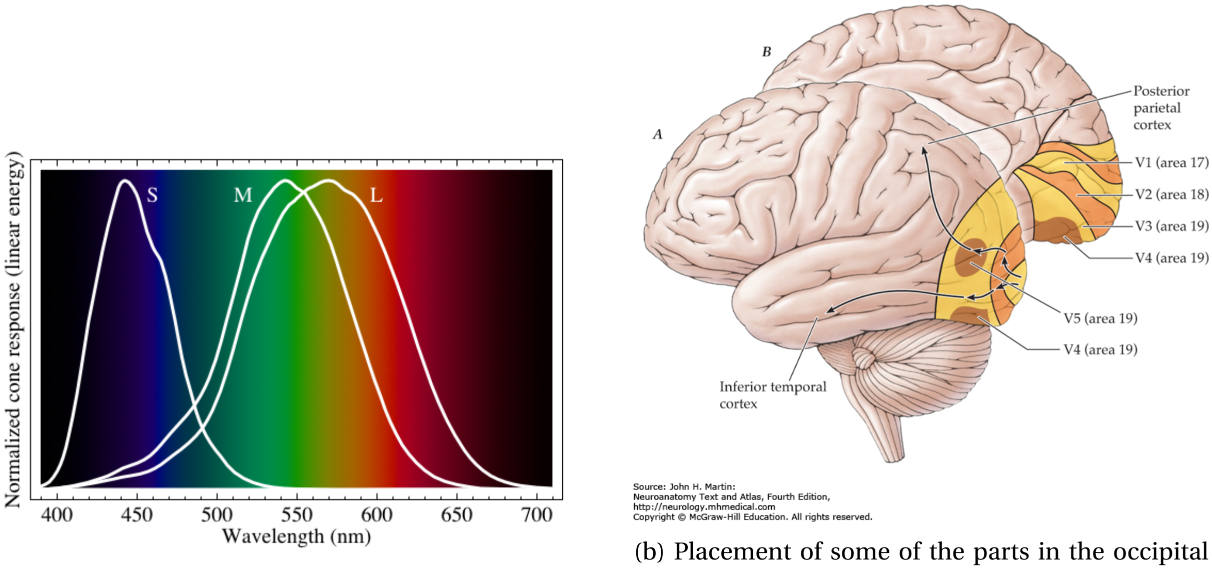
2.2.1 Brain activity

The CNS consists of neurons that send chemical and electrical signals between each other. A neuron consists of a nucleus, axons that transmit electrical impulses, and dendrites, which receive electrical impulses from other neurons. In the brain, each neuron is connected to around 10 000 other neurons [16]. A signal is sent from a neuron when its dendrites' potentials sum above a threshold, around -55 mV (millivolts). The potential is a depolarisation from the resting potential of around -70 mV. When the potential exceeds the threshold, the neuron will get an action potential and fire an electrical signal along its axon, which connects to a dendrite from another neuron. This connection is called a synapse. After a signal is sent, the neuron needs 2 milliseconds (ms) before it can send a new signal. All neurons will always send their signal with the same strength, though the frequency can vary [17].

2.2.2 Color vision

The color of an object is the light reflected by the main sensory organ of the visual system, the eyes. The perception of color is decided by variations in wavelength, purity, and intensity. Humans' spectral sensitivity ranges from about 400-700 nanometers (nm), the visible light range for humans. The purity is the bandwidth and distribution of the wavelength variations in a given light, and a more pure light has a richer color. The intensity is the number of photons emitted by light and is perceived as darker and brighter colors [18]. There are two photoreceptors, cones, and rods, in the eye, which are located in the retina. Humans have three different cones, all with equal sensitivity distribution but different peak wavelengths in their photopigments. The peaks are called short(S), medium(M)-, and long(L)-wavelengths and peaks generally at 420, 534, and 564 nm, respectively, as seen in fig. 2.2a. The rods are light-sensitive and support the vision when it is less light [19].

The light sensed by the retina, in the eye, is directly transferred to the Primary Visual Cortex (V1) in the occipital lobe with connecting neurons. From here, the signal is sent to neighboring regions, Visual Area 2 (V2) and Visual Area 4 (V4), see fig. 2.2b. Specific subregions in V2 process the signal before it is projected to V4 as well [18]. Implying that color information is *mainly* being received and processed in V4 [20]. Though, it is important to note that other regions in the occipital lobe also are important for the perception of colors [19].



Source: John H. Martin:
Neuroanatomy Text and Atlas, Fourth Edition,
http://neurology.mhmedical.com
Copyright © McGraw-Hill Education. All rights reserved.

(a) Normalized response spectra for human cones. Reprinted from [18].

(b) Placement of some of the parts in the occipital lobe, the area in parenthesis is the Brodmann area [21]. Reprinted from [22].

Figure 2.2: Visualization of the wavelength to human cones and specialized areas in the occipital lobe.

2.2.3 Eye movement

Anatomically, the eyeball is positioned in orbita, controlled by muscles in three antagonistic pairs, where all contribute to horizontal, vertical, and rotary motions. The muscles are designed with small fibers because of the need for precise contractions and are attached to the wall in orbita [19]. Motor signals to control the velocity and the position of the eyeball originate from the brain stem, which has received input from the retina and V1 [23]. A contraction from a muscle discharge electrical activity can cause a significant change in the electrical fields around the head. So, the muscle contraction is readable with EEG. However, they can distort and overwhelm the EEG signals, as this signal is stronger than signals created by neurons [24].

2.3 Electroencephalography

EEG is a method that measures the currents that flow during synaptic excitations between neurons in the brain. These currents create a magnetic field and an electrical field, where the latter can be measured with an EEG recording system. The electrical field is produced with a dipole in caudal orientation and a negative pole in the cranial direction. The dipoles are more uninterrupted and spatially synchronized than the action potentials and are therefore readable with EEG. Since a single synaptic excitation generates a low electrical current, thousands of neurons need to activate to get a readable signal. In addition, these neurons need to be placed perpendicular to the brain's surface for the signal to be strong enough [25, 16, 26]. The activity is measured with either intracranial electrodes (invasive method) or on the scalp's surface (non-invasive method). Electrodes will only detect a portion of the signals in the brain, and the signals will be a sum of many postsynaptic potentials in the close vicinity of the electrode [27]. The signal from an electrode will be denoted as a *channel*.

A problem with reading non-invasive EEG signals is that the measured signal can be overwhelmed by other electrical interference from the body (e.g., muscle contraction) or weakened by traversing from its origin to the electrode. The human head consists of several different layers with different electrical resistance, which an electrical signal must traverse before being detected and read by an electrode. The different layers also make the EEG signal to be generally a nonlinear sum of brain sources [27]. EEG signal patterns are also prone to change when using drugs for different treatments and suppression of mental and CNS abnormalities, as well as watching television and listening to music [16].

Recording EEG signals is overall very safe, both for the patient and the technologist. Still, the subject is exposed to electrical current, and the most critical injuries are skin burns, induction of seizures, or ventricular fibrillation. The sources of harmful current can come from improper grounding, leakage, or double-grounding [27].

2.3.1 EEG frequency bands

The frequencies in a EEG signals are commonly separated into five different frequency bands that correspond to the subject's condition. However, there are three problems with the separation. Firstly, literature disagrees on which band is associated with what frequencies [25]. Secondly, the association with a frequency band does not imply that this is the only brain process. Lastly, the decomposition assumes that the EEG signal is exclusively a mixture of sinusoids, though an EEG signal consists of non-sinusoidal signals and arrhythmic activity [28]. This thesis will use the following frequencies and associations for the five frequency bands presented in table 2.1.

Table 2.1: EEG frequency bands and their associations, values and association are taken from [25].

Frequency band	**Frequency range** [Hz]	**Associated with**
Delta	0.5 - 4	Homeostatic sleep drive
Theta	4 - 8	Associated with homeostatic sleep drive
Alpha	8 - 12	Relaxed wakefulness and drowsiness
Beta	13 - 30	Active thinking
Gamma	> 30	Cognitive states

2.3.2 Artifacts

Artifacts are undesired electrical signals detected by the electrodes but do not originate from the cerebrum. They may introduce changes in the measurement and affect the wanted signal. There are two different categories of artifacts: *physiological* and *non-physiological*. Physiological artifacts come from a variety of bodily activities. The most common and studied are ocular artifacts, for example, eye movements and blinks, and artifacts from the muscles. Muscle artifacts can, for example, be electrical activities caused by muscle contraction or cardiac activity from the heart. Other physiological artifacts are tongue movement, speaking, chewing, sweat, and respiration [27, 29]. Sweat can alter the impedance of the electrodes, while respiration can introduce a rhythmic activity in the signal. Non-physiological artifacts can occur from a poor connection between electrodes and the user, power line noise (50 Hz in Europe), or unique environmental, electrical noise produced by, e.g., ventilators and feeding/infusion pumps [30]. These artifacts might not be applicable in an EEG laboratory but may be outside such a controlled environment.

2.3.3 EEG signal acquisition and recording protocols

EEG uses the principle of differential amplification, recording the voltage difference between a recording electrode and a reference electrode. Where the reference electrode can be placed in two different ways, *monopolar* and *bipolar*. The data acquisition in this thesis is recorded with monopolar placement. Monopolar places the reference electrode away from the area of interest. Distancing the reference electrodes from the other electrodes maximize the rejections of the common voltages in the electrode, and the reference [25].

When creating a BCI, there are mainly two different types of signals of interest, the *task*- and *rest* signals. Note that there can be several task signals. A protocol is needed to generate and record the task signals. It consists of cues given to participants to follow in a time window while recording their brain activity. Between the tasks are, the participant instructed to rest, creating a resting state. It is equally important to understand when a user is performing a task or not, as differentiate the different tasks.

However, the brain is never entirely at rest, so a stable resting state does not necessarily exist. There are, all the time, spontaneous changes in regional neural firings in the brain from continuous metabolizing of oxygen and glucose energy from the blood in the cerebrum that fuels general maintenance and ongoing neural communication [31]. The spontaneous activations can change the local blood flow, which can give signal oscillation [32]. Implying that a resting state refers to when there is no goal-directed neuronal action with the integration of the external environment and when the subject is not actively engaged in sensory or cognitive processing [25].

2.3.4 Electrode placement

The placement of the electrodes can be decided from several layouts. This thesis will use the most acknowledged layout, the 10-20 international system, see fig. 2.3. This system represents standard electrode placement intervals of 10 or 20 percent between the electrodes. Each electrode has a letter and a number. The letter represents the lobe, while the numbers represent which side of the brain. Odd numbers represent electrodes on the left side and even numbers on the right side. A higher number is further away from the midline, while small numbers are close. Electrodes at the center are marked with the letter *z*, instead of a number.

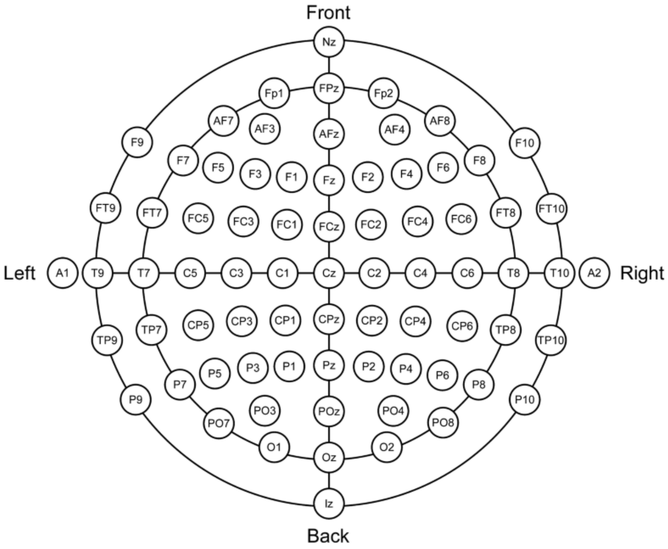


Figure 2.3: The international 10-20 system for electrode placement. Reprinted from [33].

2.4 Brain-computer interface

A BCI establishes a direct communication pathway between the brain and a computer, creating a digital environment controlled by information from the brain. It provides a way to communicate without relying on muscle contractions and has primarily been developed to help people with severe disabilities [34].

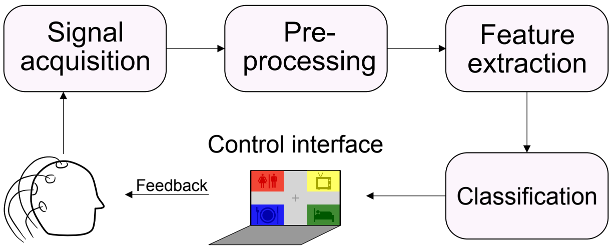


Figure 2.4: The five stages in a BCI.

A BCI has five stages [35], visualized in fig. 2.4: signal acquisition, pre-processing (enhance signal-to-noise ratio), feature extraction, classification, and control interface. There are two different types of brain signal generation used in a BCI, active and passive. Passive signals monitor the users' cognitive state, for example, drowsiness and emotional states, while active signals are generated from mental activities or reactions to real-world stimuli [9]. There are two different types of BCIs, synchronous and asynchronous. The main difference is that synchronous BCIs can only receive commands at predefined time intervals, while asynchronous are continuously ready for signals. Asynchronous BCIs offer a more general use of the BCI, though it may be problematic as it needs a stable resting state to work properly [36].

2.5 Pre-processing

The raw EEG signals are prone to be affected by artifacts, as mentioned in section 2.3.2. Therefore it is necessary to enhance the signal-to-noise ratio before any analysis of the EEG signals. The methods used in this thesis are described below.

Bandpass and Notch filter

An EEG signal can be influenced by noise from the environment, such as a common-mode signal. Such noise can come from the power line and are around 50 Hz here in Europe. Removing this noise can be done with a band-stop filter, a Notch filter, which stops a range of frequencies from passing based on two cut-off frequencies, or rather just one frequency. This thesis defines a *filter* as a mathematical procedure that lets some frequencies pass, decided by its cut-off frequencies. Another way to reduce the noise is to filter the signal with a bandpass filter. A bandpass filter has two cut-off frequencies, an upper and a lower limit, which the wanted frequencies from the EEG signal can pass through.

Common Average Reference

There is a need to identify small signal sources in noisy EEG recording, which can be amplified by having an independent reference electrode and electrodes that will not get signals from the same areas in the cerebrum [37]. However, isolating the reference electrode and the other electrodes from each other is near impossible; therefore, a synthetic reference can help with enhancing the signal-to-noise ratio and reducing the noise [25]. Common Average Reference (CAR) is such a method. It removes simultaneously-recorded common information from all electrode channels, V_i^{CAR}, where i is the number of a channel. CAR is computed with the formula:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{n}\sum_{j=1}^{n} V_j^{ER} \tag{2.1}$$

Where V^{ER} is the potential between the i^{th} electrode and the reference, and n is the number of electrodes.

Independent component analysis

EEG is a summation of electrical brain activity, though it can be contaminated with artifacts, such as muscle contractions. Independent Component Analysis (ICA) is a method that calculates independent components and separates the information from a mixed-signal. In other words, it can separate artifacts from an EEG signal. ICA has two assumptions; firstly, the independent components are statistically independent. Secondly, the independent components are non-Gaussian. In addition, does the number of outputs components equal the number of channels in the input. A problem with using ICA is that the components have to be examined and removed manually. This makes ICA a tedious process to perform on large datasets.

2.6 Data analysis

Data analysis can extract information hidden in the raw signal, as a raw EEG signal often have small amplitudes and are very noisy, even after pre-processing. The process consists of manipulating and transforming the data from one format or domain to another [25]. Choosing the correct data analysis method is crucial for extracting the best information, though there are no general best methods. In addition, the performance of the methods is dependent on the EEG signal characteristics and aim of the experiment.

2.6.1 Discrete Wavelet Transform

DWT uses a wavelet transform to decompose a signal into several frequency bands. A wavelet transform is a Fourier transform, but a wavelet is used instead of the sine and cosine functions. A wavelet also called a mother wavelet, is a finite oscillation with an amplitude that begins at zero, increases, and decreases before it ends back to zero. There exist several different wavelets and wavelets families, which all transform the signal differently. Other than the mother wavelet, two important parameters are shifting and scaling. The two parameters helps capture different frequency regions over the whole signal [25]. The DWT is presented below in eq. (2.2).

$$\mathrm{DWT}_{j,k} = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{|2^j|}} \psi \frac{t - 2^j k}{2^j} dt \tag{2.2}$$

Where j and k are the scaling and shifting parameters, respectively, ψ is the mother wavelet, and x is the signal to be transformed. The last predefined parameter is the decomposition level. Equation (2.2) returns two outputs, high- and low-frequency parts, called detail coefficients (Dn) and approximation coefficients (An), where $\{n \in \mathbb{Z}^+\}$ symbolize the current level. An is put back into the function until a predefined level is reached. So, a decomposition of signal $x(t)$ with i as the decomposition level has the structure $[A_i, D_i, D_{i-1}, ..., D_1]$.

2.6.2 Variational Mode Decomposition

VMD [38] decompose a signal into different modes, much like the Empirical Mode Decomposition (EMD) [39]. It reproduces the input, while still having specific sparsity properties. The main differences between these two methods are that VMD is calculating concurrently, allowing for backward error correction instead of a recursive calculation, which makes it adaptive and less sensitive to noise. Centrally for the method is the definition of a *mode*, called Intrinsic Mode Function (IMF). VMD uses a new definition; a mode $u_k(t)$ can be considered to be a pure harmonic signal with amplitude, $A_k(t) \geq 0$, and an instantaneous frequency that varies much slower than the phase. A mode satisfied by this new definition also satisfies the EMD mode properties, though the contrary is not necessarily true. So by letting the input signal f be decomposed into k IMFs components, then VMD will find the best solution to the constrained variational problems in eq. (2.3).

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \delta_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \quad \text{s.t.} \quad \sum_k u_k = f \tag{2.3}$$

where ω_k is the center frequencies to the mode u_k, δ is the Dirac distribution, and $*$ denotes convolution. Both ω_k and u_k are updated during the optimization. The authors in [38] use an augmented Lagrangian multiplier to enforce constraints on the problem strictly, as this is an unconstrained optimization problem. The augmented method combines a quadratic penalty and Lagrangian multipliers, where the quadratic penalty helps with reconstruction when there is noise present. In contrast, the Lagrangian multipliers help with enforcing the constraints strictly. The optimization will continue until a convergence of the modes is met, less than an error variable.

There are two main problems with VMD. Firstly, the algorithm does not guarantee a converging to the global minimum. The second problem is the need to predefine the number of modes decomposed of the original data. Defining too few modes implies that some components are constrained in other modes or discarded as noise. This problem is visible as structured noise in the modes. While defining too many nodes will either capture the noise or mode duplication.

2.6.3 Intrinsic Time-scale Decomposition

Intrinsic Time-scale Decomposition (ITD) [40] is a method for time-frequency energy analysis. It is used in application to non-stationary signals obtained from complex systems with underlying changing dynamics on several time scales simultaneously. Central in the method is the decomposition of the base signal into Proper rotation components (PRC), which defines the monotonic signal trend, as well as the instantaneous frequency and amplitude. Given a signal X_t and an operator $\mathcal{L}$, the decomposition of X_t is

$$X_t = \mathcal{L}X_t + (1-\mathcal{L})X_t = L_t + H_t, \tag{2.4}$$

where L_t is the baseline, constructed as a linearly transformed contraction of X_t, and H_t is the PRC. From X_t can the local extrema be found, denoted $\{\tau_k, k \in \mathbb{Z}^+\}$. If there is an extrema with several points, the right endpoint should be chosen. In addition, defining $\tau_0 = 0$ and the last value, denoted as τ_{end}, as an extrema. Simplifying the notation by defining $X(\tau_k) = X_k$ and $L(\tau_k) = L_k$, then the linear transformation, on the interval $(\tau_k, \tau_{k+1}]$ becomes

$$\mathcal{L}X_t = L_t = L_k + \Big(\frac{L_{k+1} - L_k}{X_{k+1-X_k}}\Big)(X_t - X_k), \quad t \in (\tau_k, \tau_{k+1}], \tag{2.5}$$

where

$$L_{k+1} = \alpha\Big[X_k + \Big(\frac{\tau_{k+1} - \tau_k}{\tau_{k+2} - \tau_k}\Big)(X_{k+2} - X_k)\Big] + (1-\alpha)X_{k+1}, \tag{2.6}$$

α is defined in the interval $(0,1)$, typically 0.5. The initial baseline are defined to $L_0 = (X_{\tau_0} + X_{\tau_1})/2$. While the final baseline is calculated as $L_{end-1} = (X_{end+1} + X_{end})/2$. The PRC can be extracted from the signal and baseline, by rewriting eq. (2.4) as

$$H_t = X_t - L_t \tag{2.7}$$

New PRCs can be found when using L_t as the new signal in the decomposition. This process is stopped when L_t is monotone, or its amplitude is below a threshold.

2.6.4 Phase Space Reconstruction

A phase space can characterize dynamic systems, where a point in this space describes a state at any given time. There are several methods for a Phase Space Reconstruction (PSR) [41], where the most used method uses delays [42]. PSR are calculated by letting $X = \{x_1, x_2, ..., x_N\}$ be a 1-dimensional time series, where N is the total number of its data points. Defining a data point in the time series to be noted as $X(i) = x_i$ for $i = 1, 2, ..., N$. Then X can be reconstructed in a multi-dimensional phase space Y by

$$Y_m = (x_j, x_{j+1}, x_{j+2}, ..., x_{j+N-(M-1)\tau}), \quad j = \tau m \tag{2.8}$$

$$Y = [Y_1, Y_2, ..., Y_M]^T \tag{2.9}$$

where $m \in [0, M-1]$ is the embedding dimension of the vector Y_m, M is the total number of dimensions, and τ is the delay time. The PSR, Y, will have the shape of $[M, N-(M-1)\tau]$. The dimensionality of the phase space can be reduced by calculating Euclidean distance (ED) of the points [43]. The ED between a point $k = 1, 2, ..., N-(M-1)\tau$ in the phase space and the origin can be calculated as

$$ED(k) = \sqrt{Y_1(k)^2 + Y_2(k)^2 + ... + Y_M(k)^2} \tag{2.10}$$

So the final dimensionality of the PSR is a vector of length $N-(M-1)\tau$.

2.7 Data features

A feature is an individual measurable property or characteristic of what is being observed. Features can be extracted directly from raw EEG signals or from transformed and decomposed data explained in section 2.6. The goal is to choose a subset of informative, discriminating, and independent features that can efficiently describe the input data while reducing the effect of noise and irrelevant variables [25, 44]. Removing irrelevant features is crucial for effective training and good generalization of, for example, classification algorithms.

2.7.1 Energy distribution

The energy of a discrete signal is defined as the area below the squared magnitude of the signal [25]. Let $w_j(r)$ denote the coefficient of one of the sub-bands with length N, from a decomposition in section 2.6, at position r. The **instantaneous energy** reflects the amplitude of the signal, and can be computed in eq. (2.11). While the **Teager energy** reflects the variations in both frequency and amplitude of the signal [45, 46], and can be computed in eq. (2.12).

$$f_{instantaneous} = log_{10}\Big(\frac{1}{N_j}\sum_{r=1}^{N_j}(w_j(r))^2\Big) \tag{2.11}$$

$$f_{Teager} = log_{10}\Big(\frac{1}{N_j}\sum_{r=1}^{N_j-1}\Big|(w_j(r))^2 - w_j(r-1) * w_j(r+1)\Big|\Big) \tag{2.12}$$

2.7.2 Fractal dimension

A fractal dimension is a statistical index of complexity describing how a time series changes with the scale at which it is measured [25]. There exist several fractal dimensions, where Petrosian fractal dimension (PFD) and Higuchi fractal dimension (HFD) have been used in this thesis.

The **Petrosian fractal dimension** provides a fast computation of the fractal dimension by translating the series into a binary sequence. It can be calculated in eq. (2.13), where n is the length of the sequence and N_∇ is the number of sign changes in the binary sequence [47].

$$f_{Petrosian} = \frac{log_{10}n}{log_{10}n + log_{10}\left(\frac{n}{n+0.4N_\nabla}\right)} \tag{2.13}$$

The **Higuchi fractal dimension** estimates the dimension of a time series in the time domain by approximating the mean length of the curve with segments of k samples [48]. Consider a finite set, length N, of time series observations taken at a regular interval $X(1), X(2), ..., X(N)$, which is used to create the following new time series X_k^m

$$X_k^m : X(m), X(M+k), X(m+2k), ..., X\Big(m + \Big(\frac{N-m}{k}\Big)k\Big), \quad m = 1, 2, ..., k \tag{2.14}$$

Where m is the initial time and k is the interval time. The length of the curve X_k^m can be defined as

$$L_m(k) = \frac{1}{k}\left(\sum_{i=1}^{\frac{N-m}{k}} \Big(X(m+ik) - X(m+(i-1)k)\Big)\right)\left(\frac{N-1}{\left(\frac{N-m}{k}\right)k}\right) \tag{2.15}$$

Defining k_{max} to be an integer greater than 1, Higuchi takes the mean length of the curve for each $k = 1, 2, ..., k_{max}$, calculated in eq. (2.16). Where the HFD is the least square slope of eq. (2.16), calculated in eq. (2.17).

$$L(k) = \frac{1}{k}\sum_{m=1}^{k}(L_m(k)) \tag{2.16}$$

$$f_{Higuchi} = \frac{ln(L(k))}{ln\left(\frac{1}{k}\right)} \tag{2.17}$$

2.7.3 Differential entropy

Entropy is a measurement of the uncertainty of a random variable, defined as $-\int_X f(x)log(f(x))dx$ in [49] for a random variable X. Differential entropy (DE) is used when the random variable is continuous and measures the complexity. Let now X be a time series following the Gauss distribution $N(\mu, \sigma^2)$, then its DE can be calculated as

$$\begin{aligned} h(X) &= -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^3}} log_e\Big(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^3}}\Big)dx \\ &= \frac{1}{2}log_e(2\pi e\sigma^2) \end{aligned} \tag{2.18}$$

Where σ^2 is the signal variance, and π and e is constants. EEG signals do not follow a fixed distribution, though it is shown that the Gaussian distribution is almost present in the signal after band-pass filtering [50].

2.7.4 Statistical features

Statistical features are explicitly present in the input data and describe its characteristics. The statistical features used in this thesis are maximum, minimum, mean, median, variance, standard deviation, Kurtosis, and skew. The variance and the standard deviation measure the spread around the mean. While Kurtosis measures the outliers, and skew the asymmetry of the distribution in the input.

2.8 Multi-class classification with Machine Learning

The EEG signals need a classification algorithm to be used in a BCI, and in this thesis, it is done with a ML algorithm. A ML algorithm uses experience to improve performance and predict a class when given data. There are three main types of learning, unsupervised-, reinforcement- and supervised learning. This thesis uses supervised learning when each input sample in a dataset has a corresponding output label. The algorithm learns an algorithm that best fits the label when given a corresponding sample [51]. ML often falls short when there is a need to extract high-level and abstract features from the data. A solution is to use either feature extraction techniques on the data or a Deep Learning (DL) algorithm. DL algorithms allow building complex concepts from simpler concepts and other ML algorithms. This method decreases the need for manual feature extraction while acquiring state-of-the-art performance [52].

2.8.1 Support Vector Machine

A SVM classifier is a supervised ML method [51], which constructs linear hyperplanes optimized for class separation. The placement of the hyperplane, called the maximum margin *separator*, maximizes the distance between the points from the different classes, which means that it minimizes its generalization loss. The main property of SVM is its *support vectors*, which are weights associated with the points closest to the separator. The points closest to the separator are most important, and SVM can remove other points to decrease memory usage and calculation cost. Not all data can be separated linearly, so SVM utilizes the *kernel trick* to map the input data in higher dimensions, where it searches for the optimal linear separation.

2.8.2 Random Forest

RF is an *ensemble* ML method, it selects an ensemble of hypothesis from the hypothesis space and aggregate their results [51]. The method utilizes the "bootstrap aggregating", short *bagging*, technique, which creates k random subsets of the input data that are used to train k independent Decision Tree (DT) classifiers. The DTs in a RF split their nodes by using the best feature from a random subset of the original features [53]. In other words, the DTs are built with different selection of input data and features from each other, which minimize the chance of equal trees. Finally, all predictions from the DTs are aggregated, implying the majority vote for a class is the final predictions for the input [54].

2.8.3 Convolutional Neural Networks

CNN is a DL method that is specialized in processing data that has a known grid-like topology and learn by back-propagation. A neural network mimics the human brain's neurons and how they work together. Special for a CNN is that its neurons have three dimensions, the spatial dimensionality (height and width) and depth [55]. In addition, that it utilizes the *convolution operation*. An example of a CNN architecture are visualized in fig. 2.5.

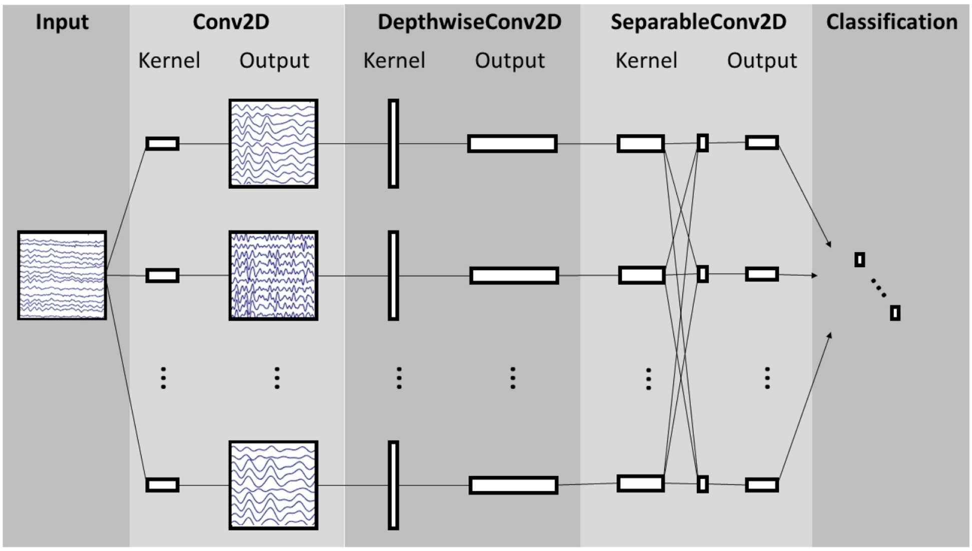


Figure 2.5: Visualization of a CNN architecture, the EEGNet, used for EEG signal classification. Reprinted from [56].

CNNs is built up with several *layers*, always an input and output layer, and $n \in \mathbb{N}$ *convolution layers* called *hidden layers*. In addition there is typically a *pooling* layers as well. A hidden layer consists of several trainable filters that extract features from the input, implying a reduced need to feature extract the input data. Only the hidden layers and functions used in this project are explained. Central in a CNN is the convolution operation, which is a linear operation on two functions of a real-valued argument. The first function is the input and the second is a learnable kernel, which has smaller spatial dimensionality than the input. The operation calculates the scalar product between the input and kernel several times until all the products are found in the spatial input space [55].

The motivation to use a CNN is its sparse interactions between neurons, accomplished by making the kernel smaller than the input, implying a decrease in the computation time. A CNN also shares its parameters by using them for more than one function in the model. Parameter sharing reduces the storage requirements of the model without affecting the forward propagation. The final motivation for a CNN is that it has the property that if the input changes, the output changes in the same way [52].

A *pooling layer* replaces the output at a specific location with a summary statistic of the nearby outputs, reducing the number of outputs. Such a statistic can be the average of the *n* neighboring inputs [52]. A *depthwise convolution* layer uses a separate kernel for each input channel when convolving the input. While a *separable depthwise convolution* layer first performs a depthwise convolution, then a pointwise convolution. In other words, a pointwise convolution applies a 1 × 1 convolution, effectively combining the output from the depthwise convolution. The pointwise convolution mixes the channels and reduces the dimensionality of the input [57]. At the end of each convolutional layer and the final output layer is there an *activation function* which defines how the weighted sum of the inputs is transformed to an output, the probability prediction.

A CNN trains over several epochs. An epoch is when the network has trained on all samples in the dataset. Repeatedly tuning the weights with the same data can result in overfitting. Overfitting is when the model has created a perfect fit to the training data but has not found a generalized fit that can work on similar unseen data. Two methods to combat overfitting are *dropout* and *early stopping*. A dropout layer randomly drops connections between neurons and layers during training. Early stopping ends the training before intended, when it meets a criterion. A problem with early stopping is that it might end the training too early, resulting in an underfitted model.

2.9 Optimization with Non-dominated Sorting Genetic Algorithm II

Optimization techniques can help create the most optimal model in the shortest amount of time. The most optimal model has the perfect hyperparameters and only uses the relevant features for a prediction.

A Genetic Algorithm (GA) solves multi-objective optimization problems by mimicking biological evolution. It defines a population to be a set of genes, where each gene is a solution to the problem and has chromosomes that can be altered and mutated for the next generation [58]. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [59] is such an algorithm and is based on four main principles. The first is *non-dominated sorting*, where each gene is getting a rank based on how many other genes this gene is dominated by, based on the Pareto dominance. All genes are then sorted by their rank. The next is *elite-preserving operator*, the best genes are transferred to the next generations without being mutated until dominated. Thirdly is the calculation of *crowding distance*, which is a value based on the density of other solutions surrounding a particular solution. The last principle is the *selection operator* for the next generation. The NSGA-II selections are based on the domination rank and the crowding distance, where a lower rank and higher crowding distance are wanted for the next generation [60].

NSGA-II works by first creating an initial population P_t of size N. A new population Q_t is created by mutating P_t, and the two populations are combined to R_t of size $2N$. Their non-dominated level ranks the population in R_t, and a new population P_{t+1}, size N, is taken out by the selection operator. The same procedure continues until stopping criteria are met [59]. A problem with NSGA-II is that it has a poor initial population. The algorithm can then be stuck in local optima, even though this is combated by favoring edge cases, genes with high crowding distances.

Chapter 3

Literature review

This chapter[1] *presents the literature on current communication methods available for LIS patients. Possible BCI paradigms and already tested BCIs created with the thought of persons with either LIS or CLIS in mind. The final section introduces state-of-the-art EEG classification methods.*

3.1 Communication practices for LIS patients

For a person with LIS, the social barriers from not being able to communicate are often more harmful than their physical conditions [7]. The most common communication technique for them is an analog alphabet board. The person with LIS signals an assistant, pointing at letters, by blinking or making eye movements [7]. Then the assistant conveys the message to the receiver. The main problem here is that it leaves the patient with no privacy. Digital solutions such as eye-tracking devices or a BCI combined with a speech generator solve the privacy problem. However, all techniques are still very slow, and a response from a LIS person can have a time lag of 10-15 minutes. The possible communication platforms for a person with LIS are visualized in fig. 3.1. Since a person with CLIS can not perform any voluntary movement, BCI communication may be their only viable communication platform [7].

In 2021, a study was conducted in the Netherlands about the preference of possible BCIs for persons with LIS [61]. The preferences were surveyed by questioning persons with LIS, caretakers, and international BCI researchers. The study concludes that the most wanted feature in a BCI is *direct personal communication, private conversation and writing,* and *general computer use.* The most desired mental strategies were *attempted speech* and *attempted hand movement.* The latter can be seen as Motor Imagery (MI), the imagination of a movement without moving the affiliated muscles. It is important to note that the questionnaire asked what they would prefer to use instead of the feasibility of the mental strategy. An example of this is that MI is not intuitive, and many people find it challenging to do and learn [62]. It is also fatiguing as it demands high cognitive effort, which leads to poor separability of features after a period of use. Good separability is essential for good accuracy when classifying the signals [63, 64].

[1] Note that this is an updated and extended version of the literature review in the author´s previous work [1].

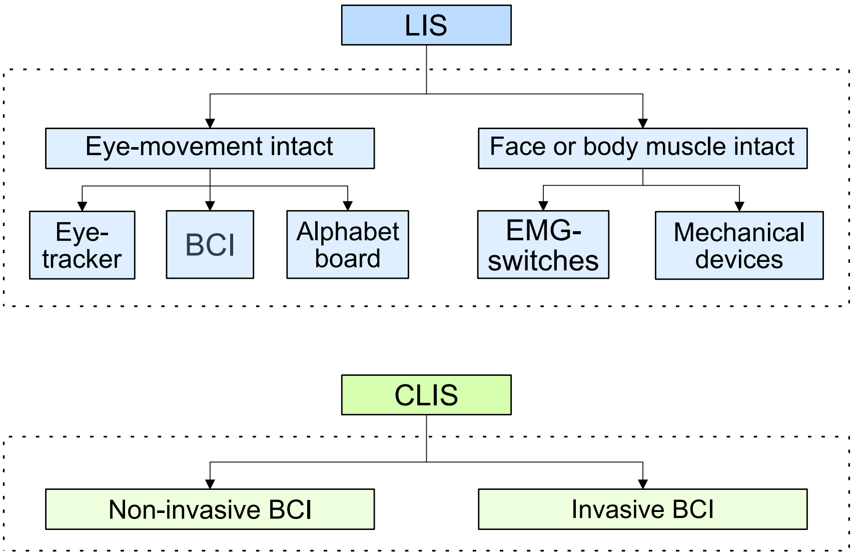


Figure 3.1: Possible communication for a person with LIS and CLIS. Electromyography (EMG) is a technique for recording of electrical signals generated by muscle activities.

3.2 Current LIS BCIs and color perception studies

A BCI creates a connection to a computer and does not necessarily need to be used for communication. Other controls can be used with the same BCI to control, for example, the environment or a wheelchair. In [65], sensorimotor rhythms from the eyes were used to create a controllable wheelchair. It had six possible states classified with RF. With the ongoing research on smart homes, a BCI can be used to control applications, such as lights and television in a house [66], increasing the QOL for LIS patients.

Several types of EEG-based BCI designs, both invasive and non-invasive, for helping paralyzed patients are researched. Invasive BCIs have a much higher success rate, but though it is expensive, difficult to use, and dangerous, so, non-invasive BCIs are preferred. Three different non-invasive BCI paradigms are [67, 68]: 1) sensorimotor rhythm BCI from, for example, MI, 2) P300-based BCI, event-related potentials, 300 ms after a new surprising stimulus, and 3) Steady-state visually evoked potential (SSVEP), produced when stimulated with flashing visual stimuli [69].

Creating a communication platform for persons with CLIS has proven very difficult. Most BCIs use two to four class protocols, and the current state-of-the-art BCIs lets the user create sentences in roughly half an hour. A binary communication platform based on MI was created in [70], where the EEG signals from the person with CLIS were classified with Riemannian geometry. In [69], SSVEP with flickering led lights in different colors were used to create a 4-class protocol for the CLIS user. [71] used P300 speller for a person with LIS towards being in a completely locked-in state. Though the methods worked, the patient used around an hour to create a sentence.

A fourth BCI paradigm is the neurological stimuli created from exposure to colors. In [72], SSVEP was used to flash three colors, and the results showed that it was possible to classify the EEG signals to a color. Two master thesis's [73, 74] showed promising results for classifying three colors when the subjects were exposed to continuous colored light. Classification between color exposure and resting-state has also been proved feasible by the study [75], where the best results were created with a SVM classifier. Using color exposure in a BCI is beneficial because it requires minimal user training and a low cognitive effort. In the study, [76], a BCI using color stimuli and eye movement were designed to create a communication platform between a person with LIS and a caretaker. The person with LIS was exposed to several colored options with icons on a screen linked to predefined messages to the caretaker. A summary of the studies and their results can be viewed in table table 3.1.

Table 3.1: Summary of researched BCIs for LIS patients and color classification studies. All studies are presented in section section 3.2.

Reference	Year	For whom?	BCI paradigm	Approach	No. classes	Results
[69]	2018	CLIS	SSVEP generated with flickering colored led lights	Calculating power spectral density	4	Accuracy 83.3 %
[70]	2019	CLIS	MI, 5 seconds epochs	Riemannian geometry	2	Online Accuracy of 87.5 %
[71]	2014	LIS patient towards CLIS	P300 speller, 0.8 seconds epochs	Stepwise linear discriminant analysis	3 × 6	Accuracy 81.5%
[72]	2020	Color perception study	SSVEP, flash colors	analysis of variance	3	Possible to differentiate colors
[73]	2019	Color perception study	Color exposure	Feature driven ML	3	Accuracy 63 %
[74]	2020	Color perception study	Color exposure	Riemannian geometry	3	Avereage accuracy above 80 %
[75]	2019	Color perception study	Resting state against color exposure	SVM	2	Accuracy 93.27 %
[76]	2017	LIS	Color perception and eye movement	Fuzzy logic pattern recognition	6	Worked for the patient

3.3 State-of-the-art EEG classification

A designed BCI often neglects the users' considerations, and its classification technique is evaluated with data from a closed and safe research environment. Implying that the distance between daily life use and research is too large. A BCI must work in an online environment and withstand real-life noise to be helpful and improve the QOL to the user [8].

The most commonly used classifier for the classification of EEG signals is SVM. SVM is easy to implement, learns fast and accurately, and often outperforms other classifiers on both small and large training datasets. RF classifiers have also been shown to perform very well with EEG signals [8]. In [77], SVM and RF are among the three best-performing methods after evaluation on 71 different datasets.

Although DL has been doing best in many research fields, it has yet to perform with EEG signals. DL can make the processing of signals redundant as it trains its own filters, implying that information lost in the data analysis and feature extraction is available for the neural network. Another reason why research on EEG classification is beneficial is the possibility of creating Generative Adversarial Networks. These networks can be used to create fake EEG data, which can be used to pre-train networks, improve the classification performance, and enlarge datasets. In [78], a deep CNN was used to get good results on an EEG dataset, and the authors concluded that a deep network is necessary. They also highlighted the normalization of the input data and the activation function ELU as essential factors for good performance.

The problem with EEG datasets is their data limit, and large datasets are a rarity. Deep networks often have performance issues because the model is overfitted on the training data. So [8] argues that shallow networks are more promising as they have fewer parameters to train and, therefore, can execute well on small datasets. However, a shallow network will lose the advantage that depth brings in a neural network. The study [56] designed a generalized CNN that should work on different types of EEG data, called EEGNet. It is a shallow network that uses different convolution methods, depthwise and separable convolution, instead of depth as the advantage.

There are two other state-of-the-art methods for classifying EEG data that exist worth mentioning, though they are not experimented with in this thesis. The first is Shrinkage linear discriminant analysis, which works well with little data. The second is Riemannian geometry, which has delivered good results in several types of EEG data [8]. Riemannian geometry was also the classification technique that performed best when classifying colors in [74].

Chapter 4

Methods

The following chapter explains the steps used to evaluate the feasibility of the BCI. The hardware and software are first described. Then the protocols and data acquisition is explained in section 4.2. The datasets used in the evaluation are explained in section 4.3. Then, the thesis's central core is presented, which is the combination of pre-processing and feature extraction, described in section 4.4, and the initialization of the classification techniques, presented in section 4.5. Finally, the evaluation methods used in the thesis are defined.

4.1 Hardware and software tools

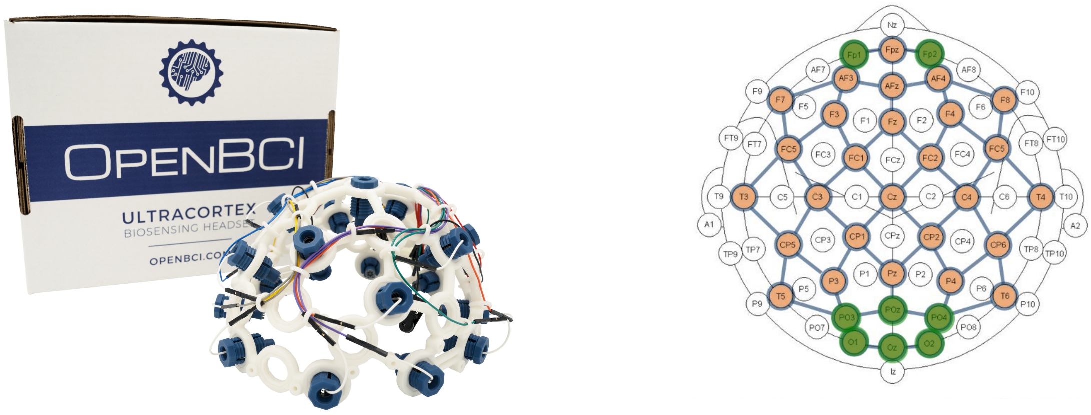
OpenBCI headset

The EEG signals were recorded with an Ultracortex "Mark IV" EEG Headset, see fig. 4.1a, created by OpenBCI[1]. The headset records EEG data from up to 16 dry electrodes at 125 Hz and transfers it to a computer via Bluetooth. However, if the number of electrodes is kept at 8 or below, the sample rate is 250 Hz. The difference in sample rates comes from the limitations of the Bluetooth bandwidth. The electrodes can be placed in 35 different locations, shown as orange-colored circles in fig. 4.1b, matching the international 10-20 system. Two extra electrodes are being placed on the ear lobes. These serve as reference and ground, which aid in rejecting the common-mode noise from the environment.

Software tools

Data acquisition and analysis was mainly done with python3 code and libraries. Implementation of the CNN was done with *TensorFlow* [80], while the other ML methods used *scikit-learn* [81]. The optimization technique NSGA-II was implemented with the library *pymoo* [82]. The feature extractions and most of the decomposition methods used the libraries *Numpy* [83], *Scipy* [84], and *MNE* [85]. The exceptions are DWT that used *PyWavelets* [86] and VMD that called on a Matlab function.

[1] `https://openbci.com`


OPENBCI
ULTRACORTEX
BIOSENSING HEADSET
OPENBCI.COM
Nz
Fp1 Fpz Fp2
AF7 AF3 AFz AF4 AF8
F9 F7 F5 F3 F1 Fz F2 F4 F6 F8 F10
FT9 FT7 FC5 FC3 FC1 FCz FC2 FC4 FC5 FT8 FT10
T9 T3 C5 C3 C1 Cz C2 C4 C6 T4 T10
A1 A2
TP9 TP7 CP5 CP3 CP1 CPz CP2 CP4 CP6 TP8 TP10
P9 T5 P5 P3 P1 Pz P2 P4 P6 T6 P10
PO7 PO3 POz PO4 PO8
O1 Oz O2
Iz

(a) The Ultracortex "Mark IV" EEG headset

(b) Electrode placement on the headset, the electrodes used in this thesis are marked with green.

Figure 4.1: The OpenBCI headset and the electrode placements, colored orange and green, viewed in the international 10-20 system. Both images are reprinted from [79].

The IDUN cluster

The high-performance cluster IDUN [87] was used to perform the data analysis and optimization. The cluster has more than 90 GPGPUs and 70 nodes, where each node contains two Intel Xeon cores and at least 128 GB of main memory. It is a platform for testing and prototyping high-performance computing software, graphic processing unit simulations, and design-space exploration. In this thesis, the cluster was used to train models and optimization search with the NSGA-II algorithm.

4.2 Data acquisition

The BCI for the simple communication platform is designed to be a computer screen that shows several options for the user, see fig. 4.2. Each option is represented with color and an icon, where the icon explains the option to the user. The study [76] inspired this design, which was chosen as the design is modular, with switchable options. This BCI is also the same design and protocols used in [1].

4.2.1 Electrode placement

The designed BCI uses the brain's color perception and eye movement detection to work. It is clear that the crucial areas are the occipital lobe, especially V4, and around the orbita, as seen in section 2.2. It is problematic to record directly from V4 with non-invasive methods, as V4 is covered by other brain regions. The solution is to record from areas close to its vicinity since the EEG signal is a sum of activations from several neurons. It is essential to get a good reading of vertical eye movement from the different possible eye movements. A person with LIS has more frequent vertical eye movement intact and not horizontal movement. The vertical eye movement is best read from electrodes just above the eye. Therefore, the following electrode placement in the 10-20 international system is used:

Frontal lobe: *Fp1* and *Fp2*

Occipital lobe: *PO3*, *POz*, *PO4*, *O1*, *Oz* and *O2*

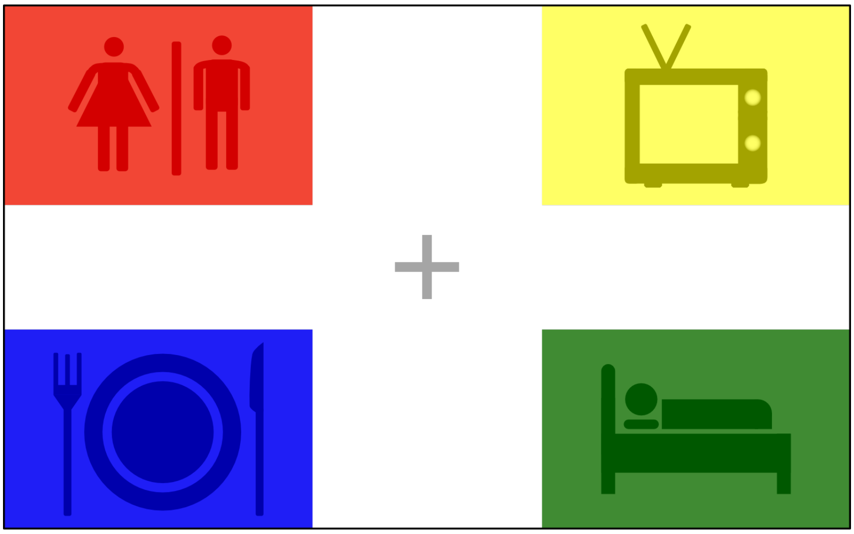
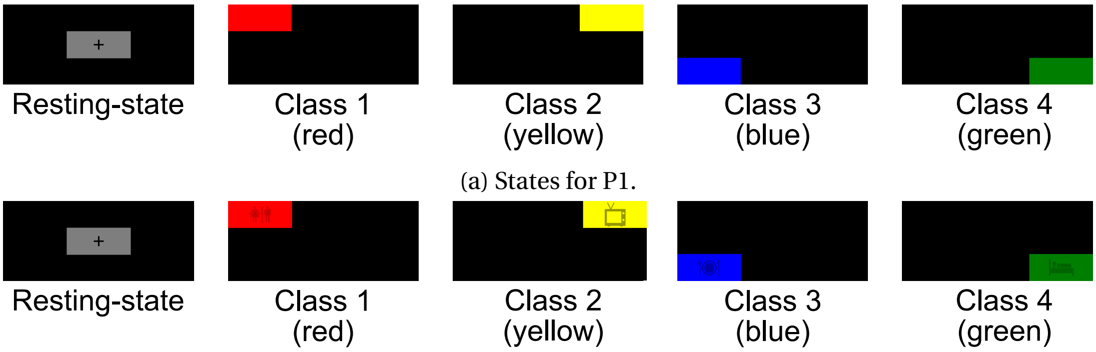


Figure 4.2: Example of the option layout for the designed BCI.

The reference and ground electrodes were placed on the right and left ear lobes, respectively. The EEG signals recorded from the frontal lobe was done with *flat* electrodes, while the six other was recorded with *comb* electrodes. The placement of the electrodes is visualized in fig. 4.1b, where the used locations are colored green. Because the total number of electrodes did not exceed 8, the EEG signals could be recorded with a sampling rate of 250 Hz.

4.2.2 Protocols

A protocol that can record the brain activity when a user looks at the different options needs to be designed. The same protocol framework used in [1] was used again. They consist of five different states that are shown to the participant in random order. One of the states is the resting state, while the reminding four are different task states. The difference between the two protocols is if a task state includes an icon or not in addition to color. From now on, the data recorded with only colored states will be referred to Protocol 1 (P1) and data recorded with colors and icons as Protocol 2 (P2). A visualization of the two protocols can be seen in fig. 4.3.



(a) States for P1.

(b) States for P2.

Figure 4.3: A visualization of the possible states in both protocols they are presented as they were displayed to the participants when collecting their EEG data.

Common for both protocols is the layout of the states and their respective colors. The resting state is displayed as a gray box in the middle of the screen. There is a cross in the middle of the box to help the participants not lose concentration on the state. The four task states have the colors red, yellow, blue, and green, and are placed in the upper left, upper right, lower left, and lower right corners, respectively. The placement of the task states was chosen to maximize the eye movement of the user from the resting state to a task state. The task-states appeared one at a time for 2 seconds and were always followed up by a rest-state. In contrast, the resting state was displayed for a 2-5 seconds random duration. See fig. 4.4 for a visualization of the timing scheme for the protocols. The task states came in a random order to break possible prediction patterns created by the brain.

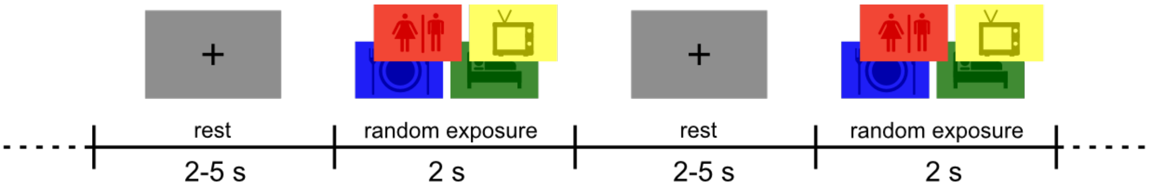


Figure 4.4: The protocols timing scheme. Note that the visualization shows the P2 task-states, P1 has equal timing scheme though its task-states are only colors.

A session consisted of 20 samples of each task state and ≥ 79 sample of the resting state. A session took a minimum of 5 minutes and 20 seconds, though the average time for a session was ≈ 7 minutes. If the random resting-state time was ≥ 4 seconds. The rest state was split into two samples, which increased the number of resting states in the dataset.

4.2.3 Screen

A wide curved computer screen with a display size of 34 inches was used to present the events to the subject. Using a wider screen generates a greater distance between the tasks, implying that the subjects have to increase their eye movement, making the signal more prominent. In addition, the different task states would project a larger area of its color. The respective hex number of the colors presented on the screen can be viewed in table 4.1. The screen's luminosity was constantly at 300 cd/m^2, candela per square meter, and the subjects were distanced 1 meter from the screen.

Table 4.1: The hex values for the colors presented on the screen for the participants during the data acquisition in both protocols. *RGB* is an abbreviation for red, green, and blue.

Color	Black	Gray	Red	Yellow	Blue	Green
Hex (RGB) background color	# 00 00 00	# 80 80 80	# FF 00 00	# FF FF 00	# 00 00 FF	# 00 80 00
Hex (RGB) icon color	# 00 00 00	-	# CC 00 00	# 99 99 00	# 00 00 AC	# 00 60 00

4.2.4 Limitations

A common drawback to BCI experiments is to obtain enough data due to subject availability and mental tiredness. There are also some specific limitations related to the specific hardware used in the data acquisition. The OpenBCI headset is wireless and transfers all its data to a computer via Bluetooth over limited bandwidth. During this transfer, bytes can be lost or corrupted. The headset can initialize with another sample rate than the expected one on initialization, usually a difference of ±2 from the expected one. This problem was solved by having an external thread continually calculating the sample rate, so the correct sample rate could be used when epoching the continuous raw EEG signal. The headset also creates a lot of electrical noise when connecting to a computer, which makes the first recorded seconds useless. This problem was solved by always having the first state be the resting state, which data were removed when saving the recorded data. The final problem with the data acquisition is that the headset is in only one size. Adjusting for different head sizes is done by tightening and loosening the electrodes, which are screwed into the headset. The problem with the headset size makes it hard to place the electrodes correctly according to the 10-20 international system. This problem is especially true for persons with smaller heads.

4.3 Datasets

Three different datasets have been worked on within this thesis. The first is a public dataset, part of the BCI Competition IV[2], while the two others were created with the protocols described in section 4.2. One was created in spring 2022, dataset A, while the other, dataset B, was created for the work in [1]. Dataset A was created because problems and errors were uncovered in dataset B after the recordings were finished. Dataset A is the most used one in the experiments, and if not stated otherwise, assume that dataset A is used to create the results. Common for both dataset A and dataset B are the electrode placements and the physical location the recording was held. The blinds were kept down, and the ceiling lamp was turned on during the experiments. So, all subjects were exposed to the same lighting in the room. In other words, the light sources in the room were from the computer screen and the ceiling lamp. The room was not sound isolated, and the subjects experienced a different degree of sound noise from either construction work in the vicinity or loud talking from the hall.

All subjects were invited to the project with a written letter. The letter explained the purpose of the study, what would happen during the experiments, risks, data privacy, their rights during the experiments, and that the data might be used beyond this study. In addition, they were obligated to inform of their gender, age, epilepsy history, and color blindness. Upon arrival, all participating subjects signed a consent form, where they agreed that they had read and understood the content of the letter and that they wanted to participate in the experiments. The consent form can be found in appendix A. During the experiments, the subjects were informed to sit relaxed and only move their eyes, nothing else. Furthermore, if there was a need to blink during a session, the participants were told to do it when a resting state was showing. The reason was that there is a lot more resting state data than specific task state data.

4.3.1 Public dataset

This data was part of the BCI Competition IV, held in 2009, and is dataset 2a on the site. The dataset consists of the recordings from nine participants who performed four different MI tasks. The imagery

[2] https://www.bbci.de/competition/iv/

movements are left hand, right hand, both feet, and tongue. All participants performed two sessions, and each session contained 288 trials. Twenty-two electrodes were used to record the EEG signals at a sample rate of 250 Hz before it was bandpass filtered between 0.5 Hz to 100 Hz and a notch filter at 50 Hz. In addition, the eye movement was recorded with three electrooculograms (EOG) channels, which could be used to remove artifacts from the eyes. This thesis did not include data from these channels when working with this dataset.

4.3.2 Dataset B

Several experiments have been conducted on this dataset already, and the results from those experiments are presented in [1]. A total of 33 healthy subjects participated in the experiments to create this dataset. They chose to complete 4 or 6 sessions, either two sessions with P1 and P2 or three sessions with P1 and P2. Only nine subjects wanted to finish after the completion of four sessions. One of the subjects, subject 3, was colorblind and had trouble differentiating between red and green. There were 15 female subjects and 18 males, and the average age was 24.9 years old. The youngest participant was 19 years old, and the oldest was 34.

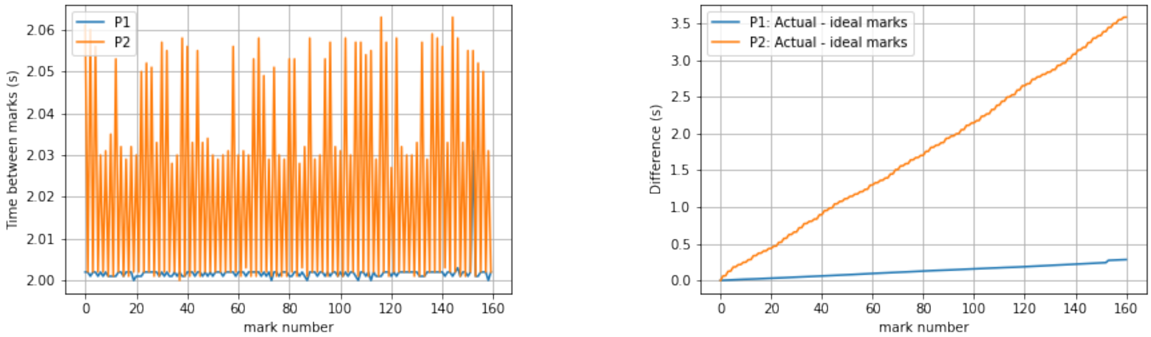
This dataset has three main problems, making it less ideal for experiments. The first problem was the lack of metadata collected during the experiments, in the form of questions to the subjects that could help remove noisy and unfocused sessions. Secondly, the rest state and task states were of constant length, 2-seconds, all the time. The periodic pattern increases the chance of rhythmic noise in the signals. The last problem and most severe problem is the recording of marks, visualized in fig. 4.5.

It was noticed in [1] that sessions based on P2 took ≈ 2 seconds more time than P1, and the conclusion was that P2 had a longer initial load time. By examining the recordings, it was shown that the 2 seconds difference comes from a delay between all state changes that were several milliseconds longer than expected, visualized in fig. 4.5a. Important to note that the time between two marks should ideally be 2 seconds. Creating a shift between the raw signals and the marks, visualized in fig. 4.5b, which in the end becomes so large that the EEG signals no longer correspond to the correct mark. The delay was highest for P2 data because of the loading time for the icons. Because only ideal marks, 2 seconds between each state change, were recorded, not the actual timestamps. All data from P2 are useless, while the P1 can be used with caution as there still is a significant time shift in the last marks.

4.3.3 Dataset A

The problems related to dataset B were fixed in this dataset. The rhythmic noise was reduced by using a random time length on the resting state, and the marker problem was fixed by recording the actual times instead of ideal marks. A questionnaire was used to gather info that could be used to explain results and remove poor sessions. The questions in the questionnaire can be found in appendix A, table A.1.

22 healthy subjects participated in this dataset, whereas eighteen were also participants in dataset B. The participants with the numbers 17, 18, and 19 had never conducted an EEG recording before. The age ranged from 20 to 28 years, with the dataset average of 24.5 years old. There is a gender



(a) The actual time difference between two marks for a P1 and P2 session.

(b) How long, in seconds, a mark are shifted from the actual mark for a P1 and P2 session.

Figure 4.5: Visualization of the marker problem with dataset B. The plots are created with dummy data generated by running a session without anybody wearing the headset.

imbalance in the dataset, with 14 female subjects and 8 males. The participants chose between 4 or 6 sessions, and all except one chose to complete six sessions. After three sessions, all participants removed the headset and paused for fresh air and walking before continuing the last sessions. The pause was performed to improve the focus and reduce mental tiredness among the participants.

4.3.4 Summary of the two main datasets

The two datasets used in this thesis are summarized in table 4.2. Both datasets are designed with the same BCI and communication system in mind. The P2 data from dataset B are unusable, so to compensate, there is a slight imbalance in the number of sessions between the protocols in dataset A. Furthermore, dataset A was created because of the problems in dataset B. Table A.2 in appendix A can be used to see which number of subjects participated participating in both datasets has in the respective datasets.

Table 4.2: Summary of dataset A and dataset B. Where the abbreviations "no." stands for *number of* and "tot." for *total.*

Name	no. subjects	tot. sessions	P1 sessions	P2 sessions	no. states
Dataset A	22	130	60	70	5
Dataset B	33	89	89	-	5

4.4 Data processing methods

There are several ways to process the data, and the main methods used to create the different classifiers will be explained here. These are the primary methods and are used in the experiments if nothing else is stated. Note that different classifiers use different processing pipelines. All processing pipelines use only one epoch as input to mimic an online environment. In this thesis, a sample and epoch are the same and are two seconds of EEG signals from all eight channels starting from a specified cue point.

4.4.1 Dataset splitting and balancing

All three datasets were split into three sets: a training set, a validation set, and a test set to ensure that the evaluation of the model could be done unbiased. The training set got 60% of all the epochs, while the validation and test set have 20% each. The splits are done pseudorandomly; the splits are done the same *random* way for each experiment. The pseudorandom splits ensured that the epochs used in training and evaluation did not affect the performance of the classifiers and that a comparison between models was not affected by different samples in datasets. A class balancing of each set was performed, so each set included an equal amount of epochs from each class.

4.4.2 Pre-processing

All data used the same pre-processing pipeline, with the exception in section 4.4.4. The pipeline is visualized in fig. 4.6, where an epoch is the input. First is the power-line noise removed with a Notch filter set to 50 Hz. Then the signal is enhanced by removing similar noise in all channels with CAR. Before the sample is bandpass filtered between 0.5 Hz and 50 Hz, as this is the frequency range where the essential information from the brain exists, as illustrated in section 2.3.1.

Models created with CNN, here EEGNet architecture, primarily used only the pre-process pipeline on the samples before training, validating, and testing. If EEGNet used another pipeline, this would be stated. The reason is that a CNN trains its own filters that extract the relevant features from the signals. These filters can find useful information in the raw signals, which a feature extracted signal neglects.

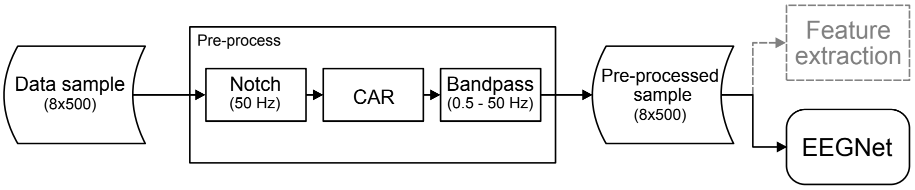


Figure 4.6: The pre-process pipeline for an epoch. The pre-processed sample will be used in either the CNN: EEGNet, or in a new feature extraction pipeline.

4.4.3 Feature extraction from DWT components

This processing pipeline is only used when using SVM and RF classifiers, visualized in fig. 4.7. The channels in the pre-processed epoch are, one at a time, decomposed into five frequency bands with DWT, four levels, with a Biorthogonal 2.2 mother wavelet. Theory about DWT can be found in section 2.6.1. From the components are four features extracted, instantaneous and Teager energy, and the fractal dimensions PFD and HFD. The features from each channel are flattened as well as the channels. So, the final output is a 1-dimensional vector, which is passed to a SVM and a RF classifier.

4.4.4 Parallel signal processing

The output from the pipeline, called Parallel signal processing (PSP) in this thesis, is used as a processing method to extract more spatial features from the raw signal. It utilizes another pre-processing

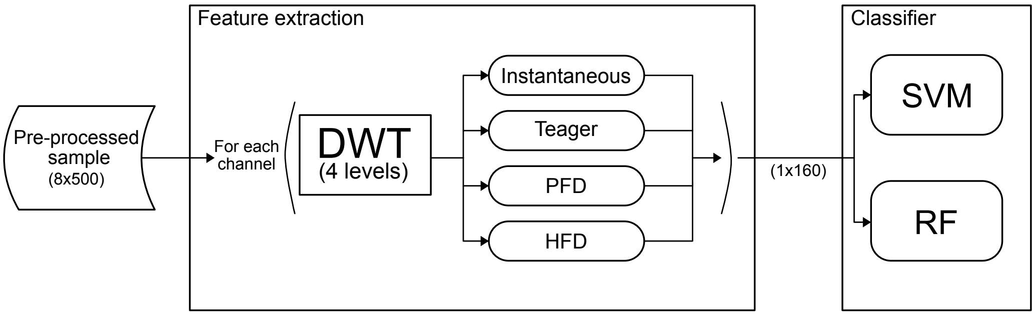


Figure 4.7: The feature extraction pipeline used on each epoch used by the SVM and RF classifiers. Note that the input are a pre-processed sample.

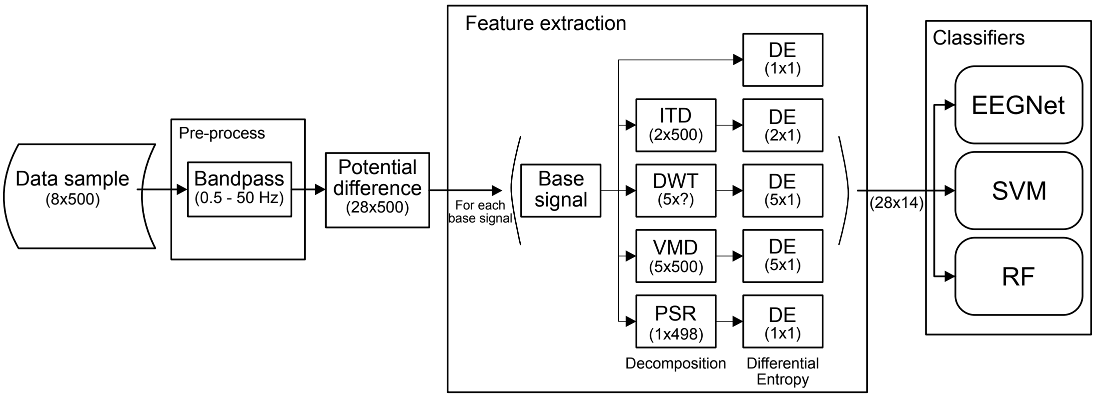


Figure 4.8: The PSP pipeline, where numbers in parenthesis are the shape of the data after the computation in the respective method. The shape after DWT is (5x?) because the different components have variable lengths. Note that the input is a raw data sample with its own pre-processing.

technique to extract better features from the sample and is visualized in fig. 4.8. First, a epoch is *only* bandpass filtered between 0.5 Hz and 50 Hz. Then, *base signals* are constructed by calculating the potential difference between all channels. The base signal creation is done to improve the spatial resolution of the original EEG signals. The potential difference is calculated by first calculating the difference between the first channel with the seven others, creating seven new signals, then the difference between the second channel and the reminding six channels, then the third with the reminding 5, and so on. The calculation is completed when the second to last channel calculates its difference with the last channel. Since Dataset A has eight channels, this method will create 28 new signals.

The base signals are input in four different decomposition and reconstruction techniques. Before the DE from the base signal, the components and reconstructions are calculated. The decomposition techniques used are: ITD, using the first two PRCs, DWT, with the same mother wavelet and level as in section 4.4.3, and VMD, which is set to create five IMFs. The reconstruction of the base signal was done with PSR. Implying that after PSP the sample has the dimensions 28 × 13. The theory behind these methods can be found in section 2.6.

4.5 Model architecture and hyperparameters

The model and hyperparameters defined in the following section are always applicable if not stated otherwise. A hyperparameter is a non-trainable parameter defined to a classification method before that method starts training.

SVM

The SVM classifiers were initialized with the scikit-learns [81] default parameters. The model uses the Radial-basis function (RBF) kernel, which computes how close two points are by looking at the exponential Euclidean distance. No approximation of this kernel was needed, as the number of training samples and features is low. A SVM model is created with 5-fold cross-validation to improve their knowledge of the data.

RF

The RF classifiers are initialized with several hyperparameters that differ from the default ones. Firstly, the forest is limited to 100 DTs, where each tree is capped to a depth of seven. Restricting the depth makes the classifiers less likely to overfit. The random number generator that controls the randomness in bootstrapping is seeded to better evaluate the different RF classifiers. The Gini impurity evaluates the quality of a split and is used to decide on a split.

CNN: EEGNet

The CNN architecture EEGNet [56] is used to create the classification models. The strength of EEGNet is its usability on different types of EEG signals, performance on limited data, and ability to produce neurophysiologically interpretable features. A simplified version of its architecture can be viewed in fig. 2.5, while its complete architecture is showcased in table 4.3.

EEGNet uses both depthwise and separable convolution. The depthwise layer helps spatial filters learn from temporal filters, enabling efficient frequency-specific extracting of spatial filters. Separable convolution helps with separating the learning, how to summarize individual feature maps, with how to combine the feature maps optimally. The layer's number of temporal and spatial filters is decided by the hyperparameters F_1 and D, respectively. The hyperparameter to affect this learning is set by F_2. If F_2 equals $F_1 \cdot D$, it learns an equal amount of feature maps as filters in the depthwise layer. However, if F_2 is lower than $F_1 \cdot D$, it learns fewer, implying a compressed representation, while a higher number implies an overcomplete representation.

The hyperparameters of the network use its default values decided by [56], $F_1 = 8$, $D = 2$, and $F_2 = 16$. The kernel length was initialized to close to half the sample rate, 128. The loss functions used were *binary cross-entropy* or *categorical cross-entropy*, depending on the number of classes. Furthermore, the network was optimized with the *Adam* algorithm with a learning rate of 0.001.

Table 4.3: The EEGNet architecture copied from [56], where C = number of channels, T = number of time points, F_1 = number of temporal filters, D = depth multiplier (number of spatial filters), F_2 = number of pointwise filters, and N= number of classes.

Block	Layer	# filters	Size	# params	Output	Activation	Options
1	Input				(C, T)		
	Reshape				(1, C, T)		
	Conv2d	F_1	(1, 64)	$64 \cdot F_1$	(F_1, C, T)	Linear	Mode = Same
	BatchNorm			$2 \cdot F_1$	(F_1, C, T)		
	DepthwiseConv2D	$D \cdot F_1$	(C, 1)	$C \cdot D \cdot F_1$	($D \cdot F_1$, 1, T)	Linear	Mode = valid, depth = D, Max norm = 1
	BatchNorm			$2 \cdot D \cdot F_1$	($D \cdot F_1$, 1, T)		
	Activation				($D \cdot F_1$, 1, T)	ELU	
	AveragePool2D		(1, 4)		($D \cdot F_1$, 1, T // 4)		
	Dropout				($D \cdot F_1$, 1, T // 4)		p = 0.25 or p = 0.5
2	SeparableConv2D	F_2	(1, 16)	$16 \cdot D \cdot F_1 + F_2 \cdot (D \cdot F_1)$	(F_2, 1, T // 4)	Linear	Mode = Same
	BatchNorm			$2 \cdot F_2$	(F_2, 1, T // 4)		
	Activation				(F_2, 1, T // 4)	ELU	
	AveragePool2D		(1, 8)		(F_2, 1, T // 32)		
	Dropout				(F_2, 1, T // 32)		p = 0.25 or p = 0.5
	Flatten				($F_2 \cdot (T//32)$)		
Classifier	Dense	$N \cdot (F_2 \cdot T//32)$			N	Softmax	Max norm = 0.25

4.6 Transfer learning and warm start

Training a model from scratch is both time and computationally expensive and can be a dealbreaker for a BCI. A solution is to use and update the knowledge learned in a previous model with new data. The advantage of using a pre-trained model as a base is that the training time with new data will decrease, while the performance may increase. The pre-trained model will be trained by cross-selecting subjects and sessions from datasets A and B. SVM creates a model that has solved an optimization problem. Reusing that model as a start for a new one with new data makes no sense as this is a new optimization problem. Reusing the knowledge from a pre-trained model is called transfer learning for a Neural network (NN), while RF uses a warm start.

Transfer learning for a NN utilizes that the weights are already trained with a different dataset, though structure and labels are similar to the new data. There are several ways to use the pre-trained model. The three most common methods are: to continue training all the weights with the new dataset, only train some hidden layers in the network with the new dataset, or lastly, to use the pre-trained network to extract features and add new layers on top of it that will be trained with the new dataset. This thesis will use the first method as EEGNet has few trainable parameters and because the pre-trained model will be trained on data equal to the new dataset. The only difference is that the new data will be from new subjects and sessions.

The **warm start** method used in an RF classifier utilizes already trained estimators, decision trees, created with a different dataset. When RF trains on new data, the previously trained trees will make up an existing forest, and the new trees trained with new data will be added to the existing forest. The generalizability learned from training on a cross-selected subject and session dataset will be utilized, while the specific and essential information in the new dataset will be used when the new model makes predictions.

4.7 Evaluation framework

In order to get a good unbiased evaluation of the models, evaluation metrics that can describe the predictions on new, unseen data need to be defined. The steps to get an unbiased evaluation are explained in section 4.4.1. CNNs are black boxes, sot it is relevant to understand how the CNN is learning from samples and predicting an epoch.

Metrics

The predicted labels from a model are compared with the actual labels and can create a confusion matrix. A confusion matrix displays the performance of the model's predictions and is an essential tool to visualize the exact predictions the model has performed. The diagonal in a confusion matrix will always display the number of true predictions for the corresponding label. At the same time, the rest of the rows and columns respective to the true predictions display false negative and false positive.

The confusion matrix can be used to calculate the accuracy score. Informally it is defined as the fraction of samples a model predicted correctly. A problem with the accuracy score is that it has a problem giving a correct picture of the model when the data is imbalanced—implying the importance of balancing the datasets to get a sensible accuracy score.

Another metric that can be calculated with the confusion matrix is Cohen's kappa, which is primarily used to evaluate one experiment in this thesis. It calculates the level of agreement between two raters, corrected for the frequency the raters may agree by chance. It returns a number between -1 and 1, where all results less than zero imply no agreement. It also has a more realistic view of imbalanced datasets than accuracy.

The final metric used in this thesis is a Receiver operating characteristic (ROC) curve and evaluation of the Area under curve (AUC). This metric is only available in a binary classification problem and plots the true positive rate against the false positive rate for those classes. The ROC curve and its AUC score can explain how good a model is at differentiating between two classes, information that is lost in, for example, the accuracy score. A curve that touches the upper left corner perfectly differentiates the two classes. Which is the same as the AUC score is 1.0. However, when the AUC score is close to 0.0, the model cannot predict correctly nor manages to separate the classes. An AUC score of 0.5 means that the model did not manage to differentiate between the classes.

Explainability of a CNN

The study [56] proposes three different methods to describe how the CNN is learning. One of the methods is with a technique called DeepLIFT [88], which is short for Deep Learning Important FeaTures. DeepLIFT calculates the relevance of individual features when giving a specific input. The method compares the activation for each neuron to a reference activation and assigns respective scores according to the difference by performing backpropagating on the input. The method can be used to explain which features are essential in a high-confidence prediction versus a low one. In this thesis, the method explains which channels are important when predicting with EEGNet, and from what period in the epoch is vital for a prediction.

Chapter 5

Results

The following chapter presents experiments conducted on the datasets and their respective results. The results are mainly displayed in plots, and accuracy is the primary evaluation metric. So if not stated otherwise, the results in a plot show the model's accuracy created from predicting on a test set. The 11 experiments can be categorized into three subgroups. The first group evaluates the proposed processing pipeline and classification methods on the public dataset and only contains one experiment, section 5.1. The next group explores different processing pipelines and feature extraction methods on models created with data from only a single subject. This group consists of the experiments from section 5.2 to section 5.7. The final group of experiments aims to create a general model that can perform well on all subjects and includes the last four sections, section 5.8 to section 5.11.

5.1 Processing and model evaluation on the public dataset

The objective of this experiment is to validate the proposed methods for the classifiers in section 4.4 with the public dataset described in section 4.3.1. PSP was not evaluated on this dataset. The first session was used as the training set and the second for evaluation, as done in the competition. No changes to the hyperparameter or feature selection were made to improve the results, so the models were evaluated more accurately. The proposed methods are designed for color differentiation, not MI, which this dataset contains, implying that the expected performance of the model is lower than the winning model in the competition. The training set splits into three sets, where all sets had an equal amount of each class. The test set is discarded because the second session computed the final results. In other words, 20% of the training data was not used to create the models.

Table 5.1 presents the final results of the performance of the models with the public dataset. Note that a separate model was created for each subject. In other words, no generalized models were tested. Two different metrics are shown for each model; on top is the Cohen's kappa presented, while below, in parenthesis, is the accuracy. The accuracy is included as this is the primary evaluation method in the thesis.

Comparing the models, EEGNet is performing best, though just by a small margin, and SVM and RF have very similar results. More interestingly, the different methods prefer data from different subjects. For example, the score was very high for subject 1 when classified with SVM and RF but low for EEG-

Table 5.1: Results of the processing and classification evaluation done on the public dataset from the experiment in section 5.1. The upper value is Cohen's kappa score, while the lower, in parenthesis, is the accuracy of the models.

	Average	**Subjects**								
		1	2	3	4	5	6	7	8	9
EEGNet	0.42	0.16	0.35	0.35	0.38	0.61	0.40	0.63	0.43	0.44
	(0.57)	(0.37)	(0.51)	(0.51)	(0.54)	(0.71)	(0.55)	(0.72)	(0.57)	(0.58)
SVM	0.41	0.67	0.36	0.64	0.28	0.11	0.25	0.46	0.53	0.40
	(0.56)	(0.75	(0.52)	(0.73)	(0.47)	(0.33)	(0.44)	(0.60)	(0.65)	(0.55)
RF	0.41	0.67	0.36	0.64	0.29	0.11	0.25	0.46	0.53	0.40
	(0.56)	(0.75)	(0.52)	(0.73)	(0.47)	(0.33)	(0.44)	(0.60)	(0.65)	(0.55)

Net. The opposite is true for subject 5, where EEGNet has one of its best scores, but the kappa score to SVM and RF are close to zero. There is also a significant difference between the kappa score and the accuracy, implying a slight imbalance in the dataset. Finally, comparing the kappa scores with the results from the competition[1], all three models make the third place, implying that all proposed methods are usable for further EEG classification.

5.2 Baseline evaluation of individual subject performance, dataset A

The objective of this experiment was to create baseline models for each subject with the two first methods described in section 4.4, in other words, not the PSP method. The baseline models will give an overview of the initial performance of each subject with the defined methods. Furthermore, the results can be used to calculate the difference between a new experiment and this baseline result, making it easier to evaluate new experiments.

The graphs in this section are read in the following way. For each subject, three different classifying methods have been performed. These are EEGNet with only pre-processed data, and SVM, and RF, where fractals and energies are extracted from the components created with DWT. Only the accuracy score is plotted, simplifying the comparison between the methods. The average performance of a technique, calculated with results from all subjects, is written in parenthesis below the method's name in the plot legend. The red dashed line shows the chance level for that classification. **Resting-state vs task** is a binary classifier, implying a chance level of 0.5, while the **4-class** classifiers have four classes and a chance level of 0.25.

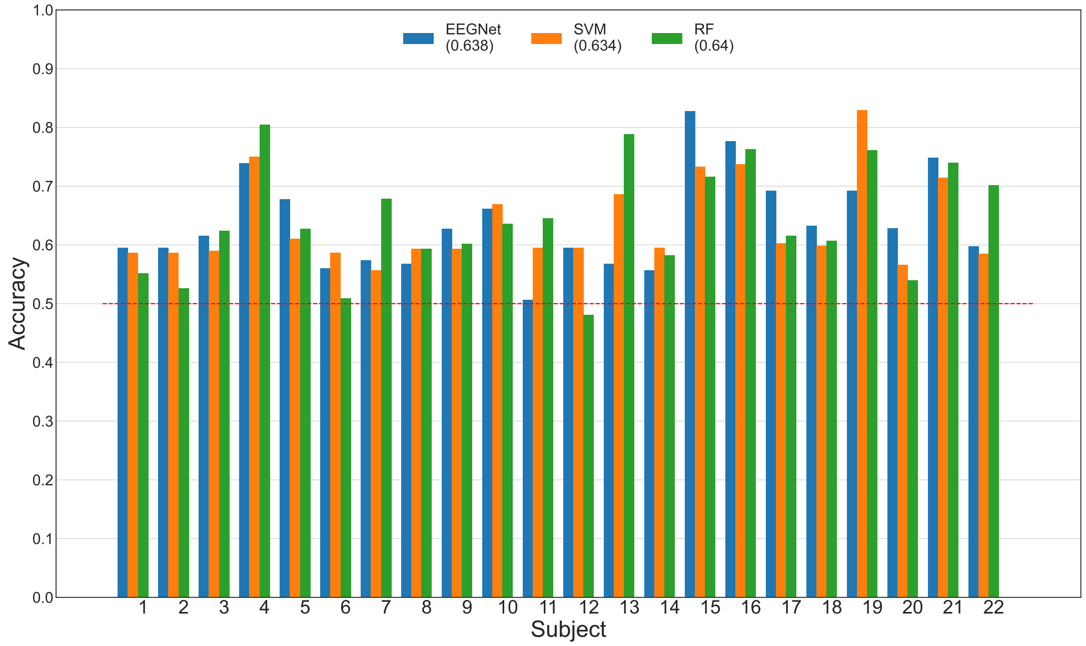
Figure 5.1 presents the results for the different methods used to create individual models with P1 data, dataset A. For the **resting-state vs task** classification, see fig. 5.1a, RF models perform better than the rest, though it is not significant. It is also the only model that went below the chance level when creating a model for subject 12. Subjects 15 and 19 created the highest accuracies with EEGNet and SVM models, respectively. Subjects 5, 6, 13, 15, 16, 17, 19, and 21 created the best individual **resting-state vs task models**. The results for the **4-class** classification display a more divergent performance of the methods, see fig. 5.1b. EEGNet has the best performance on almost all subjects and average perfor-

[1] The results for dataset 2a in the BCI competition IV, can be viewed at `https://www.bbci.de/competition/iv/results/#dataset2a`

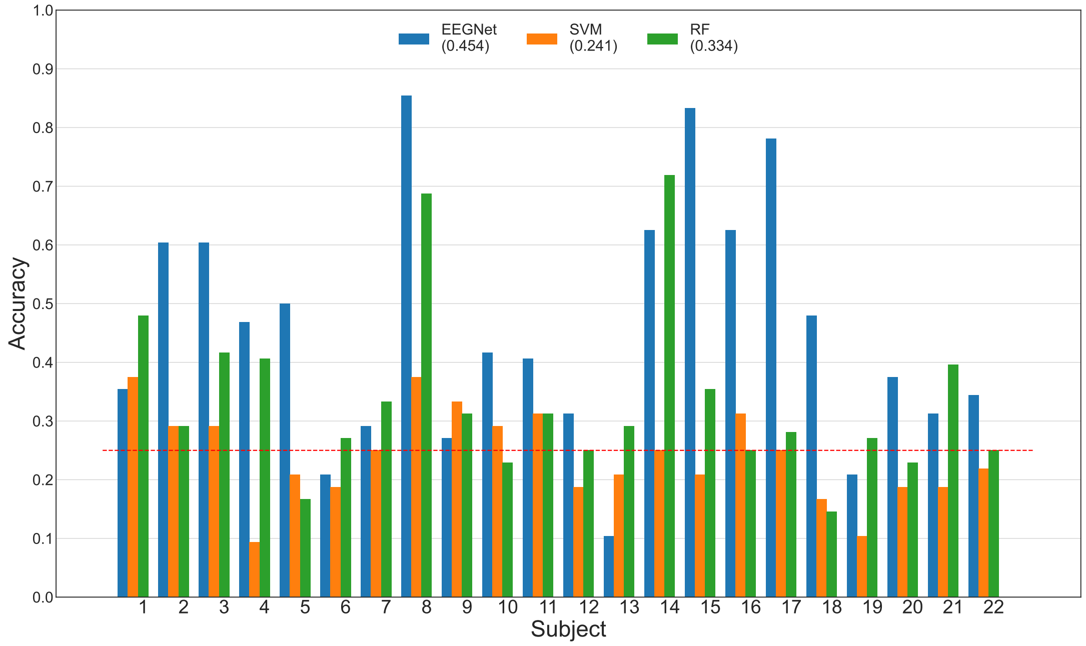
mance. The best performing subjects for EEGNet, subjects 8 and 15, have higher accuracy than the best subject in the **resting-state vs task classification**. However, it does not manage to get over the chance level for all subjects. RF manages to create some sporadic good models, while SVM struggles to perform better than chance. The subjects whose models managed to get accuracy above 0.6 were the subjects 1, 2, 3, 8, 14, 15, 16, and 17.

The results of the methods for each subject created with P2 data, dataset A, are visualized in fig. 5.2. In the **resting-state vs task** classification results, see fig. 5.2a, is RF the classification method that is performing best on average and has got the highest accuracy. However, all methods have very similar average performance. SVM has the most stable performance on all subject models, while RF and EEGNet have some models that are just above the chance level. The **resting-state vs task** models for the subjects 4, 5, 13, 15, 16, 19, and 21 were the best-performing ones. EEGNet is the best performing method when creating **4-class**, see fig. 5.2b, and none of the other classifiers are close to its performance. SVM models are primarily below the chance level, while RF has some models that are doing great. EEGNet has two models that had accuracy above 0.9, where the model for subject 2 is the best. The subject data that have created the best models are from the following subjects, 2, 3, 8, 14, 15, 16, 17, and 21.

There is a slight difference between which subjects perform best when comparing the best performing subject models between the two classification techniques. However, two subjects perform best in all four experiments, subject 15 and subject 16, where EEGNet is the method that creates the best results. It is also evident that models made with P2 data generally perform better than those created with P1 data. The subjects that performed well on both classification experiments and with P1 data are the subjects 15, 16, and 17. Subjects 15, 16, and 21 created the best models with P2 data when comparing both classification techniques.

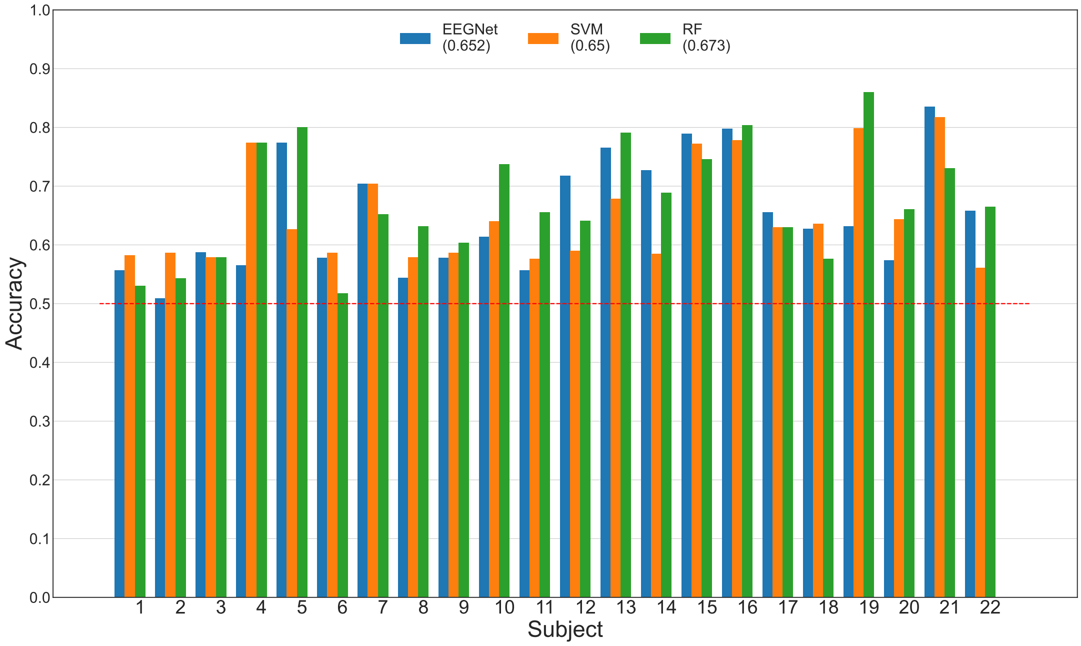


(a) Individual **resting-state vs task** classification with P1 data, dataset A.

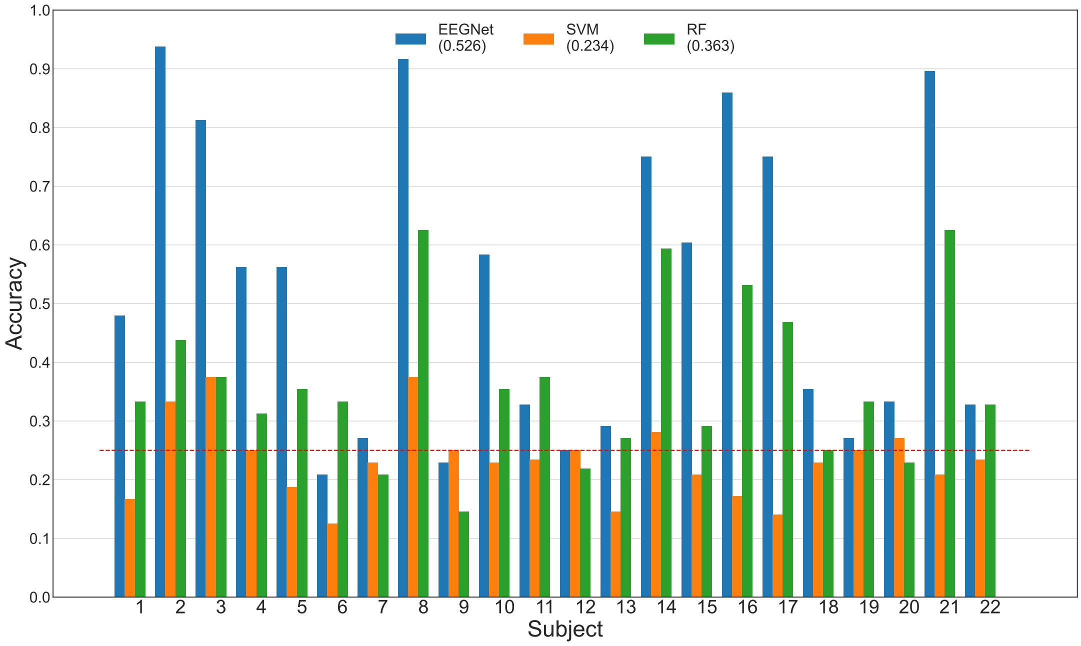


(b) Individual **4-class** classification with P1 data, dataset A.

Figure 5.1: Baseline accuracies for the individual **resting-state vs task** and **4-class** models for the experiment in section 5.2. The models were created for each subject with data from **P1**, dataset A. The plots show accuracy for the classifier EEGNet, SVM and RF classifiers. The red line illustrates the level of chance, 0.5 for the **resting-state vs task** classification and 0.25 for the **4-class** classification. The average performance overall subjects of a classifier are written in parenthesis below the legend.



(a) Individual **resting-state vs task** classification with P2 data, dataset A.



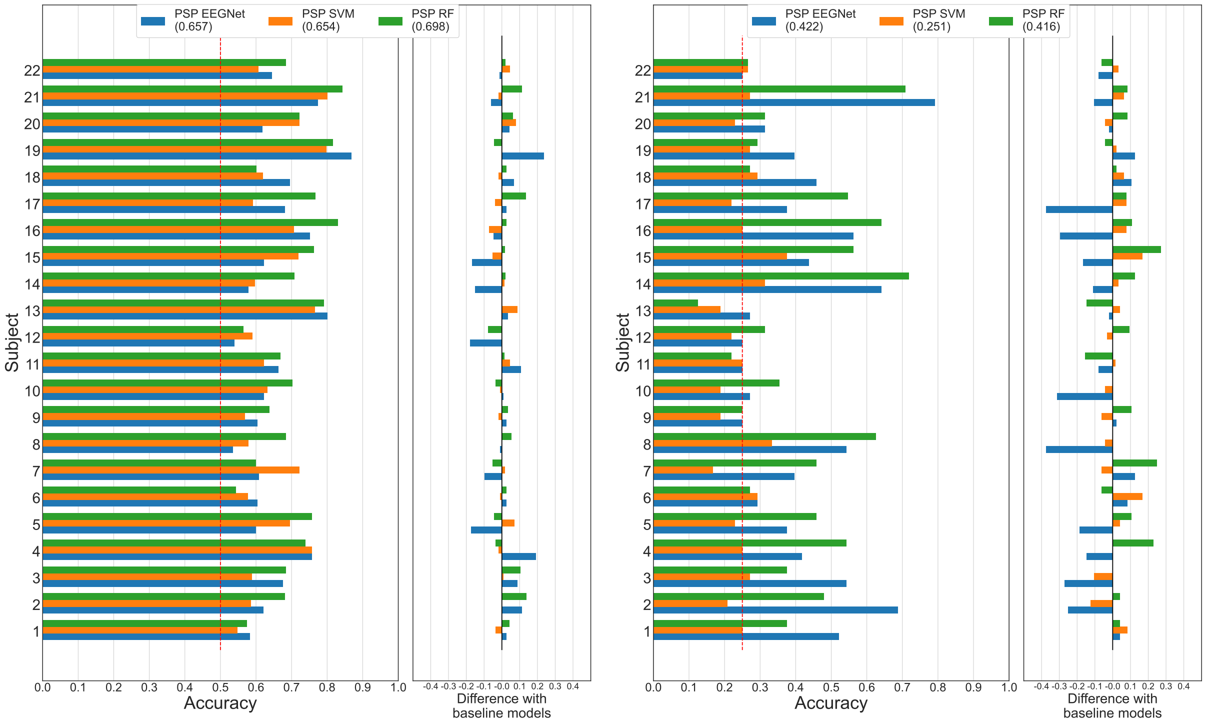
(b) Individual **4-class** classification with P2 data, dataset A.

Figure 5.2: Baseline accuracies for the individual **resting-state vs task** and **4-class** models for the experiment in section 5.2. The models were created for each subject with data from **P2**, dataset A. The plots show accuracy for the classifier EEGNet, SVM and RF classifiers. The red line illustrates the level of chance, 0.5 for the **resting-state vs task** classification and 0.25 for the **4-class** classification. The average performance overall subjects of a classifier are written in parenthesis below the legend.

5.3 Individual models trained with PSP processed data, dataset A

The classifier methods will use the PSP pipeline on the data to create their models in this experiment. The PSP pipeline is explained in section 4.4.4. The goal is to see if this processing pipeline is better than the one used in section 5.2. Only the accuracies are presented in the result figures to simplify the comparisons. Each figure consists of two plots. The left plot displays the accuracy of PSP data, while the right plot shows a comparison between the new results and the individual results in section 5.2.

Most of the same conclusions from models created with P2 data can also be drawn from models created with P1 data. So, the models made with P2 data are presented in this section, in fig. 5.3, while the results created with P1 data can be viewed in appendix B, fig. B.1. The primary difference is that the classification techniques trained with P2 data have better average performance. In addition, methods performing well on a specific subject with one of the protocols do not necessarily perform well on the other protocol.



(a) Individual **resting-state vs task** with P2 data, dataset A.

(b) Individual **4-class** classification with P2 data, dataset A.

Figure 5.3: Accuracies from creating individual EEGNet, SVM, and RF models with PSP data, from **P2**, for the experiments in section 5.3. Below the names of the methods in the legend is the average accuracy for that classification method. Each figure contains two plots, where the left one shows the accuracies created with PSP data, and the right one compares the results with the respective ones from section 5.2. The red dashed line illustrates the chance level for that classification.

The **resting-state vs task** classification results, see fig. 5.3a, show that RF is performing best on average, even though EEGNet has trained the best performing model. All classification methods manage to train models that perform above the chance level. Comparing the average accuracy for the three methods with the average in section 5.2, it is clear that using PSP data has slightly increased the accuracies. The comparison plot shows that SVM and RF difference is close to zero for nearly all subjects. In contrast, the difference for EEGNet is more divergent and has significant differences, both positive and negative, for some subjects.

Examining the results from the **4-class** classification in fig. 5.3b shows that EEGNet is performing best, though, only slightly better than the performance of RF. SVM struggles with creating models that perform better than chance. Looking at the comparison plot, it is evident that EEGNet has significantly decreased its accuracies, while SVM and RF have slightly increased theirs, RF more than SVM.

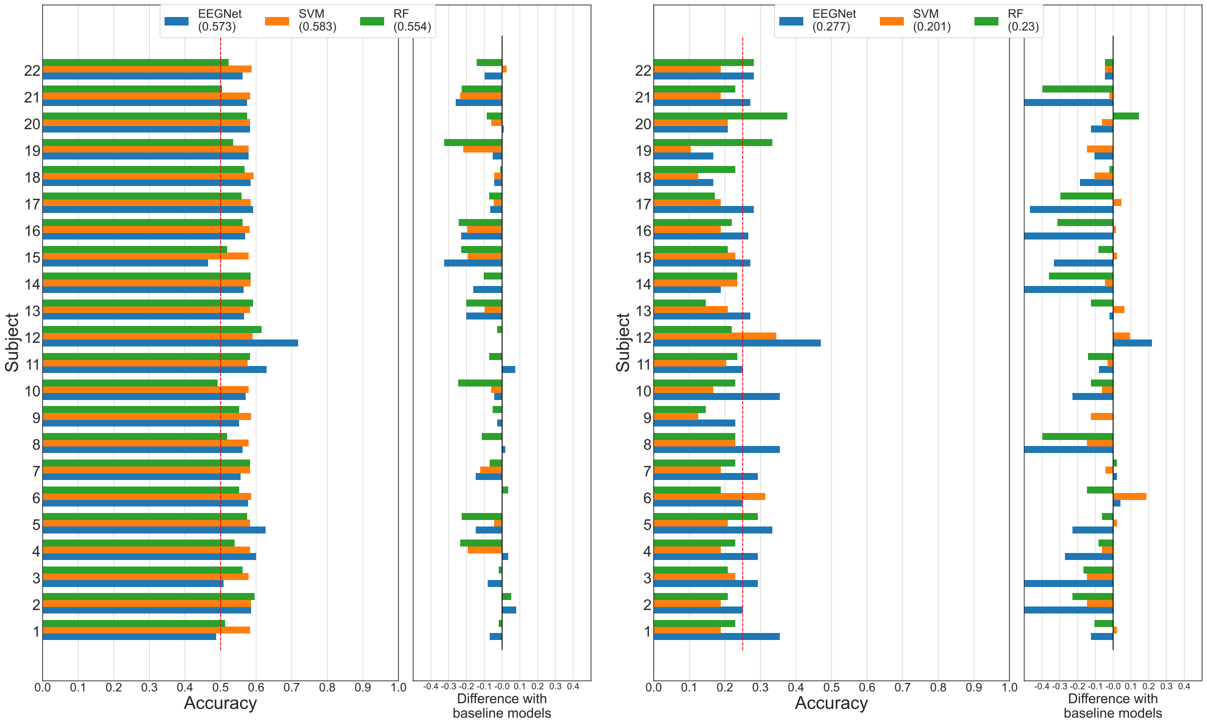
To summarize, PSP proves that it can deliver good results and, often, better than the methods used in section 5.2. However, the processing pipelines used in section 5.2 are superior, and two reasons back this hypothesis. Firstly, only pre-processing the data before EEGNet has created the best models, especially for **4-class** classification. In addition, to deliver good results consistently. Secondly, PSP is a slow technique as it has to perform several processing methods, which takes a long time. In other words, not ideal in an online environment. Implies that PSP models need to really outshine the baseline models in section 5.2, as one of the requirements for the BCI is fast online prediction. With this in mind, the following experiments in this thesis will continue to use the processing pipelines used in section 5.2.

5.4 Individual models trained on only occipital lobe data, dataset A

During the datasets recording, the participants were instructed to move their eyes and allowed to blink when the resting state was displayed. Artifacts from the eyes can be a significant noise factor in the EEG signals. Furthermore, persons with CLIS cannot move their eyes. So, only using signals from the brain's color perception in the occipital lobe when creating models can indicate the feasibility of the BCI design for CLIS patients, and if the eye movement information is necessary or just contaminants the signals with noise.

The processing methods for EEGNet are the pre-processing pipeline, explained in section 4.4.2, while SVM and RF use the feature extraction pipeline explained in section 4.4.3. The only change is that the raw input samples now contain EEG signals from six channels instead of eight, namely the six placed in the occipital lobe, *PO3*, *POz*, *PO4*, *O1*, *Oz*, and *O2*. The graphs presented in this section are read in the following way. Each figure consists of two plots; the left plot displays the accuracy of the individual models created with one of the three methods and data from the occipital lobe. The chance level is visualized as a red dashed line at 0.5 for **resting-state vs task** and 0.25 for **4-class** classification. The right plot depicts the difference in accuracy between the new individual models and the results from the baseline models created in section 5.2.

The results from this experiment are visualized in fig. 5.4. The **resting-state vs task** and **4-class** classifiers shown are created with P2 data, from dataset A. The results of models created by this experiment, and with P1 data, are very similar to the ones in fig. 5.4, and can be viewed in appendix B, fig. B.2. The accuracy is generally very low, though only three models are below the chance level of all **resting-**



(a) Individual **resting-state vs task** with P2 data, dataset A.

(b) Individual **4-class** classification with P2 data, dataset A.

Figure 5.4: Accuracies for individual **resting-state vs task** and **4-class** classification models for the experiment in section section 5.4. The figures show models created with EEGNet, SVM and RF when only **P2** data from the occipital lobe are used. The results are compared against the corresponding results in section 5.2. The average performance overall subjects of a classifier are written in parenthesis below the legend.

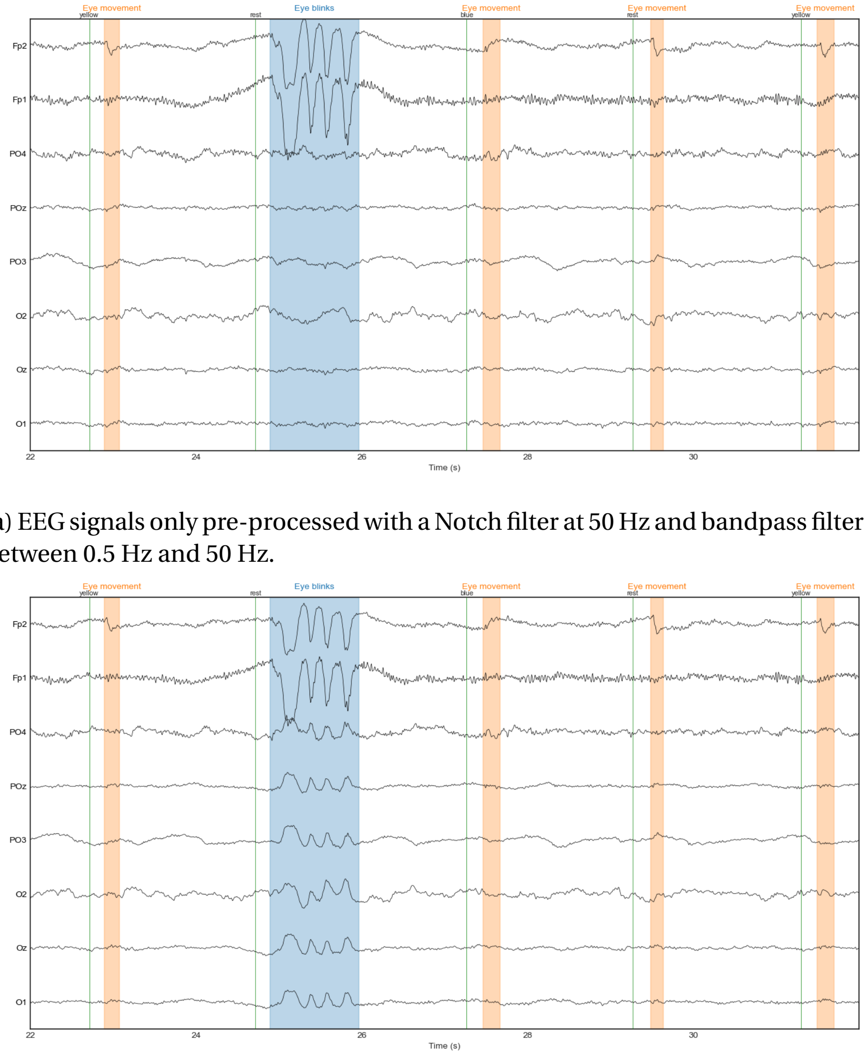
state vs task models, see fig. 5.4a. It is also evident that SVM is generally doing best, while EEGNet has the best performing models. Looking at the difference plot in the same figure, is it clear that all three methods are decreasing in performance. The few models with a positive difference, calculated from the baseline, are very sporadic, and there is no pattern between them.

For the results from the **4-class** classification, fig. 5.4b, only EEGNet had an average performance above the chance level and was the method with the highest accuracy. However, EEGNet has the most significant decrease in performance when examining the comparison plot. After EEGNet, RF is the method that has decreased the most and is the method that performs the worst of them all. SVM has the most models with positive accuracy differences. Though SVM generally has a worse performance from the baseline and struggles with performing better than chance.

EEGNet has the overall best performance when comparing fig. 5.4a and fig. 5.4b, though it is also the method that has decreased most from its baseline results. The only subject model that had a better performance than the baseline in both classification experiments was the two EEGNet models for subject 12. Looking at both difference plots, it is clear that the general performance has been significantly decreased and that the information from the two electrodes is essential for the classification. There is also no correlation that when one classification method increases a subject's performance, compared to the baseline, the other methods for the same subject will also perform better.

5.5 Individual models pre-processed without CAR, dataset A

EEG signals generated from neurons in the brain have small values. In contrast, artifacts generated from muscle contraction are significantly higher, causing more electrodes in a broader vicinity to record the noise. CAR removes the same noise picked up by several electrodes by subtracting the common average from each electrode at all times, see section 2.5. However, this can be problematic. The method can add noise to other electrodes that do not contain that noise from the beginning, implying that the pre-processed signal is noisier than the raw one. In fig. 5.5, this problem is illustrated.



(a) EEG signals only pre-processed with a Notch filter at 50 Hz and bandpass filtered between 0.5 Hz and 50 Hz.

(b) EEG signals pre-processed as explained in section 4.4.2.

Figure 5.5: A visualization of the CAR problem, explained in section 5.5. The plots show EEG signals from two different pre-processing pipelines. The signals are from subject 8, session 3, dataset A, and the x-axis shows the time in the recording and the y-axis the different channels. The green lines show the events displayed to the participant. Orange indicates eye movement, while blue indicates eye blinks.

In fig. 5.5a are the raw signals pre-processed with only a Notch filter at 50 Hz and a bandpass filter between 0.5 Hz and 50 Hz. The eye blinks that the participant performed around 25 seconds in the recording are only visible in the two front electrodes, *Fp1* and *Fp2*. The eye movement is primarily present in *Fp2* but is also visible in *Fp1* and *PO3*. The eye blinks contaminants the first two electrodes on a larger scale than the eye movements. In fig. 5.5b, the usual pre-processing method, explained in section 4.4.2, is used. It is noticeable that CAR pollutes the electrodes on the occipital lobe with noise from blinks and smooths the eye movement from the channels *Fp1* and *Fp2*. So, creating models where CAR is removed from the pre-process pipeline can show if the added noise from CAR is a problem for the classification performance.

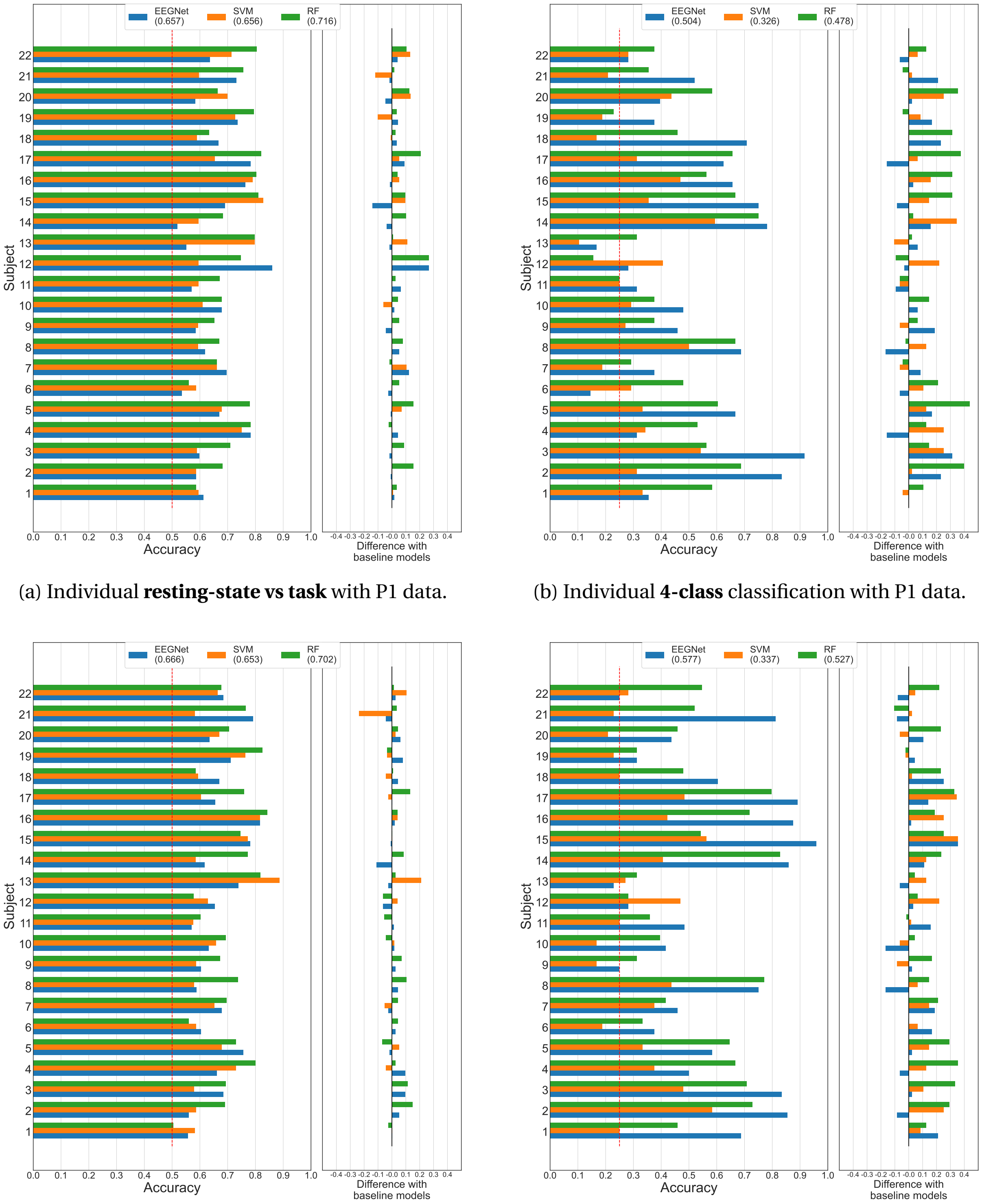
The figures presenting the results in this section are read equal to the ones in section 5.4. The comparison plot is between the new model and the baseline calculated in section 5.2. All models are created with data from dataset A, and the methods used are EEGNet, SVM, and RF. They use the processing line explained in section 4.4.2 and section 4.4.3. The only difference is that CAR is removed from the pre-processing pipeline.

The results for all individual models can be viewed in fig. 5.6. The models created with P1 data are presented in fig. 5.6a and fig. 5.6b. For the **resting-state vs task** classification, see fig. 5.6a, most models are improved by removing CAR from the processing line. RF performs best on average and is the one that has the most and best improvements compared to the baseline models. SVM and RF have an equally good average performance. In the results from the **4-class** classification, see fig. 5.6b, EEGNet is performing best on average, though RF is trying to compete. The best accuracies are computed with EEGNet and are above 0.9 for subject 3, which is a massive improvement from the baseline. In the comparison plot is the RF classifier, the one with the most improvements, followed by SVM. EEGNet has most models with a negative difference from the baseline, though the method has a generally positive performance.

Results from models created from P2 data are presented in fig. 5.6c and fig. 5.6d. From the **resting-state vs task** results in fig. 5.6c, RF is the method that is performing best on average. Additionally, this method has resulted in the highest accuracy. Looking at the differences from the baseline, SVM and RF have the best improvements by neglecting CAR. Additionally, the difference with the baseline models is generally a slight improvement for all three methods. EEGNet performs best in the **4-class** classification, see fig. 5.6d, and has several methods with around an accuracy of 0.9. RF also performs well on average, but examining the performances of its models, it is evident that it never reaches equal high accuracies as EEGNet. However, EEGNet is the method that most frequently has a negative difference from the baseline. Whereas SVM and RF often have a massive improvement from the baseline.

To summarize, SVM and RF were the methods that got the best performance increase by this change. EEGNet generally performed better with this change. However, for some subject models, the model performance decreased significantly. The **4-class** classification models had the most significant improvements, and many models improved their accuracy by 0.3 to 0.4 from the respective baseline models. There is little to no difference in improvements between models created with P1 data and

P2 data. In other words, results from both protocols were enhanced by removing CAR from the preproce-ssing pipeline. This experiment concludes that CAR pollutes the sample with noise and reduces the classification performance. So, *all further experiments are conducted with CAR removed from the pre-processing pipeline.* As these new individual models are an improvement from the baseline models in section 5.2, they will be used when calculating the performance difference in the following experiments.



(a) Individual **resting-state vs task** with P1 data.

(b) Individual **4-class** classification with P1 data.

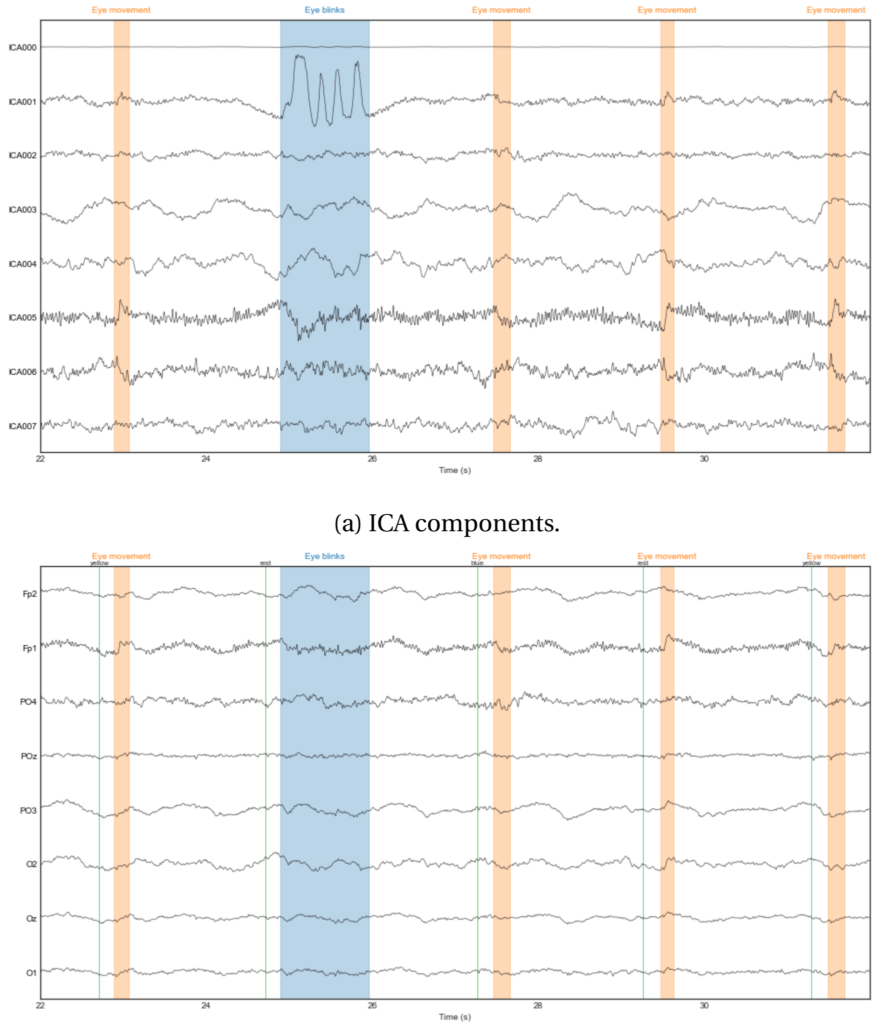
(c) Individual **resting-state vs task** with P2 data.

(d) Individual **4-class** classification with P2 data.

Figure 5.6: Accuracies from the experiment in section 5.5 created by models on dataset A where CAR is removed from the pre-processing pipeline. The results are compared against the corresponding results in section 5.2. The average performance overall subjects of a classifier are written in parenthesis below the legend.

5.6 ICA noise reduction on a selection of subjects, dataset A

From section 5.5 was it clear that eye blinks are a significant noise factor in the signal and have much larger values than standard EEG signals. The objective of this experiment is to see if the removal of the eye blinks can enhance the performance of the signal. The blinks will be removed manually with ICA so that most of the information is still kept in the data. The experiments will only be conducted on five subjects and only P2 data from dataset A. The eye blinks will, in other words, be removed from three sessions per subject, as removing eye blinks is time-consuming. In addition, CAR is removed from the pre-processing pipeline.



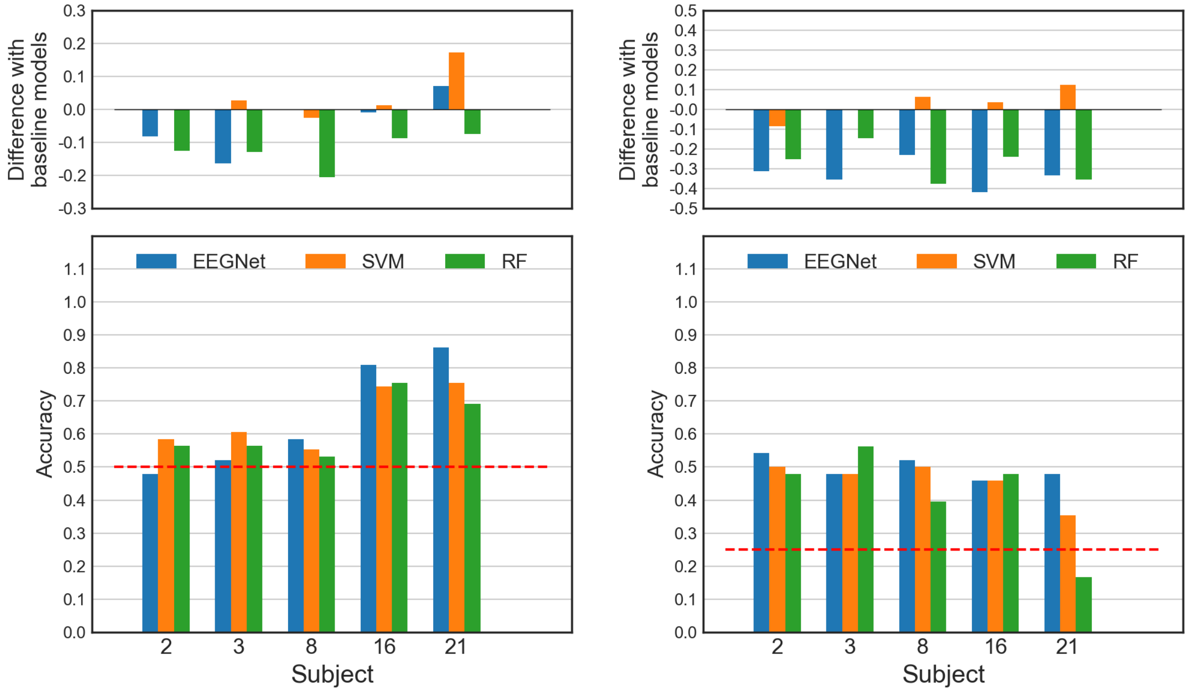
(a) ICA components.

(b) Reconstructed EEG signal after removing blinks with ICA.

Figure 5.7: Visualization of the EEG signals when removing blinks with ICA for the experiments in section 5.6. The signals are from subject 8, session 3, dataset A, and are the same time interval as used in fig. 5.5. The x-axis shows the time in the recording, y-axis the different channels. The green lines mark the events displayed to the participant. Orange indicates eye movement, while blue indicates eye blinks.

The process of removing eye blinks is visualized in fig. 5.7, which is from the same subject, session, and time interval used in fig. 5.5. The ICA components are visualized in fig. 5.7a, and it is apparent that component 1 is the one that contains information about hte blinks. It is also evident that eye movement is part of this component. By removing this component, some information about eye movement will be lost, even though components 5 and 6 contain information about eye movement. The reconstructed signal after removing component 1 is displayed in fig. 5.7b, and the blink around 25 seconds is now removed. The information on eye movement is now most distinct in *Fp1*.

Figure 5.8 presents the results from the models created with data where the blinks are manually removed. The results are compared with the respective ones in section 5.5, by calculating the accuracy difference between them. The **resting-state vs task** classification results, see fig. 5.8a, show only improvement on one subject, subject 21, with the SVM model. EEGNet does not decrease much in performance, and the model for subject 21 is closing in on accuracy of 0.9. For the results from the **4-class** classification models, see fig. 5.8b, SVM is still the one with improvements though they are insignificant. The deterioration of the performance of both EEGNet and RF in fig. 5.8b is significant and gives a clear picture that by removing blinks, too much information is lost.



(a) Five individual **Resting-state vs task** models with P2 data, dataset A.

(b) Five individual **4-class** classification models with P2 data, dataset A.

Figure 5.8: Accuracies from the individual **resting-state vs task** and **4-class** classification models, experiment in section 5.6, created with **P2** data from dataset A. The blinks in the raw signals were removed with ICA. The results are compared against the corresponding results in section 5.5. Note the difference in the y-axis between the two comparison plots.

5.7 Individual models while comparing DWT and VMD with different features, dataset A

Until now, DWT has been used to decompose an epoch into components, where features are extracted from the components. VMD, with its possibility to decompose a signal to modes, can be used instead of DWT. In addition, more features can be extracted from the components, such as statistical features explained in section 2.7.4. This experiment aims to see if modes from VMD result in a better performance of SVM and RF and looking for a performance boost by extracting statistical features in addition to energies and fractals from the components.

The processing pipeline for a sample used in this experiment is displayed in fig. 5.9. A sample is preprocessed with a Notch and bandpass filter before decomposing with DWT or VMD. Five modes are extracted from each channel in the pre-processed signal, decided after experimenting with different amounts of modes and looking at how similar or noisy they were. From the components were either only the fractals and energies extracted or also statistical features extracted.

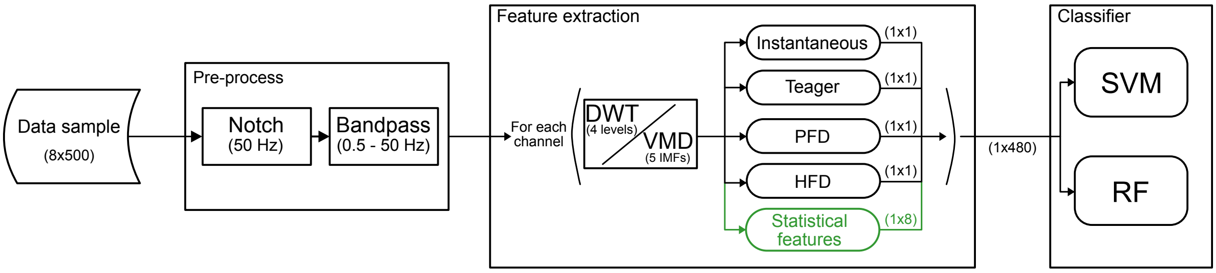


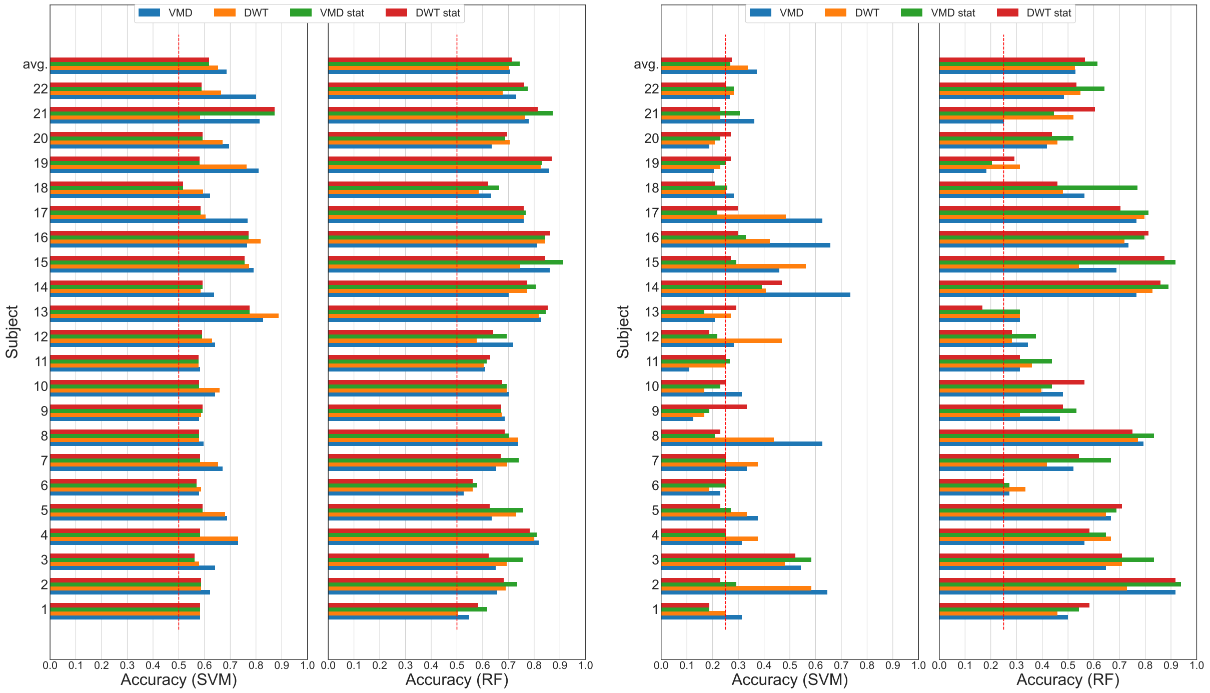
Figure 5.9: Visualization of the processing pipeline of a sample used in the experiments section 5.7

The result figures consist of two plots; on the left side are the accuracies from SVM models, while on the right side are the accuracies created with RF presented. The top bar graph in all plots displays the average accuracy for the methods, calculated with the related accuracies from all subject models. When a legend has written "stat" after the method, the statistical features were also extracted from the components. If not, only fractals and energies were extracted. The results created with P1 and P2 data from dataset A are very similar. The main difference between them is that P2 models generally perform better. A good P1 model for a random subject does not necessarily produce equally good on the P2 dataset and vice versa. So, only the results from P2 data are displayed in this section, fig. 5.10, while the results created with P1 data can be viewed in appendix B, fig. B.3.

From the accuracies created with SVM doing **resting-state vs task** classification, fig. 5.10a, it is clear that either DWT or VMD is the best processing method even though VMD stat got the best accuracy for subject 21. The methods DWT and VMD are usually the only contenders for getting the highest accuracy, and VMD wins the most. VMD still has the highest average accuracy for SVM in the **4-class** classification in fig. 5.10b, followed by DWT. However, VMD also has both the highest and lowest accuracies. In other words, the decomposition technique with the highest variance. In addition, VMD is very slow, compared to DWT, to decompose a signal as it is a concurrent optimization technique. Since DWT is performing consistently well and is a faster decomposition method, it will continue to be used when creating SVM models.

The results created with RF from **resting-state vs task** show that the VMD stat method is best on average by a small margin, see fig. 5.10a. It manages to get an accuracy score above 0.9, and most subjects above accuracy of 0.7. From RF's results in the **4-class** classification, fig. 5.10b, VMD stat is still the best method, only contested by DWT stat. Both methods generally perform well and manage to create three models with accuracies around 0.9. VMD stat has the edge on DWT stat in both classification tests, implying that it will continue to be used when creating RF models.

To summarize, RF processed with VMD stat is the method that performs best on an individual subject model on average. The method has an average accuracy of 74.3% for **resting-state vs task** classification and 61.4% for **4-class** classification. The method is also best when compared against the best average results for EEGNet



(a) Individual **Resting-state vs task** models, created with P2 data, dataset A.

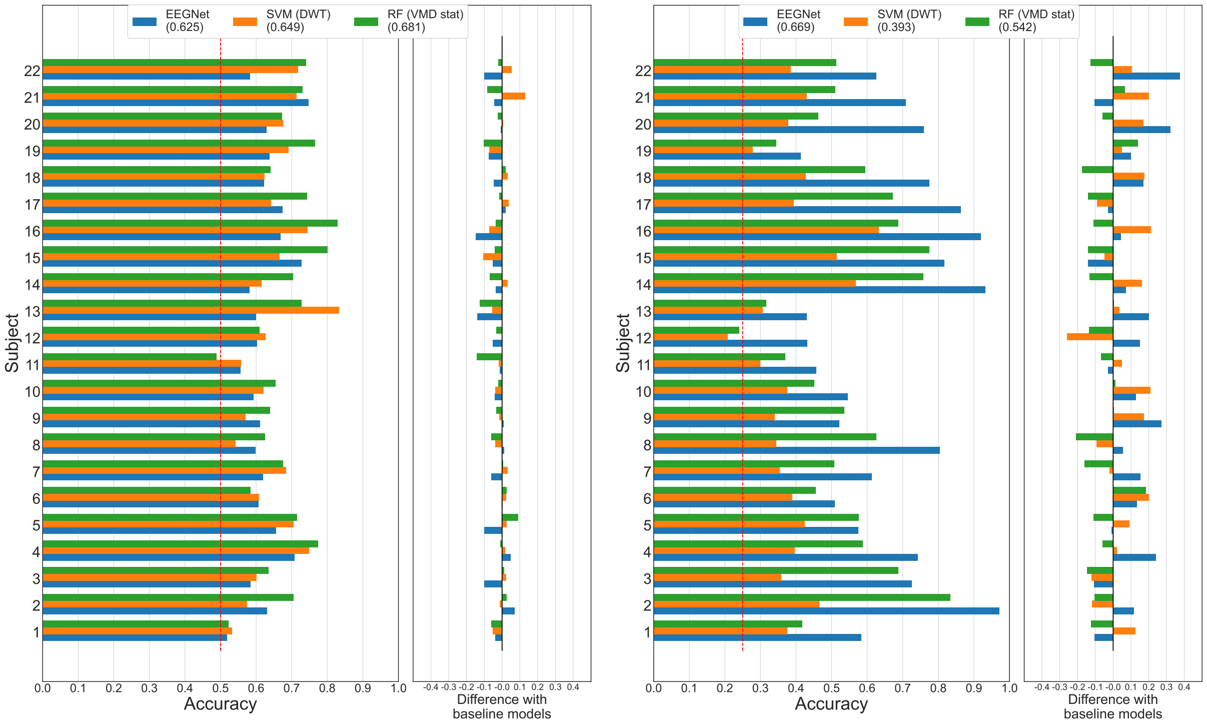
(b) Individual **4-class** classification models, created with P2 data, dataset A.

Figure 5.10: Accuracies from the individual **resting-state vs task** and **4-class** classification models, from the experiment in section 5.7. The models are created with P2 data from dataset A. The left side plot displays the accuracies from SVM models, while the right side plot displays the accuracies created with RF presented. The legends show what processing technique was used to create the results, and the top bar graph in all plots displays the average accuracy for the methods.

5.8 Generalized individual leave one out models, dataset A

The previous experiments have helped decide which methods can produce the best results. This experiment will test creating a more generalizable model with the best methods found with data from several subjects. The main problem with creating a general model is testing its performance, as this should be done with new data from unknown subjects. This problem was overcome, in this thesis, by creating an individual model for every subject. The models were created with data from all subjects except the subject data used for testing the model. For example, the models for subject 1 were created with data from the remaining 21 subjects, while subject 1's data was used to test the model. These kinds of models are called Leave one out model (LOOM) in this thesis. Even though a new model has to be trained for each subject, the advantage of getting a less biased result is essential when examining the feasibility of a more general model.

All data used in the model have the same pre-processing, only notch and a bandpass filtering. EEGNet uses the pre-processed data as input, and SVM uses the fractals and energies extracted from the components from DWT. In contrast, RF uses the energies, fractals, and statistical features extracted from the modes created by VMD. The model's performance is compared to their best individual models, EEGNet and SVM, to their results in section 5.5, while RF to its VMD stat results in section 5.7. A note to the comparison, the new models use all data from the respective subject to test the model, while the old models only used 20% of the data from the subject to calculate the accuracy.



(a) Individual **resting-state vs task** classification with P2 data, dataset A.

(b) Individual **4-class** classification with P2 data, dataset A.

Figure 5.11: Results from creating LOOMs with dataset A and **P2** data for the experiments in section 5.8. The methods used are displayed in the legend, with their average accuracy calculated from all subjects written in parenthesis below the method name. Each figure includes two plots, where the left one is the accuracies created with LOOM and the right one compares the results with the previous best for that method. EEGNet and SVM are compared to their counterparts in section 5.5, while RF is compared to the models created with VMD stat in section 5.7.

The result of the experiments are presented in fig. 5.11, found in this section, and shows results created with P2 data, and fig. B.4, located in appendix B, whose results are created with P1 data. The reason for only showcasing results created with P2 data is that the essential conclusion, drawn from the results, are similar in both experiments. The primary difference between the results is that **resting-state vs task** classification is slightly better on P1 data, and EEGNet has more frequently increased its performance on **4-class** classification with P1 data.

Examining the results in fig. 5.11, showcase the possibility of using a general model to classify data from a different new person. In the **resting-state vs task** results, see fig. 5.11a, all models, except one created with RF, have accuracies above the chance level. The highest accuracies are created with SVM and RF, and RF is, on average, performing best. Most models perform worse compared to models only created with data from the tested subject. SVM has most models with a positive difference, though not so many that this method is better than using models created with only data from the tested subject.

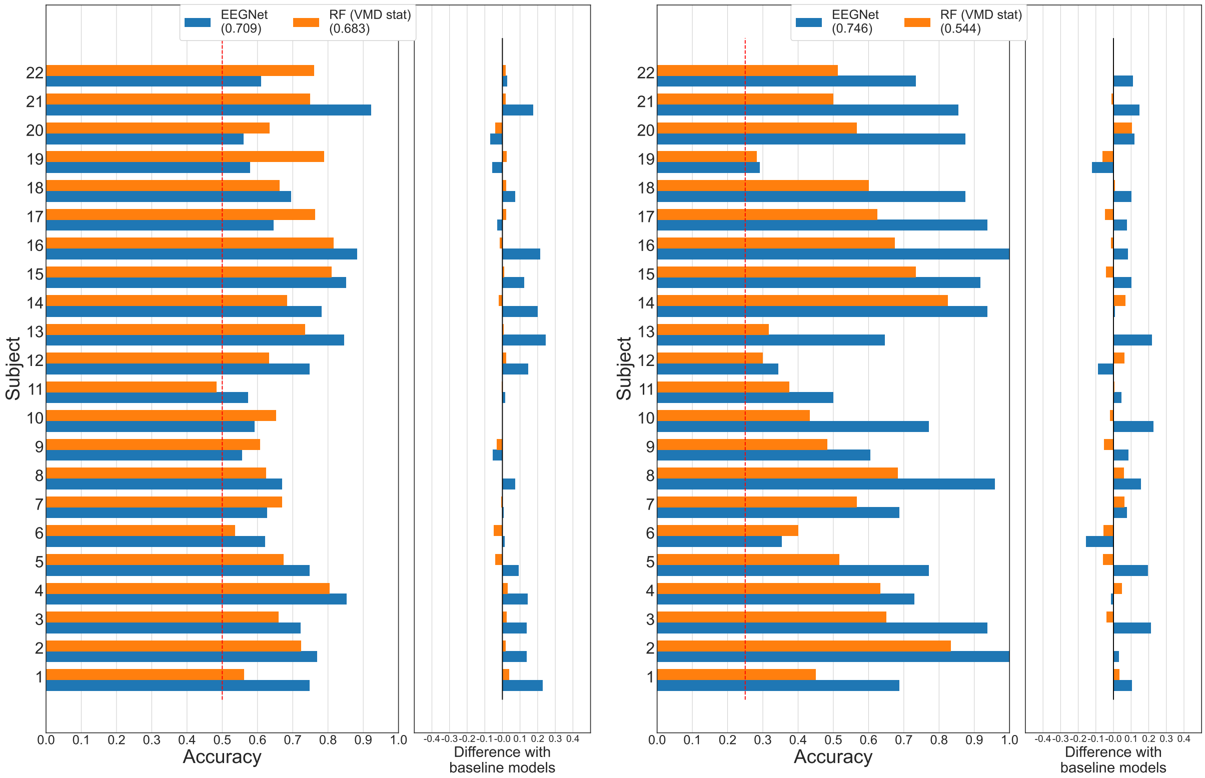
The results from the **4-class** classification indicate that EEGNet is both performing best on average and creates the highest accuracies, see fig. 5.11b. RF is also performing generally well, though it does not manage to create the same high accuracies as EEGNet. SVM is always performing worse than either EEGNet or RF. Comparing these results to the baseline, EEGNet and SVM are mainly improving from models created with only data from the tested subject, while RF primarily has decreased its accuracy.

5.9 Generalized pre-trained individual leave one out models, dataset A

This experiment aims to improve the results from the EEGNet and RF models in section 5.8. The models created in section 5.8 are used as a *base* for the new models. Data from the subject that the models test on are split into three datasets, train, validation, and test set, and used to train the base models further. For EEGNet, this is done with transfer learning, while for RF, it is done with a warm start, see section 4.6. All data used in this experiment is preprocessed as explained in section 5.8.

The results from models created with P1 and P2 are very similar. The main difference is that models created with P2 data perform slightly better. As the conclusion drawn from both protocols are similar, the results for P2 are displayed in this section, see fig. 5.12, while the results created with P1 data can be viewed in appendix B, fig. B.5. A result figure contains two plots, where the left plot presents the accuracies created with the new models, and the right plot presents the difference with the respective models from section 5.8. Note that since the data from the subject being tested is being used to train the models, only 20% of the subject data is used to test the new models.

By examining the results in fig. 5.12, it is clear that the performance increase when training on a pre-trained model. The **resting-state vs task** classification results, see fig. 5.12a, illustrate that EEGNet is the best performing model on average and achieves accuracy above 0.9. RF only manages to get accuracies above 0.8 and is generally outperformed by EEGNet, though it manages to create a much better model for some subjects. See the results for subjects 19 and 22. By examining the comparison plot, it is evident that mainly EEGNet improved by using transfer learning, while a warm start to RF



(a) Individual **resting-state vs task** classification with P2 data, dataset A.

(b) Individual **4-class** classification with P2 data, dataset A.

Figure 5.12: Accuracies from using transfer learning and warm start on the EEGNet and RF models from section 5.8, for the experiments in section 5.9. The pre-trained LOOMs and the new models are trained with data from **P2**, dataset A. The methods used are displayed in the legend, with their average accuracy calculated from all subjects written in parenthesis below the method name. Each figure includes two plots, where the left one is the accuracies created with the new LOOM and the right one compares the results with the respective ones from section 5.8.

only slightly increased its performance.

EEGNet also outperforms RF in the **4-class** classification; see fig. 5.12b. Both methods struggle with keeping the performance of their model above the chance line for some subjects, namely subjects 6, 12, and 19. Though EEGNet compensates by predicting all samples correctly for subjects 2 and 16. The comparison plot shows that both methods have improved by using pre-trained models as the base before further training. However, EEGNet is still the one that has gained the most.

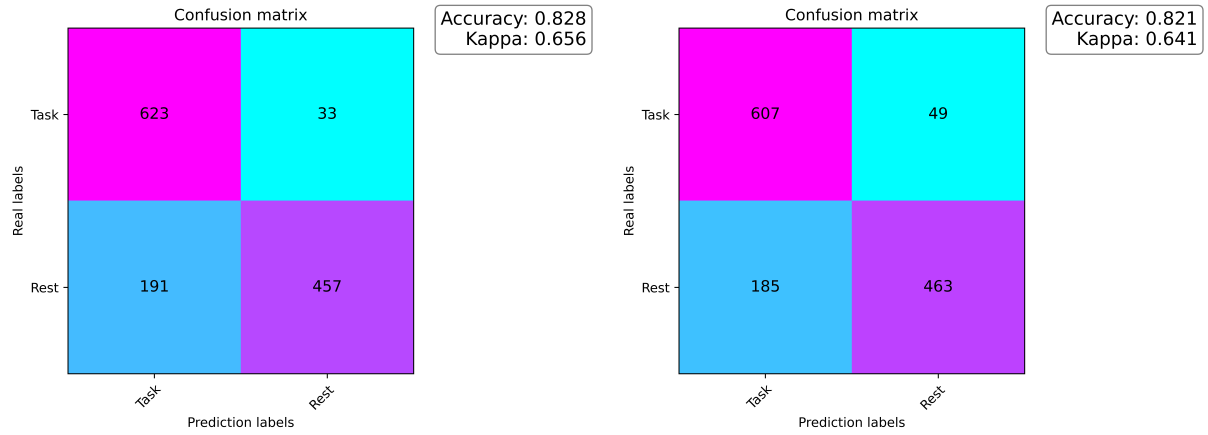
5.10 General model created with dataset B tested on dataset A

This experiment aims to create a general model with data from dataset B that can perform well on the data from different subjects in dataset A. Training and testing on two different datasets shows the strength of the classification methods. The test is now more similar to the BCI's real-life use as the data is from other subjects and recorded at a different time. Only EEGNet and RF are used for this experiment, as these two have delivered the best results in the previous experiments. In addition, it is possible to train further a model created with either of these two classifiers, which is not the case for

SVM. The processing pipeline used on dataset B is the same and was found to be the best in the previous experiments. In other words, only pre-processing data, without CAR, before using it in EEGNet, and decomposing the samples with VMD before the fractals, energies, and statistical features were extracted from the modes, where the latter is used in the RF classifier.

The general model was created with data from a selection of subjects. It is important to note that only P1 data were used to create the model because of the problems with the P2 data, as described in section 4.3.2. NSGA-II was used to discover the best performance of different subject selections. The data from the selected subjects were split into three sets, training, validation, and testing set, where the results from predictions of the testing set were used to compare the models. NSGA-II was also constrained not to create any model with date from fewer than a selection of five subjects, as this would reduce its generalizability.

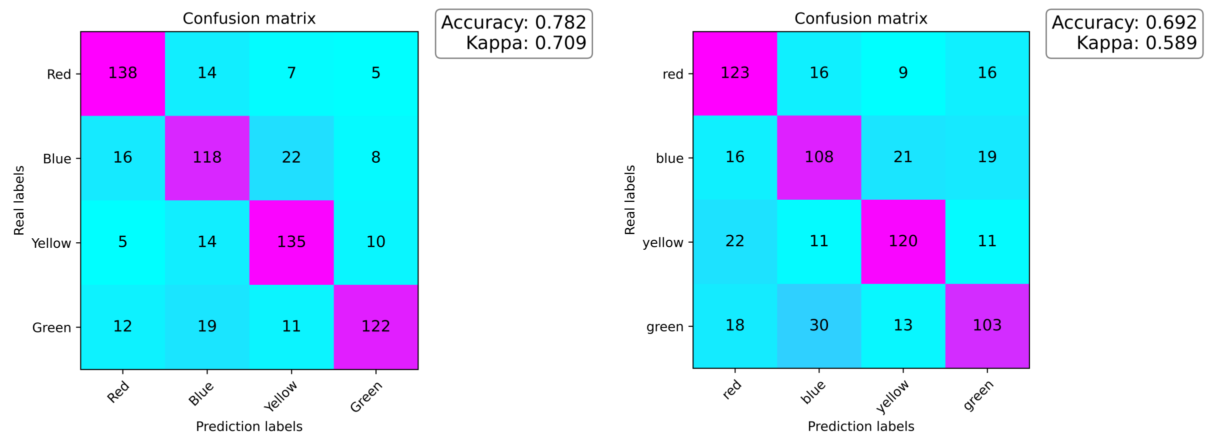
The optimization method found selecting the sixteen subjects, 7, 8, 9, 13, 14, 15, 16, 19, 22, 23, 24, 25, 28, 30, 33, and 34 to create the best model. Only 9 of these subjects are part of dataset A as well. See table A.2 in appendix A for the conversion subject conversion between the two datasets. This selection created four models: a **resting-state vs task** and a **4-class** model for EEGNet and RF. The performance of the **resting-state vs task** models is displayed in two different confusion matrices in fig. 5.13. Both classification methods manage to get accuracies above 0.8. EEGNet, fig. 5.13a, is slightly better than RF, fig. 5.13b. In addition, both models prefer to predict the task state rather than the resting state. The results from the **4-class** classification models are presented in the confusion matrices in fig. 5.14. It is evident that the EEGNet model, fig. 5.14a, outperforms the RF model, fig. 5.14b, as the difference in accuracy between them is close to 0.1. The EEGNet model has the most problems predicting green and blue, while the RF model struggles with predicting red and green.



(a) Confusion matrix for EEGNets **resting-state vs task** model created with P1 data, dataset B.

(b) Confusion matrix for RFs **resting-state vs task** model created with P1 data, dataset B.

Figure 5.13: The confusion matrices for the general **resting-state vs task** EEGNet and RF model used in the experiment in section 5.10. The two models are created with P1 data from dataset B and the subjects 8, 9, 13, 14, 15, 16, 19, 22, 23, 24, 25, 28, 30, 33, and 34. The accuracy and kappa score of the model are presented in a box to the right of the matrix.



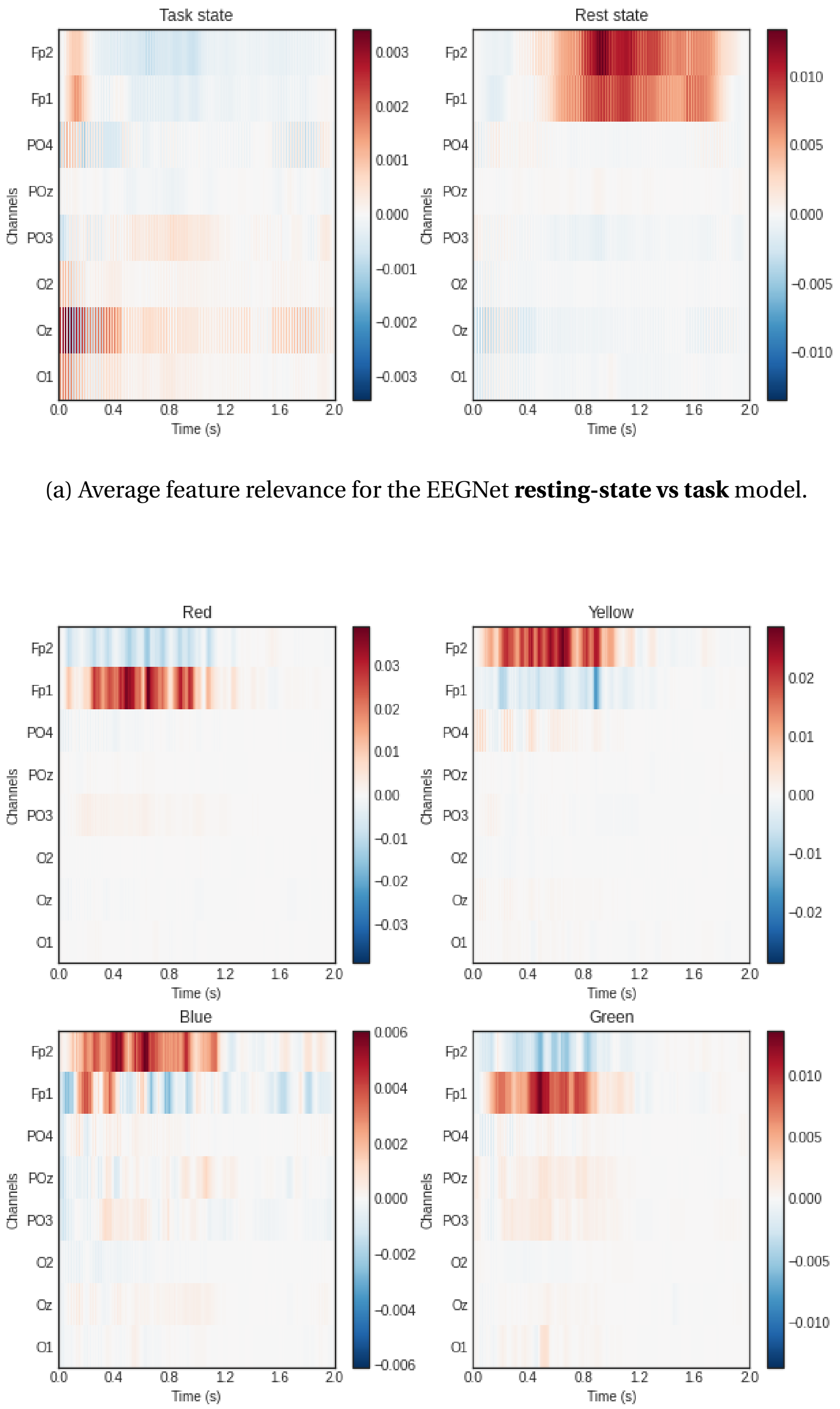
(a) Confusion matrix for EEGNets **4-class** classification model created with P1 data, dataset B.

(b) Confusion matrix for RFs **4-class** classification model created with P1 data, dataset B.

Figure 5.14: The confusion matrices for the general **4-class** classification EEGNet and RF model used in the experiment in section 5.10. The two models are created with P1 data from dataset B and the subjects 8, 9, 13, 14, 15, 16, 19, 22, 23, 24, 25, 28, 30, 33, and 34. The accuracy and kappa score of the model are presented in a box to the right of the matrix.

To get a better understanding of how EEGNet makes predictions, DeepLIFT, explained in section 4.7, was used. The dataset used to make the confusion matrices in fig. 5.13 and fig. 5.14 were used as input in DeepLIFT. The feature relevance plots, fig. 5.15, are viewed as follows. The y-axis contains the channels, while the x-axis is the sample time, zero to two seconds. A feature relevance plot explains, in this case, which periods in what channels support the outcome. So, a positive value, colored red, support the outcome, while a negative value, colored blue, denotes evidence against that outcome. Only correct predictions are used to create the plots, and a plot shows the average of all the correctly predicted samples. The plots show a more general insight into how the model predicts by showing an average of the relevance by calculating the relevance.

In the relevance plot for EEGNets **resting vs task** model, fig. 5.15a, the model uses information from all channels when predicting a *task state*. Though it primarily uses information from the two front electrodes and *Oz*. It is also evident that it looks for eye movement at the sample's start in the two front electrodes. When the model predicts a *resting state*, it mainly uses information from the two front electrodes, especially from the sample's second half. The relevance plot for the **4-class** classification model, fig. 5.15b, also shows that the information from the two front electrodes is the most relevant. It is also evident that the relevant front electrode is equal for states opposite in a diagonal direction. However, the relevant period is equal for all states. A final observation from the plot is that *PO3*, *POz*, and *PO4* are the electrodes in the occipital lobe whose information is most relevant.

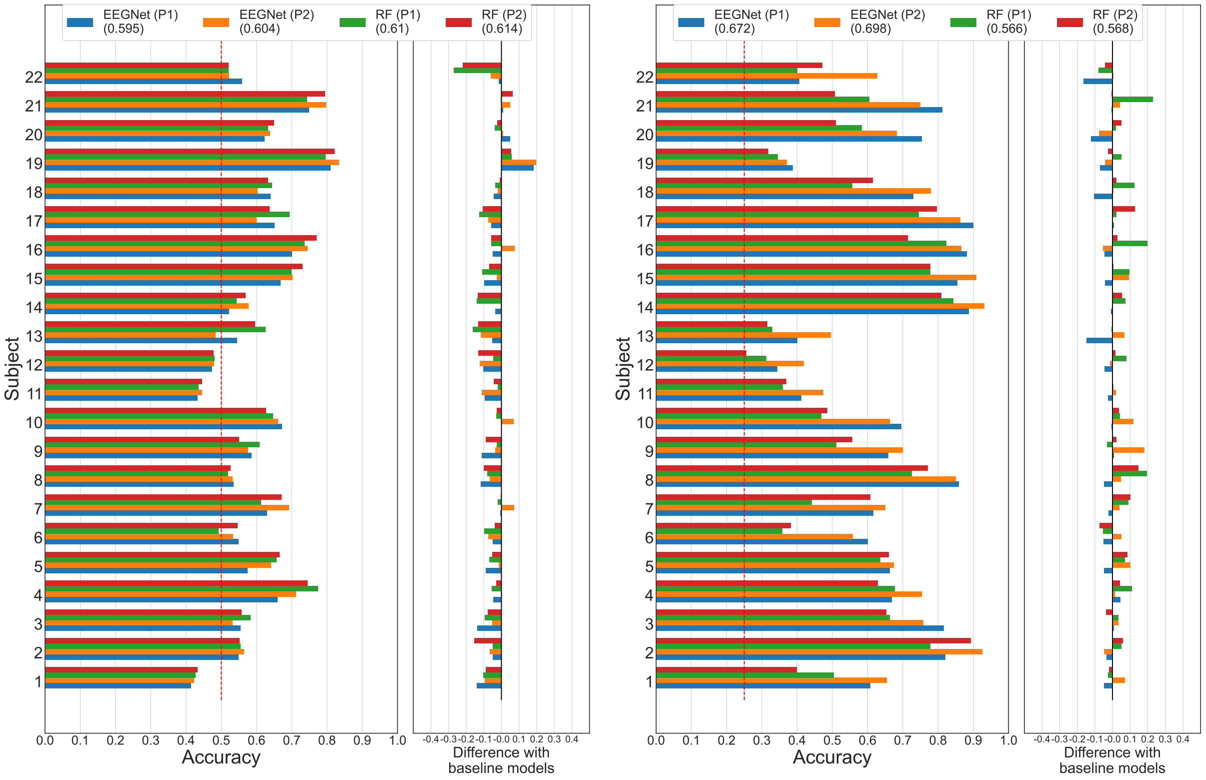


(a) Average feature relevance for the EEGNet **resting-state vs task** model.

(b) Average feature relevance for the EEGNet **4-class** classification model.

Figure 5.15: The two figures presented show the feature relevance of the general EEGNet models created in section 5.10. The figures show the channels in different periods, from an average of all samples used in the testing set to the models. The plots are created with DeepLIFT, and only correctly predicted samples were included in the plots. The y-axis shows the channel name, while the x-axis shows the time in seconds. Positive values, colored red, support the predictions, while negative values, colored blue, oppose the predictions.

The four models were finally tested on the subject data from dataset A. Subject data from both P1 and P2 were tested, even though the general models were only trained on P1 data. The accuracies for each subject are presented in fig. 5.16 and compared to the respective results in section 5.8. From the **resting-state vs task** classification results, fig. 5.16a, it is clear that the models had problems with predicting better than the chance level. However, for some subjects, the models manage to get accuracy to around 0.8. The RF model generally performs better, though EEGNet has the highest accuracy. Testing the models with P2 data is slightly better than testing with P1 data, even though the models have been trained with P1 data. In the comparison plot, it is evident that the model created on dataset B performs worse than creating models with data from dataset A.



(a) Individual subject **resting-state vs task** predictions with P1 and P2 data, dataset A.

(b) Individual subject **4-class** predictions with P1 and P2 data, dataset A.

Figure 5.16: The two figures show the accuracy scores when using the general models created in section 5.10 with dataset B to predict the P1 and P2 data from all subjects in dataset A. Model and protocol data used are displayed in the legend. Each figure includes two plots, where the left one shows the accuracies created with the general models and the right one compares the results with the respective ones from section 5.8. The red dashed line indicates the chance level for the two classification problems.

All four **4-class** classification models performed better than the chance level for all subjects, see fig. 5.16b, though some accuracies were very close to the level. The EEGNet model outperforms the RF model on every subject. The results from the P2 data are again also better than the predictions from the P1 data. The two models created with dataset B also outperform the general models created on dataset A, as seen in the comparison plot. For some subjects, such as subjects 9 and 21, the prediction accuracy increased significantly. The only decrease in performance is happening more often when predicting P1 subject data with the EEGNet model.

A final observation from the results in fig. 5.16 is that the models managed to predict very well with data from the participants who had never performed an EEG recording before. As mentioned in section 4.3.2, these subjects have the numbers 17, 18, and 19. Furthermore, subject 12's performance was very low, considering that this subject participated in both experiments, and data from the same subject in dataset B were also used to make the general models. However, this is not the case for subject 2, whose data from dataset B were used to create the models, and when tested on its data from dataset A, it got the highest **4-class** classification accuracy.

5.11 General dataset B model used as base model for dataset A

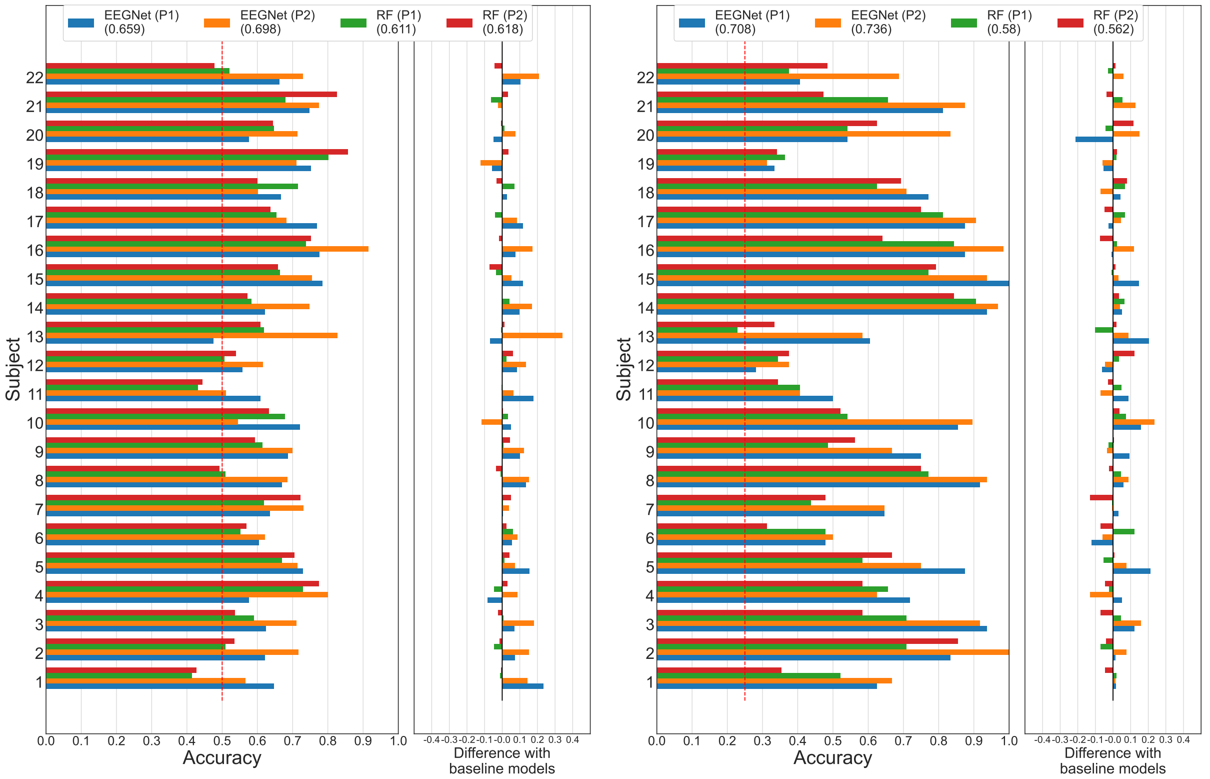
The goal of the final experiment was to improve the accuracy of each subject's data predicted with the four general models created in section 5.10. The same method used in section 5.9 was utilized for this experiment. In other words, further training the general models created in section 5.10 with a selection of the subjects' data, data from the subject being tested, on the base model. So, this experiment will test if it is possible to use only one general model to improve the accuracy.

The experiment pipeline started with loading the four general models from section 5.10 before iterating through every subject in dataset A. For every subject, its data were split into three, where the training and validation set was used for training, and the testing set was used to test the final model. The EEGNet models used transfer learning to continue their training, while the RF models used a warm start.

The accuracies for each subject are presented in fig. 5.17, where they are also compared against the results from section 5.10. It is evident in the results for **resting-state vs task** classification, fig. 5.17a, that EEGNet is the best classification method. EEGNet performs nearly 0.1 better than RF. RF managed to outperform EEGNet on data from the two subjects, subjects 19 and 21. However, the difference between the two classification methods is insignificant for those subjects. The average accuracy for EEGNet created with P2 is just below the average accuracy for the best method in section 5.9. Using P2 data to train and test the model generally leads to better models than P1 data. In the comparison plot, it is apparent that transfer learning helped EEGNet with creating better models, and a warm start was not successful for RF.

Further analysis of the **resting-state vs task** EEGNet models is done with ROC and its AUC, see section 4.7 for the theory. The results created with both P1 and P2 data are plotted for each subject, where fig. 5.18a shows the ROC and AUC for P1 data while fig. 5.18b does the same for P2 data. Only five subjects, the subjects 4, 8, 10, 15, and 16, were plotted in fig. 5.18, as more subjects in a plot would make the figure more challenging to read. The subjects were chosen because they either had the best performance, had an equal performance with data from both protocols, or where a significant difference between the results from the two protocols. Subject 16 had the highest accuracy with P2 data and the best AUC score in both ROC plots, even though its accuracy on P1 data was lower than the accuracy for subject 15. Subject 15 had a nearly equal accuracy score between data from the two protocols. However, when comparing fig. 5.18a with fig. 5.18b, it is evident that the model for subject 15 better differentiates the two classes with P2 data than with P1 data. So, a final note is that the models manage to differentiate between the two classes well, and models predicting P2 data are better than predicting P1 data.

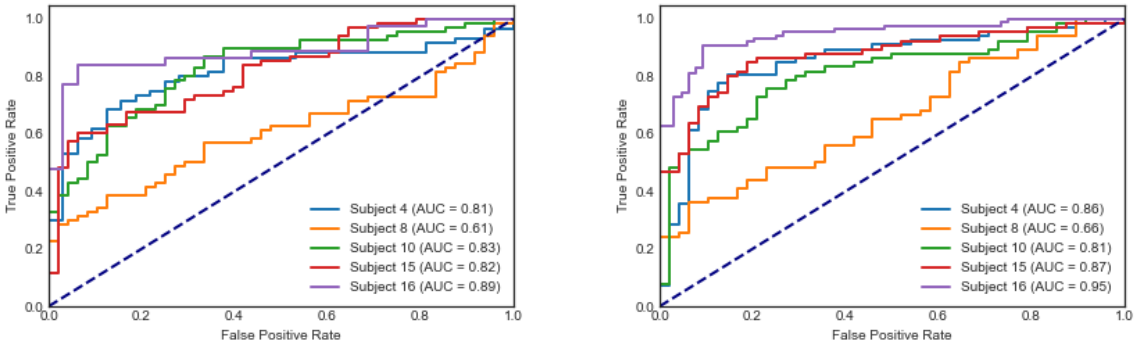
From the **4-class** classification accuracies in fig. 5.17b EEGNet is clearly the method that is performing best. The RF models have some promising results, but the average performance is only around an accuracy of 0.6. It is very even between the performance of EEGNet when using either P1 or P2 data. Results with P2 data have the highest average performance, though slightly below the best performance of the experiment in section 5.9. EEGNet perfectly predicted the P2 data for subject 2 and P1 data for subject 15. Data from both protocols, predicted with the EEGNet, managed to get over half the subjects' data above an accuracy of 0.8, which is very good. From the comparison plot in fig. 5.17b, there has been an improvement for data from both protocols for EEGNet, while only P1 data were improved with the RF model, when compared to the respective ones in section 5.10. However, the EEGNet models' performance has significantly increased more than the RF model has.



(a) Individual subject **resting-state vs task** predictions with P1 and P2 data, dataset A.

(b) Individual subject **4-class** predictions with P1 and P2 data, dataset A.

Figure 5.17: The two figures show the accuracy scores for the experiment in section 5.11 when using transfer learning with data from dataset A on the general models created in section 5.10 with dataset B. The predictions are made with both data from P1, and P2 from all subjects in dataset A. Model and protocol data used are displayed in the legend. Each figure includes two plots, where the left one shows the accuracies created with the general models and the right one compares the results with the respective ones from section 5.10. The red dashed line indicates the chance level for the two classification problems.



(a) ROC curves created with EEGNet on P1 data.

(b) ROC curves created with EEGNet on P2 data.

Figure 5.18: Two ROC curves created with the **resting-state vs task** EEGNet models from the experiment in section 5.11. Only the results for the subjects 4, 8, 10, 15, and 16 are plotted. The calculated AUC score for each curve is presented in the legend.

Chapter 6

Discussion, conclusion and further work

This chapter evaluates the BCI design, protocols, and data acquisition. In addition, the results from both individual models and the more general model will be discussed to understand the findings better. The discussion is followed by a conclusion and a recommendation for further work.

6.1 Discussion

This thesis consists of three main problems that will be discussed separately in a section for each of them. They are the design of the BCI and data acquisition, the results from the individual models, and the creation of a general model.

6.1.1 Protocols and data acquisition

The motivation has been to ease the daily lives of persons with LIS by helping them communicate better. The two main problems with today's solutions for communication platforms for LIS patients: they do not offer privacy, and they are very slow, as mentioned in chapter 3. Using a BCI to communicate, offers both privacy and a classification speed that will make the user feel like communicating at the same pace as the interlocutor.

The study [61] presented in chapter 3, found that direct personal communication was the most desired feature, while the preferred mental strategy to make this happen where attempted speech. The BCI design presented here does not directly follow these wishes. Direct communication can be achieved by, for example, presenting a sequence of pre-set words or sections of the alphabet in the colored boxes. However, the last suggestion will be a prolonged method of spelling words. The mental strategy, perception of colors in the brain, was chosen because of its ease of use and low cognitive effort for the users. In addition, recent studies have shown promising results with this strategy, while attempted speech has yet to show acceptable results.

The same protocol ideas used in this thesis were also used in the study [1], which also suggested improvements to the protocol. These suggested improvements for the protocol were followed when designing the protocols for this thesis. The first problem occurs when classifying raw epochs or not feature extracted epochs, which is the case for EEGNet. Each sample needs to be the same length and shape. In addition, a task shown to a participant has to be of minimum the length used as input to the classification method. A solution is to show the states in random time intervals. However, it is important not to increase the total session time as the participants' attention level declines during the recording session. So, only having the resting state displayed in random intervals, the participant must be alert while not massively increasing the total session time was the solution this thesis used. If it is known that only features extracted from the EEG data, will be used as input in a ML method, then the random interval lengths can be chosen much more freely, as the features always will have the same shape. However, it is a challenge because a too short period of an epoch does not contain enough information for a good classification.

The second improvement suggested in [1] was to collect more metadata from the subjects in the form of a questionnaire, which was done in this thesis. The answers to the questionnaire are challenging to use to remove bad sessions, but they can help explain the performance of the models of specific subjects. The questionnaire asked about the physical and mental state of the participant before and after the recording. Only a few participants had a decrease in their physical and mental states, which can imply that the BCI design does not require a high cognitive effort from the user.

The participants had a problem keeping their attention level high during all sessions, especially towards the end. Not because the tasks required high cognitive effort, but because of the repetitive, low effort tasks over a long stretch of time. In addition, the low effort tasks sometimes led the brain to work on autopilot, subconsciously looking at the states and moving the eyes without focusing on what is displayed. To keep the attention level high, the participants removed the OpenBCI headset and had a pause after half of the sessions were recorded. Though this helped keep attention high, it created a new problem. It is nearly impossible to get the electrodes to touch exactly the same points on the scalp when putting the headset on again. So, the EEG signals are recorded from slightly different places on the scalp during the different sessions. The result may be a lower classification accuracy for the models. However, it may increase the models' generalizability, as the model have to train on slightly different EEG signals. It was mentioned that it was harder to focus when only colors were displayed, as the icons gave a better focus point on the screen for the participant.

Eye movement can be an influential artifact in a EEG signal, though, as it was used to classify the states, these artifacts could not be removed. The most significant problem with eye movement was that the participants performed it too early when going from task state to resting state—caused by the participant expecting the resting state to show at any time. Moving the eyes back to the resting state too early will cause the subsequent resting state sample to have no eye movement information. In contrast, the task state has information on two different eye movement operations.

Blinks are the most significant cause of artifacts in an EEG signal. The participants were instructed to try only to blink when the resting state was exposed, as there are more rest states in a session than specific task states. This instruction led participants to blink each time the rest state was displayed to ensure they did not blink on a task state, leading to more blink artifacts in the dataset. A solution to this problem is to introduce a pause in the session after, for example, 20 task states displayed for 30 seconds, where the participant can blink and change their sitting position if needed.

The recording environment introduced minimal real-life noise and non-physiological artifacts, resulting in a much cleaner signal than expected in an online environment. The only real-life noise was occasional chatter from the hall and construction work, which can be neglected as the noise occurred in short periods of time. The light was constant for all participants, as the blinds were down in front of the window. Therefore, the colors were presented all the time with the same light strength, which may not be the case in an online environment. Electrical noise from electrical equipment may also be more apprehensive in online environments. The life-giving medical equipment to a LIS patient may also interfere with the EEG signals. Pre-processing of the signal can reduce the impact of the noise, as it does with the laboratory signals, though it is still unknown if the processed signal is as clean as the one gathered in the laboratory. The final BCI has to withstand real-life noise; if not, it is of little value, as mentioned in chapter 3.

6.1.2 Classification results from individual models

This section will only discuss the results of experiments whose models are created with only subject-specific data from dataset A. In other words, the model for a subject is trained and tested solely with data from the same subject. The first seven experiments conducted are found in section 5.1 to section 5.7. These experiments showcased what combination of pre-processing, processing, and classification techniques performed best on the data, which is essential for creating a general model.

The results from the processing and classifier evaluation done with the public dataset, in section 5.1, displayed that the designed methods were performing sufficiently. None of the methods managed to compete with the winner of the competition, though they worked sufficiently on data they were not designed to classify. The most interesting observation was that the methods preferred data from different subjects, which raises the question of whether a universal best method exists. Some methods work better on some subjects' data, while others work better on other subjects' data.

The baseline results in section 5.2 gave a good indication of how the methods worked with the data. The main takeaway was that the methods struggled to create a model for some subjects, while the accuracy was nearly perfect for others. The sporadic performance is evident in the **4-class** classification models. In addition, these models also had a much higher accuracy when they were performing well than what was the case for the **resting-state vs task** classification.

Only using EEG signals from the occipital lobe to create models proved difficult. As seen from the results in section 5.4 and in table 6.1. The eye movement recorded from the two electrodes in front is essential. Only subject 12 EEGNet models had a significantly positive performance. The EEGNet models improved for subject 12 because the subject may have blinked and moved their eyes a lot—creating a lot of noise in the data. Still, the performance is good enough to exclude *lucky* prediction. Though this is not enough to conclude, it is promising, and it may be possible to use the BCI design with only color perception. In other words, the design can be used by persons with CLIS after further experiments.

Using PSP data showed a slight increase from the baseline in the experiment from section 5.3, especially for RF. Furthermore, only **resting-state vs task** classification improved from the baseline, see table 6.1. A reason for the models improvement is because CAR is excluded from PSP, and the results in section 5.5 show that CAR introduced noise to the signal. So, even though PSP improved the spatial resolution of the EEG signal and included several different features, it still performed worse than just removing CAR from the pre-processing pipeline. The results from removing CAR in section 5.5 showed that primarily SVM and RF gained on the change. EEGNet did not improve much because it created its own filters, which may have managed to remove or deal with the noise added by CAR. In addition, the gain was most significant for the **4-class** classification, which now managed to create well-performing models for more subjects.

The experiment in section 5.6, where eye blinks were manually removed with ICA from the two electrodes in front, proved unsuccessful. Only the models created with SVM saw a slight improvement. The decrease in performance occurs because ICA only calculates components equal to the number of channels in the input. As the EEG signals are recorded from relatively few channels, the number of components created with ICA is few, resulting in more than one type of information in a component. Even though the blinks were obvious in one component, they were also present in others, and other information was current in the blink components. Resulting in too much valuable information being removed from the signal and the general performance of the models decreased.

Statistical features can be extracted in different ways from an EEG signal. One is to extract it from the raw signals, while another is to extract it from a transformed signal. The latter was preferred in this thesis, as seen from the results in section 5.7. The features were extracted from the transformed signal to reduce the possibility of extracting noise as a feature. The results in section 5.7 show that VMD may be a good technique in a processing pipeline, especially for RF. RF managed to outperform all the models from the other experiments with VMD as can be seen in the summary, presented in table 6.1. The only problem, which was also mentioned in that section, is that VMD is very slow compared to DWT, making it a less feasible technique in an online environment. So, this technique has to outperform other methods by a lot before being considered used in a BCI that needs to classify EEG signals fast. No experiments were conducted to find the best mother wavelet to DWT, which could have increased its accuracy.

A result summary from five of the experiments can be seen in table 6.1. From the table, it is evident that RF was generally the best method to create **resting-state vs task** models, while EEGNet created the best **4-class** classification models. EEGNets performance on the **resting-state vs task** problem was often close to the performance by RF. However, RF only managed to compete with EEGNet on the **4-class** classification problem in the last experiment. The exact result values from the best methods are also shown in table C.1 and table C.2 in appendix C.

Some general observations can be formed from all the experiments discussed in this section. First, SVM always had the worst average accuracy, even though it had some single good performances. Second, models created with P2 data were generally slightly better than models created with P1 data, opposite of what was the case in [1]. P2 models perform better because the participants found it eas-

Table 6.1: The best methods and their respective average accuracy created with **P2** data from the different individual model experiments. The results in section 5.1 and section 5.6 are left out of the table as they are non-comparable to the other experiments. In other words, only the experiments from section 5.2 to section 5.5, and section 5.7 are presented.

Experiment in	**Processing notes**	**Resting-state vs task**		**4-class classification**	
		Method	**Avg. accuracy**	**Method**	**Avg. accuracy**
Section 5.2	Explained in section 4.4.2 and section 4.4.3	RF	0.673	EEGNet	0.526
Section 5.3	Explained in section 4.4.4	RF	0.698	EEGNet	0.422
Section 5.4	Data from only the occiptial lobe	SVM	0.583	EEGNet	0.277
Section 5.5	CAR removed from pre-processing pipeline	RF	0.702	EEGNet	0.577
Section 5.7	DWT against VMD and feature extraction test	RF (VMD stat)	0.743	RF (VMD stat)	0.614

ier to concentrate on the color when there was an icon there. Third, models for **4-class** classification got higher results than the ones for **resting-state vs task** classification. It is unfortunate that **resting-state vs task** models do not get accuracies closer to 1.0 when the subject only watched a grey-colored square on the screen. There will be much more noise and movement in front of the user's eyes in an online environment, which can be easier interpreted as a task state.

In all experiments, two classification problems have been examined, differences between a resting state and a task state, and between the four task states. The same evaluation metric, accuracy, compares the results for both problems. The problem with using only accuracy is that it does not give a complete picture of how a model performs. Accuracy says nothing about which class was most popular or if a model primary only guesses one state. For the **resting-state vs task** classification, this is a problem. It is okay to predict the task state on a rest state, though the other way around is problematic. As a BCI performing tasks while the user is resting can lead to significant complications. To better understand the models, considering other metrics for evaluation, such as a receiver operating characteristic curve and the rate of false task state predictions, can give a more thorough assessment of the **resting-state vs task** model. Only accuracy has been used in this thesis because it gives a valuable initial feeling of how the model performs. In addition, the datasets are balanced. So, the chance level gives a good indication of how the models perform, as models close to the chance level usually only predict one class.

The classifiers in the different experiments were initialized with the same hyperparameters. There were no experiments conducted to optimize them. Optimization of the hyperparameters could have improved the results further, but this thesis focuses on the processing pipeline, and therefore, these experiments were dismissed. In addition, from the results in [1], the default values for EEGNets hyperparameters performed better than the model created after a hyperparameter search.

The individual models were trained and tested with data from the three sessions with the same protocol. All sessions were mixed before splitting into the three sets used to increase the training set for model creation and evaluation. The usual practice in this field is to train and test in the same session. A session in this thesis has only 20 samples of each task state, implying that the training set would likely overfit the few samples it had to train on after the data splitting. Another solution to this dilemma could be to use the first two sessions as the training and validation set and the last session as a test set. It might give a better picture of how the model would have performed on new data, which would be the case in a real-life test. The data pre-processing and processing were done sample-wise, as this mimics real life.

6.1.3 Towards a general model

A general model is crucial for the online implementation of the designed BCI, as it makes the BCI more available and easier to use. In this thesis, two different classification problems have been experimented on, implying that the finished BCI has to contain two different models. There is no problem that the two models are different classification methods, though the classification speed can be reduced if the processing pipeline is different. If the two models are already trained general models, the computational power needed to train a model decreases. Hence, the BCI will be more available as it can work on tech devices with low computational capacity, which is important as the IDUN cluster will not be available in later stages.

The four last experiments were conducted to test if a general model could be created for the BCI design, and a summary of their average results is presented in table 6.2. The first experiment, see section 5.8, created 22 LOOMs to examine how data from a new subject would perform on a general model. The results showed that SVM were unsuitable for a general model and were therefore not included in the final three experiments. The **resting-state vs task** classification performance also slightly decreased for both EEGNet and RF. However, the **4-class** classification showed promising results for a general model, especially for EEGNet.

Table 6.2: The best methods and their respective average accuracy created with **P2** data from the experiments for creating a general model. So, the experiments from section 5.8 to section 5.11 are presented.

Experiment in	Experiment notes	Resting-state vs task		4-class classification	
		Method	Avg. accuracy	Method	Avg. accuracy
Section 5.8	LOOM	RF	0.681	EEGNet	0.669
Section 5.9	LOOM with transfer learning	EEGNet	0.709	EEGNet	0.746
Section 5.10	General model	RF	0.614	EEGNet	0.698
Section 5.11	General model with transfer learning	EEGNet	0.698	EEGNet	0.736

In section 5.9, transfer learning was utilized to improve the LOOMs created in section 5.8. A model must be trained when using transfer learning, increasing the computational cost. However, the training can be done on a small dataset and only on a few hidden layers without the model overfitting, which implies that the computational cost needed is low. The same is true for a warm start as well. The results in section 5.9 demonstrated the potential for transfer learning for EEGNet and that a warm start for RF was more of a waste of time and computational power. The different results between the two methods can come from the continuation of updating the weights in EEGNet to fit the new data better, while RF only adds new DTs in its forest. Adding more trees to an already created forest makes the model more uncertain in its predictions.

The two previous experiments showed the feasibility of creating a general model. The experiment in section 5.10 created only one general model on a different dataset, which was tested on data from each subject. There is one problem with the general model created on dataset B. In the optimization to find the best subject selection, the testing set changed accordingly to the subject selection. So, the models were tested on different data, making them more challenging to compare. This problem could have been solved by using a subset of dataset A as the testing set. However, this solution could also decrease the general models' generalizability, as it now tries to optimize results on dataset A.

The confusion matrices to the two general **resting-state vs task** models, fig. 5.13, showed that the models preferred to predict the task state rather than the resting state. As mentioned in section 6.1.2, this prediction execution is problematic because predicting a false task state, can lead to unwanted complications. Furthermore, using such a model for transfer learning will make the new models more fond of predicting task states. There are several solutions to fix this problem. One is to use class weights when training, making the model prefer to predict rest states. This solution will reduce the number of false task states, though, with the cost of the general accuracy of the model. Another method is to imbalance the training set by having more rest state samples than task states. A final solution is to optimize the model around another or several metrics that can give a broader picture of the model's performance.

The confusion matrices for the **4-class** classification models, fig. 5.14, showcased an acceptable ability to differentiate between the four task states. If the models had differentiated the task states perfectly, the models might have overfitted, and the models would not have been general enough for new data. The results in this section, section 5.10, showcased precisely that the models were general enough to make good predictions on the **4-class** classification problem. However, this was not the case for the **resting-state vs task** problem results, as the general models had problems getting an accuracy score above 0.8.

The problem with **resting-state vs task** classification can be explained from the feature relevance plot for the rest state created with DeepLIFT, fig. 5.15a. The plot shows that the EEGNet model mainly extracts information from the two front electrodes between the halfway point and the end of an epoch. This period was usually when the participants blinked to prepare for the next task state. In other words, the model has learned that blinking means rest state and is looking for blinks in an epoch, which may not be the case for all epochs. This learned idea for the model is an issue, as a real-life resting state usually does not contain blinks at all. The model's best way to remove this learned feature is to record more resting state data mimicking a real-world situation. For example, the participants do more or less nothing and are periodically exposed to real-world noise, such as music or chatter.

The features relevance plot for the **4-class** classification, fig. 5.15b, expressed that the information in the two front electrodes was mainly used to make predictions. The relevant period was at the sample's start when the participants moved their eyes. However, the model also used information from the occipital lobe to differentiate between the task states, implying that a model predicting samples with only data from the occipital lobe may be feasible. Using data from only the occipital lobe could make the BCI available for persons with CLIS. However, as the experiment results in section 5.4 showed, there is still a long way to make this possible.

In the final experiment, section 5.11, were transfer learning is used to examine if a short training with the expected data could improve the general model from section 5.10. Transfer learning showed that creating a model specified to one specific subject is the way to create a model as the accuracies increased significantly. See the summary presented in table 6.2. The results showed that EEGNet was superior to RF and had the best average results for both classification problems. The best average accuracy for **resting-state vs task** classification was 69.8%, while for **4-class** classification, 73.6%. The exact result values for the EEGNet and RF are presented in table 2 and table 3 in appendix C. The ROC plots also showed that EEGNet managed to differentiate between the two classes well.

The performance on P2 data was very good, even though the general models were trained with P1 data. A reason is that the participants said after the recording that it was easier to keep the focus when icons were displayed. In other words, the P2 samples may be easier to classify than the P1 samples. Another reason is that the two protocols are so similar that the same feature can be extracted when predicting from both protocols. If this is the case, there may be a possibility of mixing the data from both protocols, to get a larger dataset from which a general model can be created.

RF was primarily outperformed in all the experiments by EEGNet. However, if more time had been used to optimize the hyperparameters to RF, it might have changed. The only time RF slightly outperformed EEGNet, was when using the same general **resting-state vs task** model to test the subject data without any transfer learning. These results show in the long run that RF may be better for creating a general model that does not need any training before running on a new subject. For a general **4-class** model, EEGNet was the best to predict the task states. So, in a future BCI that do not need any training should use RF for **resting-state vs task** and EEGNet for **4-class** classification.

These four experiments have displayed that transfer learning on a general model is significantly better than using only a general model. However, the average accuracy of using transfer learning on a general model was slightly lower than when using transfer learning on the LOOMs, as shown in table 6.2. This behavior can come from the cue marking problem still present in the P1 data from dataset B, as shown in section 4.3.2. Training a general model with data with no problems or errors, such as dataset A, may further increase the average accuracy for a general model for the two classification problems.

4-class classification significantly improved when using transfer learning on a general model, compared to the best results from the individual models created with RF with VMD stat processed, the experiment in section 5.7. However, for **resting-state vs task** classification, the accuracies have decreased, showing that the resting-state may differ greatly from person to person and from different mental states. Using pre-trained weights may leave the model in a local optimum that does not fit the new subjects' resting state. So, there may be a greater need to collect new resting state data from a subject that will use the BCI, as this is more different from subject to subject.

6.2 Conclusion

The main goal of this research was to design a BCI paradigm that could predict fast with high accuracy and be comfortable for the user to use over an extended period. The classification speed is primarily specified by the processing pipeline and the length of the epoch. In this thesis, the 2 seconds epoch length is the bottleneck for faster classification for the user. The dataset recording showed that eye movement and color exposure requires minimal mental work. Rather, the participants got tired from such an easy task instead of mentally exhausted from challenging mental tasks.

The search for high classification accuracy has been pursued in this thesis, as can be seen from the defined objectives in chapter 1. The first objective was to explore different feature extraction methods and ML classifiers. Several state-of-the-art pre-processing, decomposition, and feature extraction methods were tested. The experiments showed that the pre-processing method CAR, included for noise removal, added eye artifacts to all channels. Furthermore, they demonstrated the problem with a stable resting state and that the models created in this thesis have mainly learned that a resting state consists of eye blinks.

A summary of the results for the two last objectives is displayed in table 6.3. The second objective was to find the best method to differentiate between a task state and a resting state and the best method for differentiating the four task states. The experiments showed that models created with P1 and P2 data performed similarly well, though models created with P2 data were slightly better. RF was the best classification method for both classification problems, with the average accuracies of 74.3% and 61.4% on P2 data from the 22 subjects from dataset A for **resting-state vs task** and **4-class** classification, respectively. The processing line to get these results was pre-processing an epoch with Notch and bandpass filters before fractals, energies, and statistical features were extracted from the five modes created with VMD. It should be noted that the CNN EEGNet, with epochs only pre-processed with Notch and bandpass filter, performed just slightly lower than RF on average. However, it created the best-performing models.

Table 6.3: The table presents the best methods and their respective average accuracy created with **P2** data from the experiments finding the best method for models created on individual data and for creating a general model. The processing column displays the pre-processing, decomposition method, and feature extraction used.

Experiment type	**Resting-state vs task**			**4-class classification**		
	Processing	**Method**	**Avg. accuracy**	**Processing**	**Method**	**Avg. accuracy**
Individual models	Notch, bandpass VMD, fractals, energies, and statistical features	RF	74.3 %	Notch, bandpass VMD, fractals, energies, and statistical features	RF	61.4 %
General model	Notch, bandpass and transfer learning	EEGNet	69.8 %	Notch, bandpass and transfer learning	EEGNet	73.6 %

The final objective was to create a general model that could predict well on data from subjects and sessions the model had never seen before. The model was created on data from dataset B and tested on dataset A. The experiments showed that using the general EEGNet model, as a base model and doing transfer learning with a portion of the subject data the model should predict from—creates the best results for both classification problems. The average accuracies for **resting-state vs task** and **4-class** classification were 69.8% and 73.6%, respectively. Where over half of the subject **4-class** models achieved accuracy above 80 %. These final results demonstrate that it may be possible to have a general model in the designed BCI, that only requires a short recalibration with a few training samples from the user. Before the BCI is ready to use.

6.3 Further work

Jean-Dominique Bauby writes in his book that his locked-in body becomes less oppressive when his mind takes flight like a butterfly [5]. A BCI can bring new persons along with a locked-in person's mind journey and make that person's thoughts available. The designed BCI in this thesis is still far from making this happen, though further work can make it feasible.

The next step could be to create better resting-state samples so the models do not learn that blinking means that a person is resting. As mentioned in the discussion, this can be done by continuously recording a participant's brain activity while that person is just resting and occasionally looks at a task state. Periodically exposing the subject to noise from, for example, music or chatter can help with mimicking a real-world setting. Note that this method will only be feasible to implement in a general model. The best classification model should manage to differentiate between a user wanting to perform a task and everything else the user may think or see.

There is also a need to increase the accuracies further. Four methods that can positively affect the models' performance are optimizing the models' hyperparameters with, for example, Hyperband [89]. Next is to find the best decomposition method and features by running an NSGA-II optimization. The third method is to classify the data with Riemannian geometry, which has shown promising results in EEG classification, as mentioned in chapter 3. The final method is to create a general model with better-epoched data, for example, from dataset A, and use transfer learning on data from a new recording and subject.

When the accuracies have reached a satisfactory level, a real-time implementation of the model can be developed. The model performance can then be evaluated in an online environment, testing the feasibility of the designed BCI. The designed BCI should also be evaluated by discussing its advantages and disadvantages with persons with LIS and their caregivers and consulting what uses they can see the designed BCI may have for them. In addition, recording a dataset on one or several persons with LIS. This dataset can be used to improve the classification models further. While also indicating the feasibility of the BCI paradigm for persons with LIS.

Transfer learning on a general CNN EEGNet model showed the most promising results. However, it is essential to understand how much training data and time the pre-trained model needs to recalibrate, before it can be used to classify new data. Less data and time means that the model can be implemented on tech devices with low computational power, while more data and time demand higher computational power. Finally, the classification time is currently only decided by the length of the input epoch to a classifier. The best length of an epoch can be found by conducting experiments with different sub-lengths of the epochs. Performing experiments trying to reduce the period of an input epoch while also increasing the number of task states will decrease the BCI's classification time in real-time. Decreasing the classification time is a crucial step in reducing the time a person with LIS needs to express themself.

The following list summarizes further work necessary for the BCI to be available and aid a person with LIS:

- Record a dataset where the main goal is to capture the subjects' resting state.
- Try to improve the classification problems, evaluate the models on several different metrics, for example, by:
  - Hyperparameter optimization,
  - Feature extraaction optimization,
  - Riemannian geometry classification, or,
  - Create a new optimized general model with better data used in transfer learning.
- Shorten the length of the input epochs while not decreasing the model's performance
- Create a state machine of the models and do an online test to understand the perfect balance between transfer learning with new data and the accuracy of the BCI.
- Discuss the BCI design with persons with LIS and their caregivers.
- Record a dataset on persons with LIS.

References

[1] Tobias Treider Moe. Designing an eeg based communication system for patients with locked-in syndrome. Technical report, NTNU, 2021.

[2] Tobias Treider Moe. Lis eeg protocol. `https://github.com/2bben/LIS-EEG-Protocol`, 2022.

[3] Tobias Treider Moe. Lis eeg classification. `https://github.com/2bben/LIS-EEG-Classification`, 2022.

[4] Taras Halan, Juan Fernando Ortiz, Dinesh Reddy, Abbas Altamimi, Abimbola O Ajibowo, and Stephanie P Fabara. Locked-In Syndrome: A Systematic Review of Long-Term Management and Prognosis. *Cureus*, 13(7), 2021.

[5] Jean-Dominique Bauby. *The diving bell and the butterfly*. SD Books, [1997] 2019.

[6] Trude Lorentzen. Mathias (21) er norges yngste med locked-in syndrom. `https://www.nrk.no/osloogviken/xl/mathias-_21_-er-norges-yngste-med-locked-in-syndrom-1.15932211`, Apr 2022. [Online; accessed 29-April-2022].

[7] Veronica Johansson, Surjo R Soekadar, and Jens Clausen. Locked out: Ignorance and responsibility in brain–computer interface communication in locked-in syndrome. *Cambridge Quarterly of Healthcare Ethics*, 26(4):555–576, 2017.

[8] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

[9] Xiaorong Gao, Yijun Wang, Xiaogang Chen, and Shangkai Gao. Interface, interaction, and intelligence in generalized brain–computer interfaces. *Trends in Cognitive Sciences*, 2021.

[10] Eimear Smith and Mark Delargy. Locked-in syndrome. *Bmj*, 330(7488):406–409, 2005.

[11] Kajsa Svernling, Marie Törnbom, Åsa Nordin, and Katharina S Sunnerhagen. Locked-in syndrome in Sweden, an explorative study of persons who underwent rehabilitation: a cohort study. *BMJ open*, 9(4):e023185, 2019.

[12] Sunnaas sykehus. Locked-in syndrom. `https://www.sunnaas.no/behandlinger/locked-in-syndrom`, (n.d.). [Online; accessed 20-April-2022].

[13] Gerhard Bauer, Franz Gerstenbrand, and Erik Rumpl. Varieties of the locked-in syndrome. *Journal of neurology*, 221(2):77–91, 1979.

[14] Terese Winslow. National cancer institute - pons. `https://www.cancer.gov/publications/dictionaries/cancer-terms/def/pons`, 2010. [Online; accessed 20-April-2022].

[15] Johns Hopkins Medicine. Brain anatomy and how the brain works. `https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain`, journal=Johns Hopkins Medicine, (n.d.). [Online; accessed 14-December-2021].

[16] Saeid Sanei. *Adaptive processing of brain signals*. John Wiley & Sons, 2013.

[17] D. Purves, D. Fitzpatrick, L.C. Katz, A.S. Lamantia, J.O. McNamara, S.M. Williams, and G.J. Augustine. *Neuroscience*. Sinauer Associates, 5 edition, 2011.

[18] R. Tootell and S. Nasr. Primate color vision. *Brain Mapping*, pages 489–506, 2015.

[19] Per Brodal. *Sentralnervesystemet*. Universitetsforlaget, Oslo, 5 edition, 2012.

[20] Inês Bramão, Luís Faísca, Christian Forkstam, Alexandra Reis, and Karl Magnus Petersson. Cortical brain regions associated with color processing: An FMRI study. *The open neuroimaging journal*, 4:164, 2010.

[21] K Brodmann. Vergleichende lokalisationslehre der groβhirnrinde: in ihren prinzipien dargestellt auf grund des zellenbaues (english translation published as brodmann's localisation in the cerebral cortex: The principles of comparative localization in the cerebral cortex based on cytoarchitectonics). *Transl. ed Garey LJ (Leipzig: Johann Ambrosius Barth*, 1909.

[22] John H. Martin. *The Visual System, Neuroanatomy: Text and Atlas*. McGraw Hill, New York, NY, 5 edition, 2021.

[23] J.L. Chan, A. Kucyi, and J.F.X. DeSouza. Oculomotor system. *Brain Mapping*, pages 483–488, 2015.

[24] Rodney J Croft and Robert J Barry. Removal of ocular artifact from the eeg: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 30(1):5–19, 2000.

[25] Luis Alfredo Moctezuma. *Towards Universal EEG systems with minimum channel count based on Machine Learning and Computational Intelligence*. PhD thesis, Norwegian University of Science and Technology, 2021.

[26] D. Millett, P. Coutin-Churchman, and J.M. Stern. Basic principles of electroencephalography. *Brain Mapping*, pages 75–80, 2015.

[27] St Louis Erik K., Lauren C. Frey, and Jeffrey W. Britton. *Electroencephalography (EEG): An introductory text and Atlas of Normal and abnormal findings in adults, children, and infants*. American Epilepsy Society, 2016.

[28] Andrea Biasiucci, Benedetta Franceschiello, and Micah M Murray. Electroencephalography. *Current Biology*, 29(3):R80–R85, 2019.

[29] Jose Antonio Urigüen and Begoña Garcia-Zapirain. Eeg artifact removal—state-of-the-art and guidelines. *Journal of neural engineering*, 12(3):031001, 2015.

[30] William O Tatum, Barbara A Dworetzky, and Donald L Schomer. Artifact and recording concepts in EEG. *Journal of clinical neurophysiology*, 28(3):252–263, 2011.

[31] D. Bzdok and S.B. Eickhoff. The resting-state physiology of the human cerebral cortex. *Brain Mapping*, pages 203–209, 2015.

[32] Jinhu Xiong, Liangsuo Ma, Binquan Wang, Shalini Narayana, Eugene P Duff, Gary F Egan, and Peter T Fox. Long-term motor training induced changes in regional cerebral blood flow in both task and resting states. *Neuroimage*, 45(1):75–82, 2009.

[33] Li Hu and Zhiguo Zhang. *EEG Signal Processing and Feature Extraction*. Springer Singapore Pte. Limited, Singapore, 2019.

[34] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[35] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *sensors*, 12(2):1211–1279, 2012.

[36] Chang-Hee Han, Klaus-Robert Müller, and Han-Jeong Hwang. Brain-switches for asynchronous brain–computer interfaces: A systematic review. *Electronics*, 9(3):422, 2020.

[37] Kip A Ludwig, Rachel M Miriani, Nicholas B Langhals, Michael D Joseph, David J Anderson, and Daryl R Kipke. Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *Journal of neurophysiology*, 101(3):1679–1689, 2009.

[38] Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544, 2013.

[39] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995, 1998.

[40] Mark G Frei and Ivan Osorio. Intrinsic time-scale decomposition: time–frequency–energy analysis and real-time filtering of non-stationary signals. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2078):321–342, 2007.

[41] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[42] B Sivakumar. A phase-space reconstruction approach to prediction of suspended sediment concentration in rivers. *Journal of Hydrology*, 258(1-4):149–162, 2002.

[43] Sang-Hong Lee, Joon S Lim, Jae-Kwon Kim, Junggi Yang, and Youngho Lee. Classification of normal and epileptic seizure eeg signals using wavelet transform, phase-space reconstruction, and euclidean distance. *Computer methods and programs in biomedicine*, 116(1):10–25, 2014.

[44] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[45] Emmanuel Didiot, Irina Illina, Dominique Fohr, and Odile Mella. A wavelet-based parameterization for speech/music discrimination. *Computer Speech & Language*, 24(2):341–357, 2010.

[46] Hesham Tolba and Douglas O'Shaughnessy. Automatic speech recognition based on cepstral coefficients and a mel-based discrete energy operator. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 2, pages 973–976. IEEE, 1998.

[47] Cindy Goh, Brahim Hamadicharef, Goeff Henderson, and Emmanuel Ifeachor. Comparison of fractal dimension algorithms for the computation of eeg biomarkers for dementia. In *2nd international conference on computational intelligence in medicine and healthcare (CIMED2005)*, 2005.

[48] Tomoyuki Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2):277–283, 1988.

[49] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[50] Li-Chen Shi, Ying-Ying Jiao, and Bao-Liang Lu. Differential entropy feature for eeg-based vigilance estimation. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6627–6630. IEEE, 2013.

[51] P Russel Norvig and S Artificial Intelligence. *A modern approach*. Prentice Hall Upper Saddle River, NJ, USA:, 2002.

[52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[53] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[54] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[55] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[56] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[57] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[58] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[59] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[60] Shanu Verma, Millie Pant, and Vaclav Snasel. A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*, 9:57757–57791, 2021.

[61] Mariana P Branco, Elmar GM Pels, Femke Nijboer, Nick F Ramsey, and Mariska J Vansteensel. Brain-Computer interfaces for communication: preferences of individuals with locked-in syndrome, caregivers and researchers. *Disability and Rehabilitation: Assistive Technology*, pages 1–11, 2021.

[62] Han-Jeong Hwang, Kiwoon Kwon, and Chang-Hwang Im. Neurofeedback-based motor imagery training for brain–computer interface (BCI). *Journal of neuroscience methods*, 179(1):150–156, 2009.

[63] Upasana Talukdar, Shyamanta M Hazarika, and John Q Gan. Adaptation of Common Spatial Patterns based on mental fatigue for motor-imagery BCI. *Biomedical Signal Processing and Control*, 58:101829, 2020.

[64] Upasana Talukdar, Shyamanta M Hazarika, and John Q Gan. Motor imagery and mental fatigue: inter-relationship and EEG based estimation. *Journal of computational neuroscience*, 46(1):55–76, 2019.

[65] Evangelos Antoniou, Pavlos Bozios, Vasileios Christou, Katerina D Tzimourta, Konstantinos Kalafatakis, Markos G Tsipouras, Nikolaos Giannakeas, and Alexandros T Tzallas. EEG-Based Eye Movement Recognition Using the Brain–Computer Interface and Random Forests. *Sensors*, 21(7):2339, 2021.

[66] Nataliya Kosmyna, Franck Tarpin-Bernard, Nicolas Bonnefond, and Bertrand Rivet. Feasibility of bci control in a realistic smart home environment. *Frontiers in human neuroscience*, 10:416, 2016.

[67] U Chaudhary, N Birbaumer, and MR Curado. Brain-machine interface (bmi) in paralysis. *Annals of physical and rehabilitation medicine*, 58(1):9–13, 2015.

[68] Ujwal Chaudhary, Niels Birbaumer, and Ander Ramos-Murguialday. Brain–computer interfaces for communication and rehabilitation. *Nature Reviews Neurology*, 12(9):513–525, 2016.

[69] Yoji Okahara, Kouji Takano, Masahiro Nagao, Kiyohiko Kondo, Yasuo Iwadate, Niels Birbaumer, and Kenji Kansaku. Long-term use of a neural prosthesis in progressive paralysis. *Scientific reports*, 8(1):1–8, 2018.

[70] Chang-Hee Han, Yong-Wook Kim, Seung Hyun Kim, Zoran Nenadic, Chang-Hwan Im, et al. Electroencephalography-based endogenous brain–computer interface for online communication with a completely locked-in patient. *Journal of neuroengineering and rehabilitation*, 16(1):1–13, 2019.

[71] Eric W Sellers, David B Ryan, and Christopher K Hauser. Noninvasive brain-computer interface enables communication after brainstem stroke. *Science translational medicine*, 6(257):257re7–257re7, 2014.

[72] Xavier Duart, Eduardo Quiles, Ferran Suay, Nayibe Chio, Emilio García, and Francisco Morant. Evaluating the effect of stimuli color and frequency on ssvep. *Sensors*, 21(1):117, 2020.

[73] Sara Hegdahl Åsly. Supervised learning for classification of eeg signals evoked by visual exposure to rgb colors. Master's thesis, NTNU, 2019.

[74] Sara Lund Ludvigsen and Emma Horn Buøen. The augmented human: Development of bci for rgb colour-based automation. Master's thesis, NTNU, 2021.

[75] Alejandro A Torres-García, Luis Alfredo Moctezuma, Sara Åsly, and Marta Molinas. Discriminating between color exposure and idle state using eeg signals for bci application. In *2019 E-Health and Bioengineering Conference (EHB)*, pages 1–5. IEEE, 2019.

[76] Wessam Shehieb, Sara Alansari, and Nada Jadallah. EEG-based communication system for patients with locked-in syndrome using fuzzy logic. In *2017 10th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE, 2017.

[77] Chongsheng Zhang, Changchang Liu, Xiangliang Zhang, and George Almpanidis. An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications*, 82:128–150, 2017.

[78] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

[79] OpenBCI Online Store. Ultracortex 'mark iv' eeg headset. `https://shop.openbci.com/collections/frontpage/products/ultracortex-mark-iv?variant=23280741699`, (n.d.). [Online; accessed 14-December-2021].

[80] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker,

Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[81] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[82] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[83] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[84] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[85] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013.

[86] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[87] Magnus Själander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure, 2019.

[88] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[89] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

Appendix A

Experimental setup

A.1 Questionnaire to Dataset A

Table A.1: Questionnaire given to all participants before the recording started and after all sessions for the subjecte were finished to dataset A.

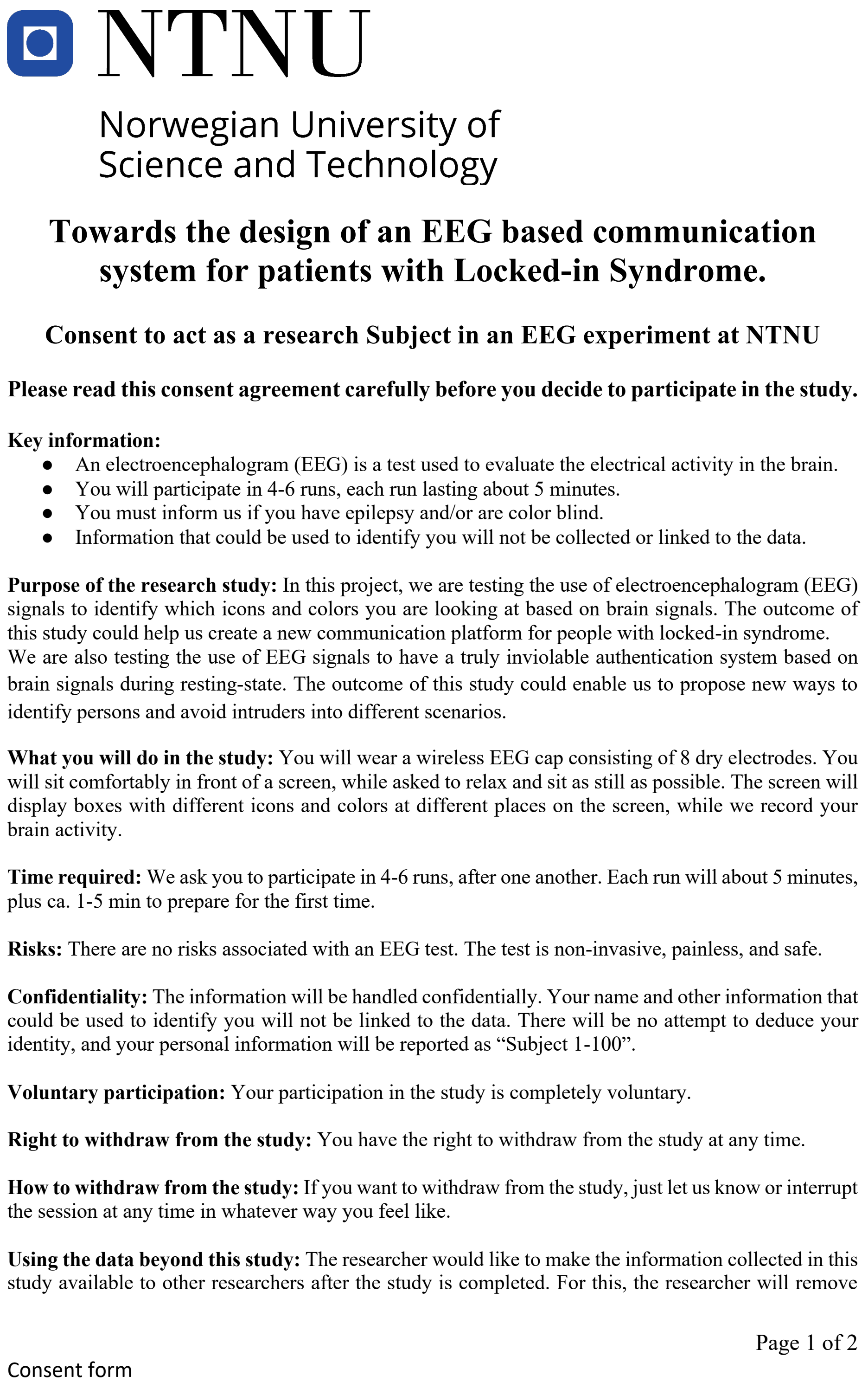
Before	Handedness	left / right	
	How long did you sleep?	no. hours	
	Did you drink coffee the past 24 hours?	no. hours before	
	Did you drink alcohol the past 24 hours?	no. hours before	
	Did you smoke within the past 24 hours?	no. hours before	
Before and after	How do you feel? (Physical state)	1-5	1 = very bad/tired, 5 = very good
	How do you feel? (Mental state)	1-5	1 = very bad/tired, 5 = very good
After	How do you feel? (Attention level)	1-5	1 = low, 5 = high
	Did you look away from the screen?	no. of times	
	Did you close your eyes consciosly in a task-state?	no. of times	
	How many triales (state-changes) do you think you missed?	no. of times	
	Other comments?		

A.2 Subject conversion between dataset A and dataset B

Table A.2: This table presents the number of subjects who participated in datasets A and B. The two numbers on the same row correspond to data from the same subject.

Dataset A	**Dataset B**
1	2
2	7
3	18
4	34
5	26
6	4
7	15
8	6
9	9
10	11
11	12
12	13
13	16
15	22
16	24
20	29
21	33
22	32

A.3 Consent form



NTNU

Norwegian University of
Science and Technology

Towards the design of an EEG based communication system for patients with Locked-in Syndrome.

Consent to act as a research Subject in an EEG experiment at NTNU

Please read this consent agreement carefully before you decide to participate in the study.

Key information:

- An electroencephalogram (EEG) is a test used to evaluate the electrical activity in the brain.
- You will participate in 4-6 runs, each run lasting about 5 minutes.
- You must inform us if you have epilepsy and/or are color blind.
- Information that could be used to identify you will not be collected or linked to the data.

Purpose of the research study: In this project, we are testing the use of electroencephalogram (EEG) signals to identify which icons and colors you are looking at based on brain signals. The outcome of this study could help us create a new communication platform for people with locked-in syndrome.
We are also testing the use of EEG signals to have a truly inviolable authentication system based on brain signals during resting-state. The outcome of this study could enable us to propose new ways to identify persons and avoid intruders into different scenarios.

What you will do in the study: You will wear a wireless EEG cap consisting of 8 dry electrodes. You will sit comfortably in front of a screen, while asked to relax and sit as still as possible. The screen will display boxes with different icons and colors at different places on the screen, while we record your brain activity.

Time required: We ask you to participate in 4-6 runs, after one another. Each run will about 5 minutes, plus ca. 1-5 min to prepare for the first time.

Risks: There are no risks associated with an EEG test. The test is non-invasive, painless, and safe.

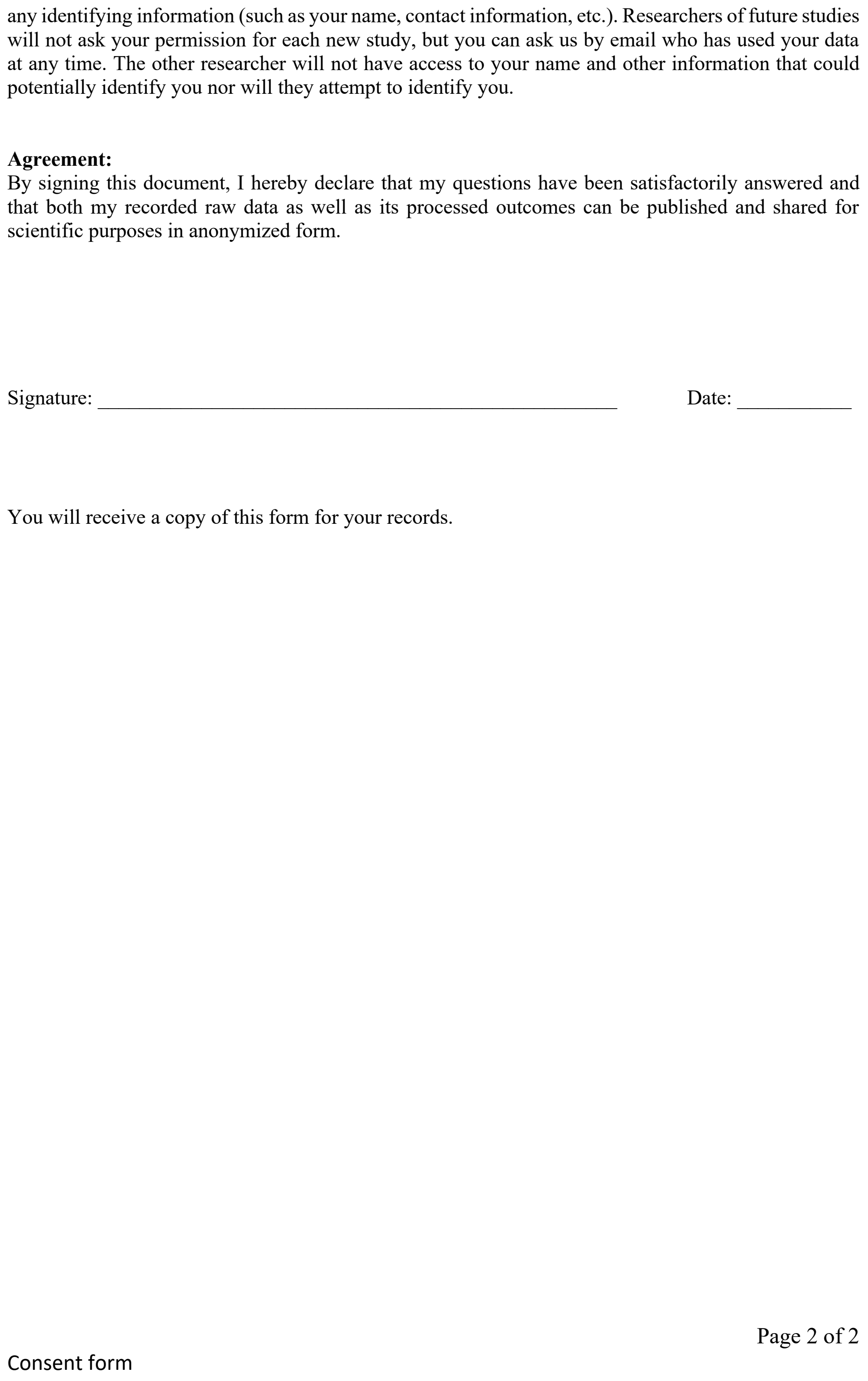
Confidentiality: The information will be handled confidentially. Your name and other information that could be used to identify you will not be linked to the data. There will be no attempt to deduce your identity, and your personal information will be reported as “Subject 1-100”.

Voluntary participation: Your participation in the study is completely voluntary.

Right to withdraw from the study: You have the right to withdraw from the study at any time.

How to withdraw from the study: If you want to withdraw from the study, just let us know or interrupt the session at any time in whatever way you feel like.

Using the data beyond this study: The researcher would like to make the information collected in this study available to other researchers after the study is completed. For this, the researcher will remove



any identifying information (such as your name, contact information, etc.). Researchers of future studies will not ask your permission for each new study, but you can ask us by email who has used your data at any time. The other researcher will not have access to your name and other information that could potentially identify you nor will they attempt to identify you.

Agreement:
By signing this document, I hereby declare that my questions have been satisfactorily answered and that both my recorded raw data as well as its processed outcomes can be published and shared for scientific purposes in anonymized form.

Signature: ____ Date: ___________

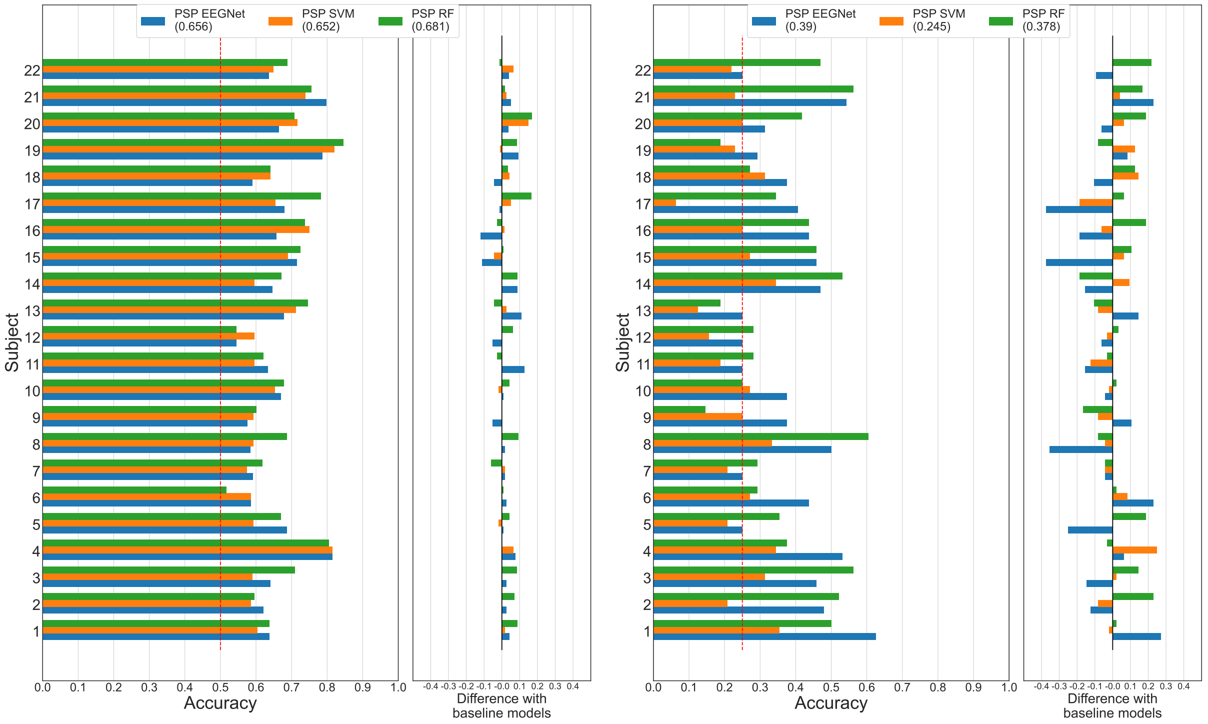
You will receive a copy of this form for your records.

Page 2 of 2

Consent form

Appendix B

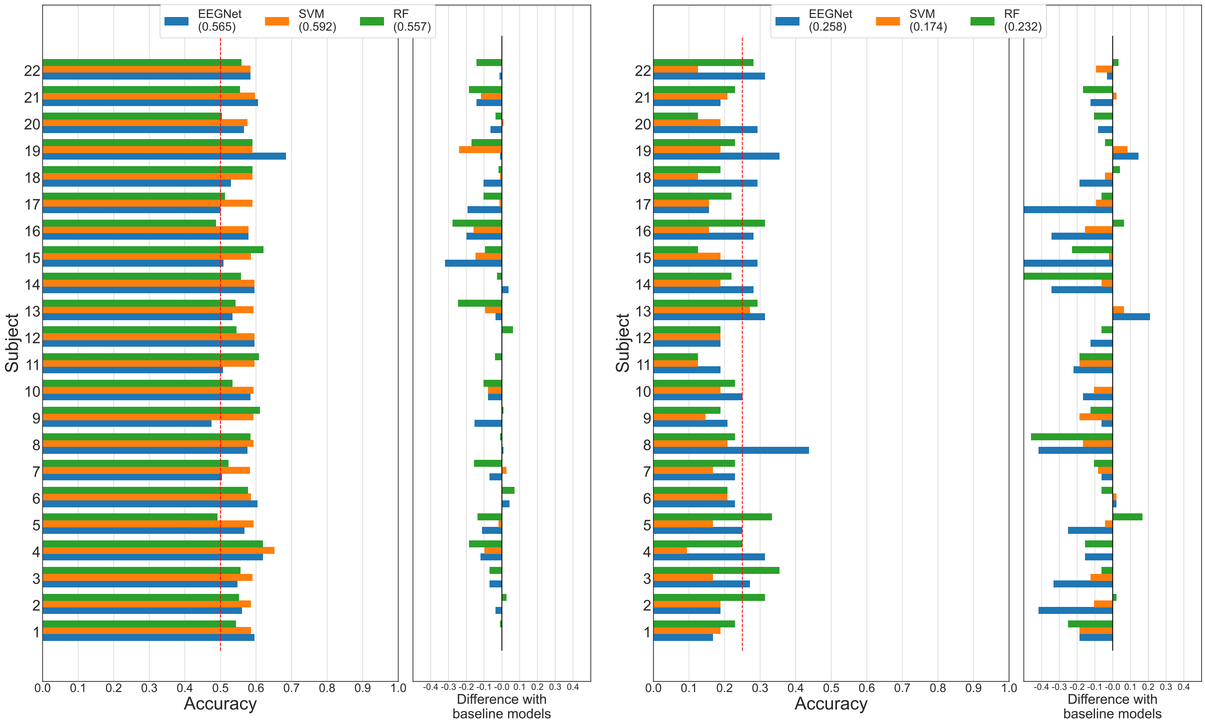
Additional results from the experiments



(a) Individual **resting-state vs task** with P1 data, dataset A.

(b) Individual **4-class** classification with P1 data, dataset A

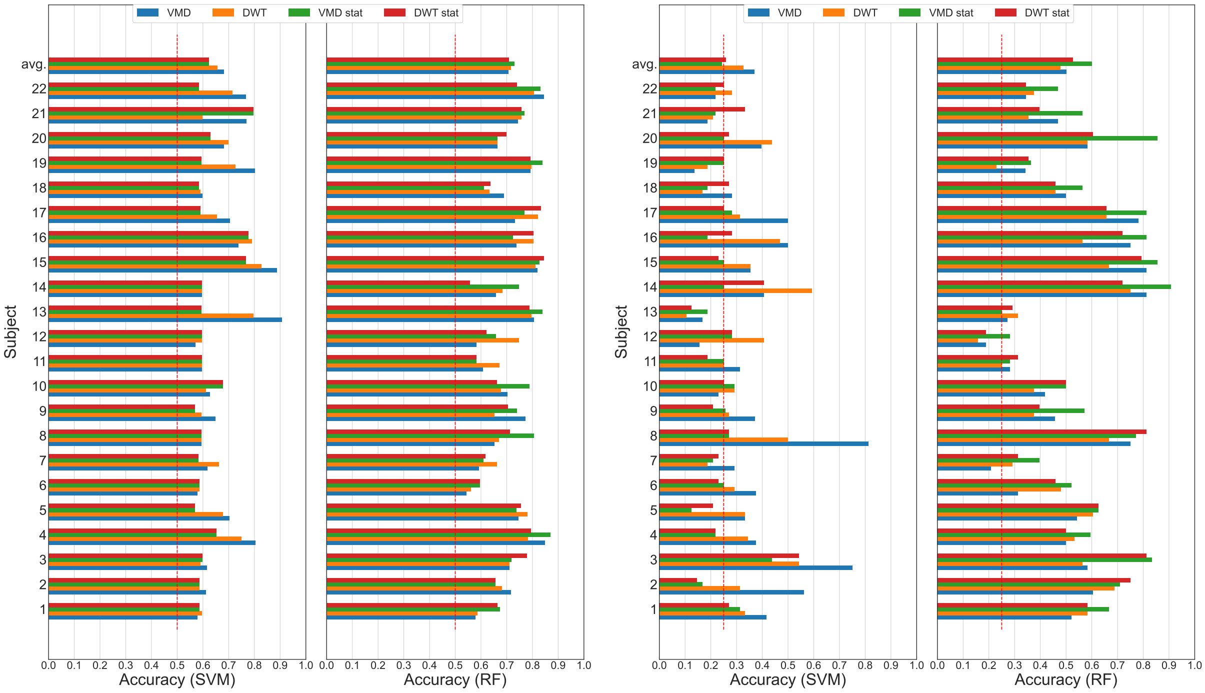
Figure B.1: Accuracies from creating individual EEGNet, SVM, and RF models with PSP data, from **P1**, for the experiments in section 5.3. Below the names of the methods in the legend is the average accuracy for that classification method. Each figure contains two plots, where the left one shows the accuracies created with PSP data, and the right one compares the results with the respective ones from section 5.2. The red dashed line illustrates the chance level for that classification.



(a) Individual **resting-state vs task** with P1 data, dataset A.

(b) Individual **4-class** classification with P1 data, dataset A.

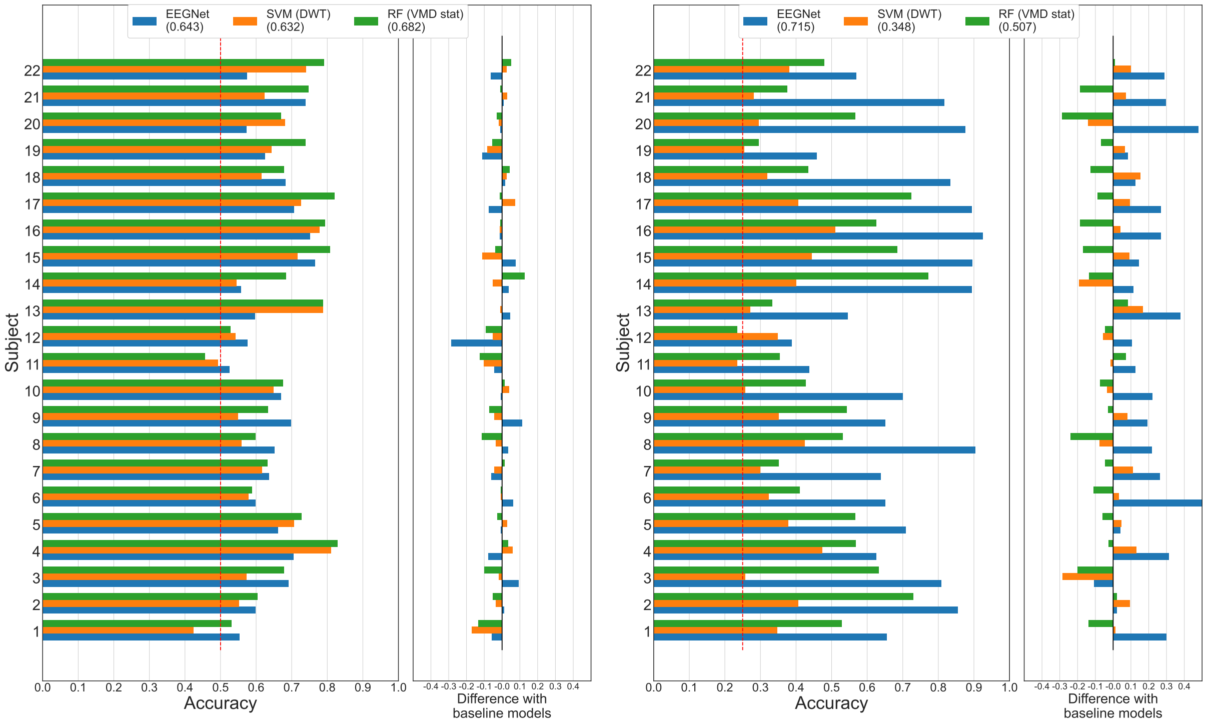
Figure B.2: Accuracies for individual **resting-state vs task** and **4-class** classification models for the experiment in section section 5.4. The figures show models created with EEGNet, SVM and RF when only **P1** data from the occipital lobe are used. The results are compared against the corresponding results in section 5.2. The average performance overall subjects of a classifier are written in parenthesis below the legend.



(a) Individual **Resting-state vs task** models, created with P1 data, dataset A.

(b) Individual **4-class** classification models, created with P1 data, dataset A.

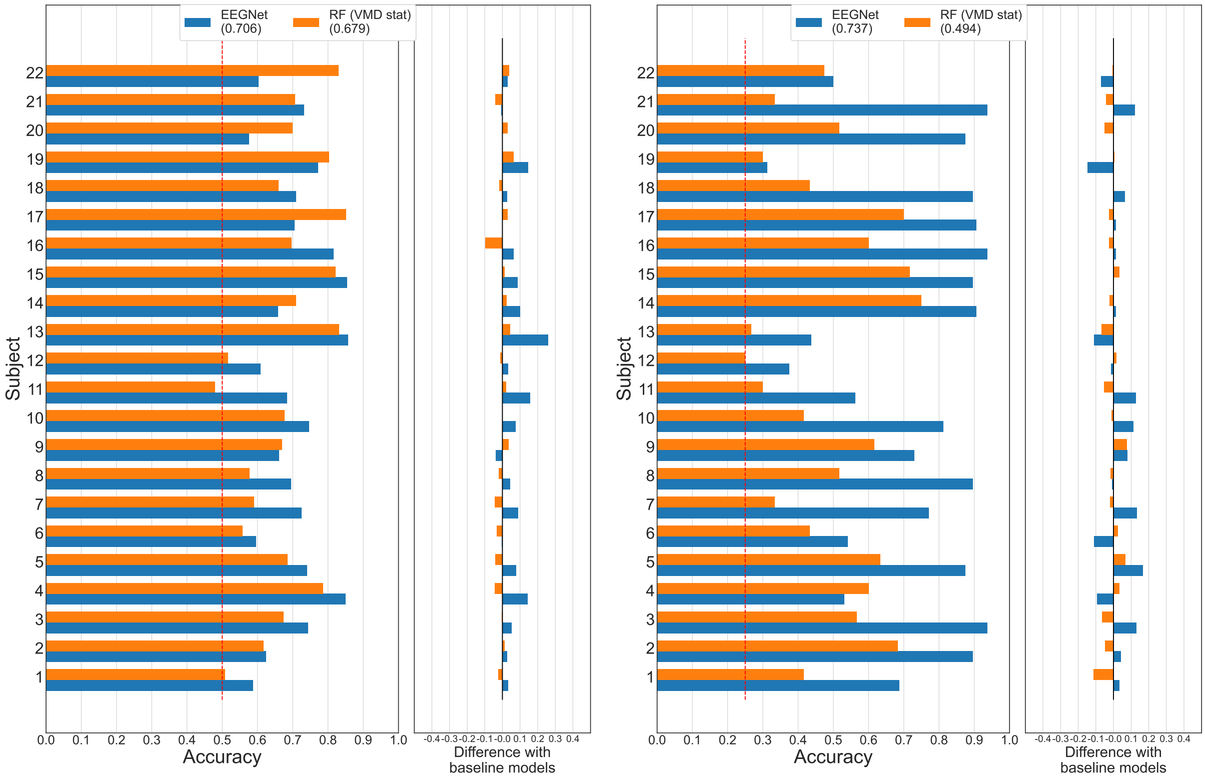
Figure B.3: Accuracies from the individual **resting-state vs task** and **4-class** classification models, from the experiment in section 5.7. The models are created with P1 data from dataset A. The left side plot displays the accuracies from SVM models, while the right side plot displays the accuracies created with RF presented. The legends show what processing technique was used to create the results, and the top bar graph in all plots displays the average accuracy for the methods.



(a) Individual **resting-state vs task** classification with P1 data, dataset A.

(b) Individual **4-class** classification with P1 data, dataset A.

Figure B.4: Results from creating LOOMs with dataset A and **P1** data for the experiments in section 5.8. The methods used are displayed in the legend, with their average accuracy calculated from all subjects written in parenthesis below the method name. Each figure includes two plots, where the left one is the accuracies created with LOOM and the right one compares the results with the previous best for that method. EEGNet and SVM are compared to their counterparts in section 5.5, while RF is compared to the models created with VMD stat in section 5.7.



(a) Individual **resting-state vs task** classification with P1 data, dataset A.

(b) Individual **4-class** classification with P1 data, dataset A.

Figure B.5: Results from using transfer learning and warm start on the EEGNet and RF models from section 5.8, for the experiments in section 5.9. The pre-trained LOOMs and the new models are trained with data from **P1**, dataset A. The methods used are displayed in the legend, with their average accuracy calculated from all subjects written in parenthesis below the method name. Each figure includes two plots, where the left one is the accuracies created with the new LOOM and the right one compares the results with the respective ones from section 5.8.

Appendix C

Tables with the results from the best methods

This appendix supplements some of the plots presented in the thesis. Only the results from the best-performing methods are presented. The **resting-state vs task** classification problem also presents the metric *specificity* that can be used to evaluate the models. Specificity explains how if the model predicts several false task states or not. High specificity means that the model does predict few to no false task states, while a low specificity says the opposite.

Table C.1: The table presents the results from each individual **resting-state vs task** model with P1 and P2 data from dataset A. *Acc.* is short for accuracy, while *Spec.* is short for specificity. Only the results from the best methods are displayed. All methods are pre-processed with a Notch and bandpass filter. EEGNet uses the pre-processed signal as input. SVM decomposes the signal with DWT and extracts energies and fractals from the components. RF decomposes the signal with VMD and extracts the modes' energies, fractals, and statistical features.

Subject	**EEGNet**				**SVM (DWT)**				**RF (VMD stat)**			
	P1		P2		P1		P2		P1		P2	
	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.
1	0.612	0.941	0.557	0.612	0.595	1.000	0.583	1.000	0.664	0.853	0.583	0.851
2	0.586	0.691	0.560	0.735	0.586	1.000	0.586	1.000	0.655	0.691	0.681	0.721
3	0.598	0.551	0.684	0.773	0.590	1.000	0.579	1.000	0.778	0.783	0.623	0.697
4	0.783	0.783	0.661	0.448	0.750	0.900	0.730	0.806	0.793	0.817	0.783	0.776
5	0.669	0.614	0.757	0.627	0.678	0.600	0.678	0.642	0.754	0.629	0.626	0.507
6	0.534	0.794	0.603	1.000	0.586	1.000	0.586	1.000	0.595	0.794	0.560	0.838
7	0.696	0.821	0.678	0.597	0.661	0.776	0.652	0.597	0.617	0.701	0.670	0.657
8	0.619	0.714	0.588	0.606	0.593	1.000	0.579	1.000	0.712	0.800	0.684	0.742
9	0.585	0.686	0.603	0.809	0.593	1.000	0.586	1.000	0.705	0.673	0.671	0.705
10	0.678	0.743	0.632	0.803	0.610	0.843	0.658	0.924	0.661	0.729	0.675	0.833
11	0.570	0.745	0.570	0.931	0.595	1.000	0.576	1.000	0.582	0.830	0.629	0.816
12	0.861	0.915	0.654	0.543	0.595	1.000	0.628	0.957	0.620	0.851	0.641	0.761
13	0.551	0.243	0.739	0.612	0.797	0.743	0.887	0.866	0.788	0.729	0.852	0.821
14	0.519	0.745	0.617	0.722	0.595	1.000	0.584	1.000	0.557	0.532	0.773	0.744
15	0.690	0.588	0.781	0.682	0.828	0.765	0.772	0.773	0.845	0.853	0.842	0.879
16	0.763	0.636	0.817	0.753	0.789	0.750	0.817	0.730	0.803	0.773	0.863	0.820
17	0.782	0.739	0.656	0.822	0.654	0.522	0.604	0.522	0.833	0.783	0.760	0.744
18	0.667	0.449	0.669	0.586	0.590	1.000	0.593	1.000	0.636	0.556	0.621	0.679
19	0.735	0.609	0.711	0.545	0.726	0.899	0.763	0.955	0.793	0.833	0.867	0.918
20	0.584	0.338	0.635	0.448	0.699	0.554	0.670	0.642	0.699	0.692	0.696	0.657
21	0.731	0.690	0.791	0.672	0.597	1.000	0.583	1.000	0.756	0.696	0.814	0.720
22	0.636	0.467	0.684	0.692	0.714	0.756	0.665	0.736	0.740	0.778	0.761	0.758
Average	0.657	0.659	0.666	0.683	0.656	0.868	0.653	0.870	0.709	0.744	0.736	0.757

Table C.2: The table presents the results from each individual **4-class** classification model with P1 and P2 data from dataset A. *Acc.* is short for accuracy. Only the results from the best methods are displayed. All methods are pre-processed with a Notch and bandpass filter. EEGNet uses the pre-processed signal as input. SVM decomposes the signal with DWT and extracts energies and fractals from the components. RF decomposes the signal with VMD and extracts the modes' energies, fractals, and statistical features.

subject	**EEGNet**		**SVM (DWT)**		**RF (VMD stat)**	
	P1 acc.	P2 acc.	P1 acc.	P2 acc.	P1 acc.	P2 acc.
1	0.354	0.688	0.333	0.250	0.667	0.542
2	0.833	0.854	0.313	0.583	0.708	0.938
3	0.917	0.833	0.542	0.479	0.833	0.833
4	0.313	0.500	0.344	0.375	0.594	0.646
5	0.667	0.583	0.333	0.333	0.625	0.688
6	0.146	0.375	0.292	0.188	0.521	0.271
7	0.375	0.458	0.188	0.375	0.396	0.667
8	0.688	0.750	0.500	0.438	0.771	0.833
9	0.458	0.250	0.271	0.167	0.571	0.531
10	0.479	0.417	0.292	0.167	0.500	0.438
11	0.313	0.484	0.250	0.250	0.281	0.438
12	0.281	0.281	0.406	0.469	0.281	0.375
13	0.167	0.229	0.104	0.271	0.250	0.313
14	0.781	0.859	0.594	0.406	0.906	0.891
15	0.750	0.958	0.354	0.563	0.854	0.917
16	0.656	0.875	0.469	0.422	0.813	0.797
17	0.625	0.891	0.313	0.484	0.813	0.813
18	0.708	0.604	0.167	0.250	0.563	0.769
19	0.375	0.313	0.188	0.229	0.364	0.205
20	0.396	0.438	0.438	0.208	0.854	0.521
21	0.521	0.813	0.208	0.229	0.563	0.444
22	0.281	0.250	0.281	0.281	0.469	0.641
Average	0.511	0.572	0.326	0.341	0.597	0.698

Table C.3: The table presents each subject's pre-trained general **resting-state vs task** model and its results with P1 and P2 data from dataset A from section 5.11. *Acc.* is short for accuracy, while *Spec.* is short for specificity. Only the results from the best methods are displayed. All methods are pre-processed with a Notch and bandpass filter. EEGNet uses the pre-processed signal as input. RF decomposes the signal with VMD and extracts the modes' energies, fractals, and statistical features.

Subject	EEGNet				RF (VMD stat)			
	P1		P2		P1		P2	
	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.	Acc.	Spec.
1	0.647	0.897	0.565	0.552	0.414	0.044	0.426	0.015
2	0.621	0.691	0.716	0.750	0.509	0.176	0.534	0.206
3	0.624	0.667	0.711	0.667	0.590	0.333	0.535	0.197
4	0.576	0.367	0.800	0.806	0.728	0.633	0.774	0.642
5	0.729	0.557	0.713	0.597	0.669	0.486	0.704	0.493
6	0.603	0.912	0.621	1.000	0.552	0.368	0.569	0.382
7	0.635	0.597	0.730	0.672	0.617	0.627	0.722	0.627
8	0.669	0.743	0.684	0.803	0.508	0.186	0.491	0.152
9	0.686	0.771	0.698	0.868	0.614	0.385	0.592	0.318
10	0.720	0.814	0.544	0.333	0.678	0.914	0.632	0.864
11	0.608	0.617	0.510	0.253	0.430	0.043	0.444	0.046
12	0.557	0.936	0.615	0.652	0.506	0.234	0.538	0.217
13	0.475	0.129	0.826	0.821	0.619	0.357	0.609	0.343
14	0.620	0.809	0.747	0.778	0.582	0.298	0.571	0.278
15	0.784	0.706	0.754	0.652	0.664	0.456	0.658	0.485
16	0.776	0.750	0.915	0.888	0.737	0.591	0.752	0.618
17	0.769	0.739	0.682	0.911	0.654	0.457	0.636	0.411
18	0.667	0.435	0.602	0.586	0.714	0.511	0.600	0.411
19	0.752	0.652	0.711	0.545	0.802	0.909	0.857	0.967
20	0.575	0.677	0.713	0.687	0.646	0.431	0.643	0.493
21	0.748	0.704	0.774	0.657	0.679	0.500	0.826	0.700
22	0.662	0.444	0.729	0.747	0.519	0.222	0.477	0.165
Average	0.659	0.664	0.698	0.692	0.611	0.416	0.618	0.410

Table C.4: The table presents each subject's pre-trained general **4-class** classification model and its results with P1 and P2 data from dataset A from section 5.11. *Acc.* is short for accuracy. Only the results from the best methods are displayed. All methods are pre-processed with a Notch and bandpass filter. EEGNet uses the pre-processed signal as input. RF decomposes the signal with VMD and extracts the modes' energies, fractals, and statistical features.

subject	**EEGNet**		**RF (VMD stat)**	
	P1 acc.	P2 acc.	P1 acc.	P2 acc.
1	0.625	0.667	0.521	0.354
2	0.833	1.000	0.708	0.854
3	0.938	0.917	0.708	0.583
4	0.719	0.625	0.656	0.583
5	0.875	0.750	0.583	0.667
6	0.479	0.500	0.479	0.313
7	0.646	0.646	0.438	0.479
8	0.917	0.938	0.771	0.750
9	0.750	0.667	0.486	0.563
10	0.854	0.896	0.542	0.521
11	0.500	0.406	0.406	0.344
12	0.281	0.375	0.344	0.375
13	0.604	0.583	0.229	0.333
14	0.938	0.969	0.906	0.844
15	1.000	0.938	0.771	0.792
16	0.875	0.984	0.844	0.641
17	0.875	0.906	0.813	0.750
18	0.771	0.708	0.625	0.692
19	0.333	0.313	0.364	0.341
20	0.542	0.833	0.542	0.625
21	0.813	0.875	0.656	0.472
22	0.406	0.688	0.375	0.484
Average	0.712	0.739	0.583	0.572

Tobias Treider Moe

Master's thesis in Cybernetics and Robotics

NTNU
Norwegian University of
Science and Technology