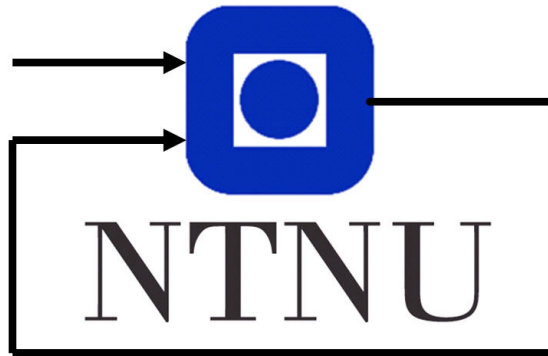




TTK4550  
ENGINEERING CYBERNETICS, SPECIALIZATION PROJECT

---

# Exploring Motor Imagery as a Brain-Computer Interface for aerial drones

---

*Arthur:* Arvin Das

*Supervisor:* Marta Molinas

*Co-supervisor:* Luis Alfredo Moctezuma Pascual

30.06.2025

---

# Abstract

Brain computer interfaces (BCIs) enable direct communication between neural activity in the brain and external devices. In this project, we investigate motor imagery (MI) as an EEG-based control paradigm for aerial drone operation. This report build on prior work carried out in the lab.

We first start by learning and reproducing the previous work. After that we check for how neuroplasticity can affect performance over a longer period of time. Then we explore new ideas such as LSTM models. Then we analyze how a time window can affect the performance. Lastly we compare our dataset based on OpenBCI against datasets based on Mentalab.

We worked with EEG data from two and three class MI paradigms (left vs right hand with rest). These raw data were preprocessed with a 50 Hz notch filter and a bandpass filter, common average re-referencing or Laplacian re-referencing, and features extracted via Common Spatial Patterns or Riemannian covariance methods. The classification experiments compared CSP + SVM, Riemannian-geometry classifiers, and novel LSTM networks. We achieved peak offline accuracies over 90% for two class tasks and over 80% for three class based on the performance of subject bc. Otherwise most subjects performed at random.

LSTM models underperformed classical pipelines and showed limited cross subject generalization with about 50% in external validation. Neuroplasticity analysis over 15 sessions revealed no systematic accuracy gains, while temporal window optimization showed subject specific optimal window lengths between 0.2 and 3.4 seconds.

These results showcase the challenges of robust, generalizable MI-BCI control, but also provide a scalable and user-friendly platform for future multi command drone demonstrations and neurorehabilitation research.

---

# Table of Contents

|                                                  |           |
|--------------------------------------------------|-----------|
| <b>Abstract</b>                                  | <b>i</b>  |
| <b>List of Figures</b>                           | <b>iv</b> |
| <b>List of Tables</b>                            | <b>v</b>  |
| <b>Abbreviations</b>                             | <b>vi</b> |
| <b>1 Introduction</b>                            | <b>1</b>  |
| 1.1 Background and Motivation . . . . .          | 1         |
| 1.2 Objective. . . . .                           | 1         |
| 1.3 Approach. . . . .                            | 2         |
| 1.4 outline. . . . .                             | 2         |
| 1.5 UN's Sustainable Development Goals . . . . . | 2         |
| <b>2 Theoretical background</b>                  | <b>4</b>  |
| 2.1 The Human brain . . . . .                    | 4         |
| 2.1.1 Neuro plasticity . . . . .                 | 5         |
| 2.1.2 Motor imagery . . . . .                    | 5         |
| 2.1.3 Brain computer interface, BCI . . . . .    | 7         |
| 2.1.4 Electroencephalography, EEG . . . . .      | 8         |
| 2.2 Preprocessing . . . . .                      | 8         |
| 2.2.1 Filters . . . . .                          | 8         |
| 2.2.2 Re-referencing . . . . .                   | 9         |
| 2.3 Feature extraction . . . . .                 | 10        |
| 2.3.1 Common Spatial Patterns, CSP . . . . .     | 10        |
| 2.4 Classification . . . . .                     | 11        |
| 2.4.1 Support vector machine, SVM . . . . .      | 12        |
| 2.4.2 Riemannian . . . . .                       | 12        |
| 2.4.3 Long short term memory, LSTM . . . . .     | 13        |

---

|          |                                                                                                          |           |
|----------|----------------------------------------------------------------------------------------------------------|-----------|
| <b>3</b> | <b>Literature review</b>                                                                                 | <b>16</b> |
| <b>4</b> | <b>Methodology</b>                                                                                       | <b>17</b> |
| 4.1      | Data acquisition . . . . .                                                                               | 17        |
| 4.2      | Tools . . . . .                                                                                          | 21        |
| 4.2.1    | IDUN . . . . .                                                                                           | 21        |
| 4.2.2    | Artificial intelligence, AI . . . . .                                                                    | 21        |
| <b>5</b> | <b>Results</b>                                                                                           | <b>22</b> |
| 5.1      | Recreating the results from [3]. . . . .                                                                 | 22        |
| 5.2      | Neuro plasticity. . . . .                                                                                | 23        |
| 5.3      | The LSTM models. . . . .                                                                                 | 26        |
| 5.4      | Train the LSTM on one group of subjects and evaluate it on a different<br>group (cross-subject). . . . . | 26        |
| 5.5      | Temporal windowing. . . . .                                                                              | 28        |
| 5.6      | Comparison with the apple catcher game. . . . .                                                          | 30        |
| <b>6</b> | <b>Discussion</b>                                                                                        | <b>33</b> |
| 6.1      | Neuro plasticity. . . . .                                                                                | 33        |
| 6.2      | The LSTM models. . . . .                                                                                 | 34        |
| 6.3      | Cross subject. . . . .                                                                                   | 34        |
| 6.4      | Temporal Windowing. . . . .                                                                              | 34        |
| 6.5      | Appel catcher game comparison. . . . .                                                                   | 35        |
| <b>7</b> | <b>Conclusion.</b>                                                                                       | <b>36</b> |
| 7.1      | Further work. . . . .                                                                                    | 36        |
|          | <b>Bibliography</b>                                                                                      | <b>38</b> |

---

---

## List of Figures

|    |                                                                                                              |    |
|----|--------------------------------------------------------------------------------------------------------------|----|
| 1  | The human brain anatomy . . . . .                                                                            | 5  |
| 2  | Cortical homunculus . . . . .                                                                                | 6  |
| 3  | Two types of BCI caps . . . . .                                                                              | 7  |
| 4  | SVM illustrated . . . . .                                                                                    | 12 |
| 5  | Riemannian illustration . . . . .                                                                            | 13 |
| 6  | LSTM cell . . . . .                                                                                          | 14 |
| 7  | Electrode layout . . . . .                                                                                   | 17 |
| 8  | The record protocol . . . . .                                                                                | 18 |
| 9  | Mentalab electrode setup . . . . .                                                                           | 19 |
| 10 | The Mentalab cap that were used . . . . .                                                                    | 20 |
| 11 | Accuracies and standard deviations over 15 session using the 2 and 3<br>class problem with CSPSVM . . . . .  | 24 |
| 12 | Accuracies and standard deviations over 15 session using the 2 and 3<br>class problem with RIEMANN . . . . . | 25 |
| 13 | Accuracy vs time window for subject bc, using CSPSVM on the 2<br>and 3 class problem. . . . .                | 28 |
| 14 | Accuracy vs time window for subject bc, using RIEMANN on the 2<br>and 3 class problem. . . . .               | 29 |
| 15 | Accuracy vs time window for subject zz, using CSPSVM on the 2 and<br>3 class problem. . . . .                | 29 |
| 16 | Accuracy vs time window for subject zz, using RIEMANN on the 2<br>and 3 class problem. . . . .               | 29 |
| 17 | Accuracy for subject 15. . . . .                                                                             | 30 |
| 18 | Accuracy for subject 22. . . . .                                                                             | 31 |
| 19 | Accuracy for subject 36. . . . .                                                                             | 31 |
| 20 | Accuracy for subject 38. . . . .                                                                             | 32 |
| 21 | Accuracy trend for subject 15. . . . .                                                                       | 33 |

---

## List of Tables

|   |                                                                       |    |
|---|-----------------------------------------------------------------------|----|
| 1 | EEG Frequency Bands . . . . .                                         | 7  |
| 2 | Participant demographics . . . . .                                    | 18 |
| 3 | Recreating the accuracies from Erlend and Kjersti . . . . .           | 22 |
| 4 | Hyperparameter grid used for tuning the LSTM model . . . . .          | 26 |
| 5 | Best hyper-parameter configurations per subject and task type . . . . | 26 |
| 6 | Best overall hyperparameter for the LSTM. . . . .                     | 27 |
| 7 | Accuracies for the external validation. . . . .                       | 27 |
| 8 | Accuracies from the source reconstruction pipeline. . . . .           | 35 |

---

## Abbreviations

- **BCI:** Brain computer interface
- **EEG:** Electroencephalogram
- **SVM:** Support vector machine
- **LSTM:** Long short-term memory
- **LR:** Logistic regression
- **LDA:** Linear Discriminant Analysis
- **MI:** MI Motor Imagery
- **FFT:** Fast Fourier Transform
- **TPR:** True positive rate
- **FPR:** False positive rate
- **ML:** Machine learning
- **AI:** Artificial intelligence
- **ERD:** Event-Related Desynchronization
- **ERS:** Event-Related Synchronization
- **SNR:** Signal to noise ratio
- **CAR:** Common Average Reference
- **LLM:** Large Language Models

---

# 1 Introduction

## 1.1 Background and Motivation

A big use case of technology is to inspire and invoke interest in people. To do this, our lab wants to create a drone show consisting of drones being controlled with one’s imagination, that is, controlling a drone with your mind. To make this possible, we first need a reliable way to decode brain signals and turn them into useful control commands. Furthermore, we plan to develop a plug-and-play system so that multiple users can step in, imagine a command, and fly a drone with their mind, no special setup or training required.

To capture and decode the brain signals, we use a brain–computer interface (BCI). A BCI is a technology that lets the brain’s electrical activity directly communicate with external devices, most commonly computers. In motor-imagery (MI) BCIs, users imagine limb movements such as left or right hand movements, which produces characteristic event-related desynchronization (ERD) and synchronization (ERS) patterns in the EEG signal. These patterns can be detected and classified both offline and online, for example to control a two dimensional drone interface.

One of BCI’s applications is linked with research on repairing human cognitive and sensorimotor functions, also called neurorehabilitation. MI-based BCIs are emerging as powerful tools for motor rehabilitation in patients with stroke, spinal cord injury, and other neurological conditions. MI BCIs ask users to mentally simulate limb movements while EEG sensors record their brain activity. By leveraging the brain’s neuroplasticity, these systems can “rehearse” motor pathways even when physical movement isn’t possible. Clinical studies show that imagined movements activate the same neural networks as actual motion, leading to measurable gains in motor-related brain regions and functional recovery, [1].

## 1.2 Objective.

The objective of this project is to investigate the use of motor imagery as a brain computer interface for aerial drones. This investigation has been done by previous students, but we wanted to further develop their work and introduce new ideas to enhance performance.

To do this, we tested previously developed pipelines and experimented with new ones, such as LSTM models. We then investigated how neuroplasticity affects our ability to predict MI tasks over a longer period of time, and how the time window influences the results. Furthermore, we experimented with external validation using our LSTM models. Finally, we compared results from our dataset—based on the OpenBCI cap with datasets generated using a Mentalab cap.

---

### 1.3 Approach.

In this project, I further developed the work of Daniel Nicolai Grønn’s specialization project, **Imagine and Fly: A Brain–Computer Interface for Motor Imagery Drone Control** [2], which is based on the master’s thesis of Kjersti Brynstad and Erlend Vatsvåg, **An Asynchronous Motor Imagery–based Brain–Computer Interface for Two-dimensional Drone Control** [3].

I began by learning the fundamentals of this field, which was completely new to me. To understand the current state of the research, I first recreated the previously obtained results. Then I developed a new pipeline based on Long Short-Term Memory (LSTM) neural networks. I tested this model on multiple subjects to tune the hyperparameters and finally applied it in a cross-subject evaluation. Next, I performed a time window analysis to determine the optimal window length for our dataset. Finally, I compared our OpenBCI cap datasets against datasets collected with a Mentalab cap.

In this report, I worked with both the two class and three class problems. The two class problem consists of MI tasks distinguishing right vs. left hand movement, while the three class problem adds a rest class (no movement). We focus primarily on the three class problem because it more accurately reflects real world use. In everyday scenarios, users don’t continuously imagine left or right hand movements there are natural pauses or “rest” periods where no intentional command is issued. By including a rest class, we capture these non-informative EEG segments and ensure our system can distinguish between active motor imagery and true inactivity.

The codebase for this project can be found at [https://github.com/wavesresearch/drone\\_demo/tree/LSTM](https://github.com/wavesresearch/drone_demo/tree/LSTM).

### 1.4 outline.

This report is organized as following: section 2 present the theoretical background needed to understand the worked carried out in this project. Section 3 shortly introduce works i have been inspired off. Section 4 describe how we acquired our data and which tools we have used throughout the project. Section 5 present the results we produced, while section 6 further discuss the results. Lastly section 7 concludes this report and showcase further development.

### 1.5 UN’s Sustainable Development Goals

This project contributes to the UN’s Sustainable Development Goals 3 and 4. Goal 3 states: “Ensure healthy lives and promote well-being for all at all ages.”. We support this through research on motor imagery (MI) and brain–computer interfaces (BCIs) for use in neurorehabilitation. Goal 4 states: “Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.” We address this by inspiring interest in technology among young people through exposure to BCIs and

---

drones.

---

## 2 Theoretical background

In this chapter, we'll cover the key theory you need to follow the rest of this report. We begin by exploring how the human brain generates the electrical activity captured in EEG recordings. Next, we'll discuss how we can preprocess the raw data and extract features from it. Finally, we'll introduce the specific classification methods used throughout the project.

### 2.1 The Human brain

To understand how BCI and EEG works, we first need to understand the human brain, how it functions and how we can collect and decrypt the EEG signals from it.

The human brain is the most complex organ we have and is responsible for all of our emotions and actions. It consists of nervous tissue and is responsible for all the processing of information from the sensory nervous system and the coordination of motor control [4].

The brain can be divided into three main parts, the cerebellum, the brainstem, and the cerebrum. The cerebellum's main purpose is to control voluntary muscle movements and to maintain balance and coordination. The brainstem connects cerebellum and cerebrum to the spinal cord, which is the communication channel between the body and the brain. The brainstem regulates bodily activities such as heart rhythm, breathing, blood flow, focusing vision and blinking. The cerebrum is the largest part of the brain and can be further divided into four parts which we can see in the figure below, 1. The four parts are the frontal lobe, parietal lobe, occipital lobe and the temporal lobe. The temporal lobe is mainly responsible for short term memory, speech and musical rhythm. The occipital lobe is at the back of the brain and is responsible for all the vision related tasks. The parietal lobe helps with identifying objects and understanding the spatial relationship between objects. Furthermore it is responsible for interpreting pain and touch throughout the body. Lastly we have the frontal lobe, which is responsible for the personality, problem solving and voluntary movement [5].

---

## Human Brain Anatomy

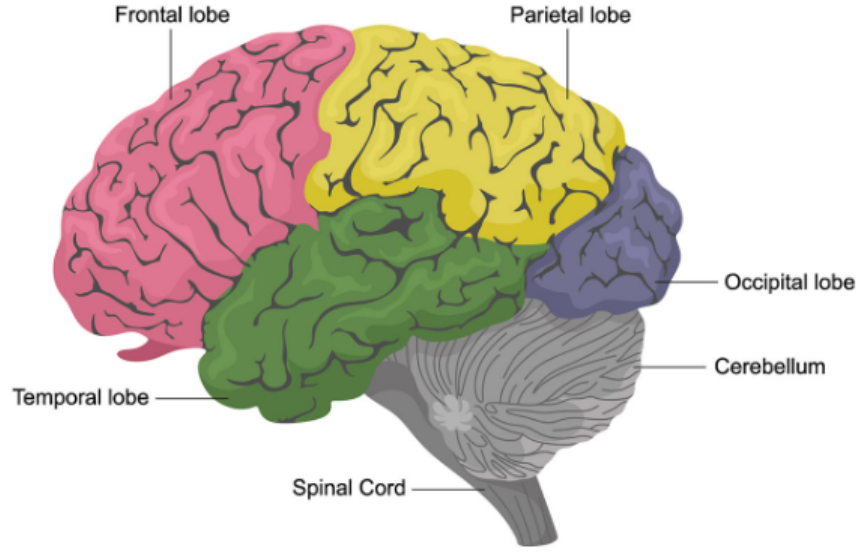


Figure 1: The human brain anatomy, taken from [5]

The brain is primarily made up of two types of cells, which are neurons and glial cells. These cells are responsible for sending information in the form of chemical and electrical signals throughout the body [5]. We will discuss the working of neurons in a more detailed manner in section 2.1.4.

We are especially interested in the parietal lobe, for MI tasks, because of its involvement in our voluntary movements.

### 2.1.1 Neuro plasticity

Neuro plasticity describes the brain’s capacity to reshape and rewire its neural circuits over time. Through this process, neural pathways grow and reorganize in response to new learning, environmental shifts, injury recovery, or compensation for sensory and cognitive changes. Whether it’s individual neurons forming fresh synapses or larger scale adjustments like cortical remapping and altered brain rhythms, neuroplasticity tells about the brain’s remarkable flexibility, even well into adulthood [6].

### 2.1.2 Motor imagery

Motor imagery (MI) is the mental process where we imagine a body movement without performing the movement in real life. MI is based on the mental stimulation theory, which states that cognitive motor processes like motor imagery share similar representations of brain activity as motor execution [7].

---

Alongside its widespread application as a neuro-paradigm in BCIs, this mental process has been studied for neurorehabilitation in stroke patients. The idea behind this is that stroke patients have lost their ability to physically move parts of their body, but the brain and nervous system are still intact. It has been proven that with the strengthening of the motor abilities through mental rehearsal, stroke patients have been able to regain some of their previous movements according to [1].

MI triggers particular alterations in brainwave patterns, specifically influencing the mu ( $\mu$ ) and beta ( $\beta$ ) frequency bands. The mu band, known as alpha activity, ranges from 8 to 12 Hz and the beta band spans from 12 to 30 Hz. The rest of the frequency bands can be seen in the table below 1.

These alternations are caused by event-related desynchronization (ERD) and event-related synchronization (ERS). ERD signifies a decrease in amplitude and attenuation of the frequency band, whereas ERS indicates an increase in amplitude. ERD and ERS follow the cortical homunculus, which can be seen in the figure bellow 2. This means that an imaginary movement in a specific limb will result in ERD's in the cortical area for that limb [8]. MI can also induce event-related synchronizations in non-focused cortical regions of the body, that is, in all motor cortex sections except those linked to the limb currently being visualized, [9].

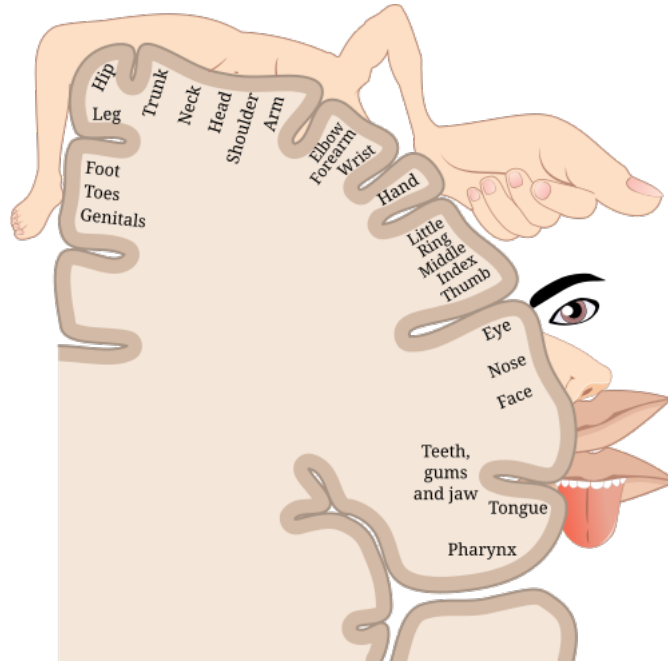


Figure 2: Cortical homunculus

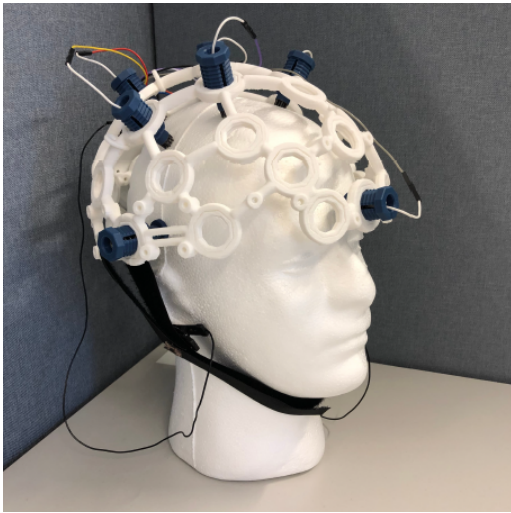
In this project, we work extensively with MI tasks based on left and right hand movements. For that reason, we are especially interested in the parietal lobe, and more specifically in the primary motor cortex and the premotor cortex. This is reflected in the way we place the electrodes on our headset to collect data. Specifically, electrodes can be placed at positions corresponding to each limb in the motor cortical homunculus to map ERD/ERS patterns to imagined movements of different limbs. This brings us to the next chapter, the BCI

Table 1: EEG Frequency Bands

| Frequency band     | Frequencies |
|--------------------|-------------|
| Delta ( $\delta$ ) | 0.5–4 Hz    |
| Theta ( $\theta$ ) | 4–8 Hz      |
| Alpha ( $\alpha$ ) | 8–12 Hz     |
| Beta ( $\beta$ )   | 12–35 Hz    |
| Gamma ( $\gamma$ ) | > 35 Hz     |
| Mu ( $\mu$ )       | 8–13 Hz     |

### 2.1.3 Brain computer interface, BCI

A brain computer interface (BCI) is a technology that lets the brain’s electrical activity directly communicate with external devices, most commonly with computers. The brain’s electrical activity can be detected using both invasive methods such as Microelectrode arrays (MEA) and non-invasive methods such as magnetoencephalography (MEG) and electroencephalography (EEG). EEG signals can be collected using wet or dry electrodes. Examples of different types of BCI caps can be seen in the picture below.



(a) Cap from OpenBCI taken from [3]



(b) Cap from OpenBCI taken from Mentalabs webpage

Figure 3: Two types of BCI caps

Users often need sufficient training with BCI systems to ensure good performance, but according to [10]:

One of the biggest challenges in BCI research is to understand and solve

---

the problem of “BCI Illiteracy”, which is that BCI control does not work for a non-negligible portion of users (estimated 15% to 30%).

this means that no matter how much we train users, for about 15% to 30% of them we will only get gibberish out, which can be a major issue we want to make an application that we can provide for a wide range of people.

In our case, our BCI utilizes EEG to capture information from the brain.

#### **2.1.4 Electroencephalography, EEG**

As stated, neurons communicate with the use of electrical and chemical processes. Electroencephalography (EEG) detects these electrical signals with the help of electrodes placed on the scalp. Although the electrical signal of a single neuron is too weak to be detected, in EEG’s we actually see a summation of multiple neurons that are active at the same time.

These electrical signals are produced by a phenomenon called the action potential. An action potential is a rapid, all or none voltage spike that travels along a neuron’s membrane by flipping its inside outside charge difference. At rest, the membrane potential sits at about  $-70$  mV relative to the outside. Once incoming signals depolarize the membrane past a threshold, voltage gated  $\text{Na}^+$  ions channels open and  $\text{Na}^+$  ions rushes in, driving the inside up to roughly  $+30$  mV (and locally making the exterior correspondingly more negative). Almost immediately, those  $\text{Na}^+$  ions channels inactivate and voltage-gated  $\text{K}^+$  channels open, letting  $\text{K}^+$  flow out and repolarizing the membrane back toward  $-70$  mV. Finally, the  $\text{Na}^+/\text{K}^+$  pumps restore the original ion gradients, resetting the inside to  $-70$  mV relative to the outside. It is the summation of this voltage changes among multiple neurons that we are able to detect with EEG [11].

## **2.2 Preprocessing**

The raw EEG data are often corrupted by noise and artifacts, because of this we often want to preprocess our data before extracting features and putting it into our classifier.

### **2.2.1 Filters**

Filtering is a process that removes unwanted components from our signal. Although filtering removes unwanted components, it will also suppress and remove actual information in our signals. Filtering in signal processing often revolves around attenuating specific frequencies in the frequency domain. [12].

The most common forms of filtering are lowpass, highpass and bandpass filtering. Highpass filtering attenuates frequencies below the given frequency, while higher frequencies pass. Lowpass filter are the opposite of the highpass filter, so it let

---

lower frequency pass while is attenuates higher frequencies. Band pass filters are a combination of the low and high pass filter. It lets frequencies between two different specified limits pass, while frequencies under the lower limit and higher then the upper limit get attenuated. Band stop filters are just opposite of the band pass filter.

We also apply a notch filter to eliminate power-line interference. A notch filter is simply a band stop filter with a very narrow stopband. Since the European powerline frequencies are at 50 Hz, we use a 50 Hz notch filter in our EEG preprocessing.

The formula for the band pass 1 and band stop 2 filters are as following. Let  $\omega_1$  and  $\omega_2$  be the lower and upper cutoff frequencies.

$$H_{BP}(s) = \frac{(\omega_2 - \omega_1) s}{s^2 + (\omega_2 - \omega_1) s + \omega_1 \omega_2}. \quad (1)$$

$$H_{BS}(s) = \frac{s^2 + \omega_1 \omega_2}{s^2 + (\omega_2 - \omega_1) s + \omega_1 \omega_2}. \quad (2)$$

### 2.2.2 Re-referencing

When recording EEG, our goal is to capture the local electrical signal at each electrode without interference from distant sources or noise. Because every measurement is relative to a chosen reference, the position and behavior of that reference electrode shape all our signals. Ideally, the reference would be placed at a neutral electrical position, but in practice the scalp's conductivity makes it impossible. As a result, every channel reflects both the brain's activity at its site and whatever the reference sees [13].

To reduce this dependency and to improve signal-to-noise ratio researchers often replace the physical reference with a computed or virtual one during preprocessing. Two of the most widely used schemes are the Common Average Reference (CAR), which subtracts the mean across all electrodes, and Laplacian re-referencing, which subtracts a distance weighted average of each electrode's immediate neighbors. Both approaches have been shown to enhance detection motor imagery related ERD/ERS and to outperform simpler ear references in BCI applications [13].

The **Laplacian re-referencing** scheme is computed by subtracting the weighted average of each of the neighboring electrode from the current electrode. When computing the Laplacian for a certain electrode, the surrounding electrodes are weighted according to their distance to the electrode currently being considered. The goal of this referencing is to highlight localized activity, because by subtracting from the neighbor, activity that is shared among electrodes is canceled out [14]. The formula for Laplacian re-referencing is as following:

---


$$V_i^{\text{LAP}} = V_i^{\text{ER}} - \sum_{j \in S_i} g_{ij} V_j^{\text{ER}} \quad (3)$$

$$g_{ij} = \frac{\frac{1}{d_{ij}}}{\sum_{k \in S_i} \frac{1}{d_{ik}}} \quad , \quad \sum_{j \in S_i} g_{ij} = 1 \quad (4)$$

$V_i^{\text{ER}}$  is the voltage between electrode  $i$  and the original reference, while  $V_i^{\text{LAP}}$  is the new voltage over electrode  $i$  after the re-referencing.  $S_i$  is the subset containing the surrounding electrode of the  $i^{\text{th}}$  electrode and  $d_{ik}$  is a weight between electrode  $i$  and  $j$  which tells us about the correlation between them.

The **common average reference (CAR)** scheme works in a similar manner to the Laplacian re-referencing, but instead of only looking at the nearest neighboring electrodes, CAR looks at all the electrodes in the headset and subtract the average in the  $i^{\text{th}}$  electrode [14]. The formula of CAR looks like this:

$$V_i^{\text{CAR}} = V_i^{\text{ER}} - \bar{x} \quad , \quad \bar{x} = \frac{1}{n} \sum_{j=1}^n V_j^{\text{ER}} \quad (5)$$

Here,  $n$  is the number of electrodes in our headset and  $V_i^{\text{CAR}}$  is the voltage over electrode  $i$  after applying the CAR scheme.

## 2.3 Feature extraction

Feature extraction is a fundamental step in machine learning that transforms raw data into a set of compact and informative features. This reduces the resources that requires to process the data while retaining relevant information for our classifier to solve the problem.

In many domains such as signal processing the datasets can contain a large amount of measurements, many of which add little useful detail. Feature extraction addresses this by breaking the data down to its essence.

This is beneficial because it lowers the computational cost, by focusing only on the most information and reducing the data size and complexity. Furthermore it enhanced model accuracy. Well chosen features often reveal the underlying structure of the problem, enabling classifiers to learn more effectively, [15].

### 2.3.1 Common Spatial Patterns, CSP

The Common Spatial Pattern (CSP) is an algorithm which calculates spatial filters for ERD and ERS. CSP is widely used for feature extraction on EEG based BCI signals based on MI tasks.

---

Given  $K$  trials for class  $c$ , each trial produces an  $N \times T$  data matrix  $S_i^c$  with  $N$  channels and  $T$  time samples, assumed to be zero mean over time. The goal of CSP is to find an  $N \times M$  filter matrix  $W$ . This filter matrix  $W$  works as a linear transformation which works according to:

$$s_{\text{CSP}}(t) = W^T s(t) \quad (6)$$

Where  $s(t)$  is the signal at time  $t$ . To find the filter matrix we first need to estimate the class conditional covariances as following.

$$R_c = \frac{1}{K} \sum_{i=1}^K S_i^c (S_i^c)^T, \quad (7)$$

Furthermore, the  $W$  matrix need to satisfy the following equation:

$$W^T R_1 W = \Lambda_1, \quad W^T R_2 W = \Lambda_2, \quad \Lambda_1 + \Lambda_2 = I, \quad (8)$$

This is for a binary problems, that is  $c \in \{1, 2\}$ .

Here each  $\Lambda_c$  is diagonal. Equivalently, one solves the eigenvalue problem, as following:

$$R_1 w = \lambda R_2 w \quad (9)$$

To obtain eigenvectors  $w_j$  (the  $j^{\text{th}}$  rows of  $W$ ).

The eigenvectors  $w_j$  that satisfy the equation above form the columns of  $W$  and represent the spatial filters in the CSP.

Once we solve the CSP eigenvalue problem, we end up with many possible spatial filters each corresponding to a particular eigenvalue. The filters linked to the highest and lowest eigenvalues stand out because they produce signals whose variance is most different between, in this case, two mental-imagery tasks. In plain terms, these two filters pull out the brain activity patterns that change dramatically in one condition but stay relatively flat in the other, making it easier for a computer to tell the two states apart. This theory and formulas was taken from [16].

Although originally formulated for binary tasks, CSP has been extended to multi-class scenarios. In this project, we employ the multiclass CSP.

## 2.4 Classification

Classification is a technique that teaches machines to assign items to classes by learning from labeled examples. Classifiers are trained on data that represent the

---

real world as closely as possible. After classifiers are trained on labeled datasets, they can take new data as input and the output will be a prediction of which class the input belongs to.

How efficient a classifier is determined by a number of metrics like accuracy, precision, recall, and F1-score [17].

There exist a wide range of classification algorithms, among some of them that have been used in this project are, Support Vector Machine (SVM), Riemannian classification and Long Short Term Memory (LSTM).

#### 2.4.1 Support vector machine, SVM

Support Vector Machines (SVMs) are supervised learning algorithms that seek an optimal decision boundary called a **hyperplane** to separate data points of different classes. The best hyperplane is the one that maximizes the margin, that is, the distance between the closest points of each class and the hyperplane itself. Those closest points are known as the **support vectors**. To handle non linear data, SVMs use a **kernel** function to map inputs into a higher-dimensional space where a linear separation becomes possible. The picture below will try to illustrate the workings of a SVM, [18].

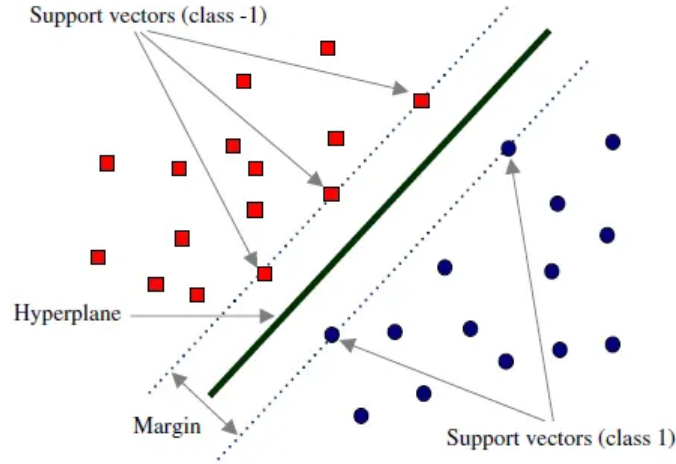


Figure 4: SVM illustration, taken from [18]

#### 2.4.2 Riemannian

In recent years, Riemannian geometry based approaches have emerged as a promising method for feature extraction and classification. Unlike methods that treat feature extraction, spatial filtering, and classification as separate stages, Riemannian techniques unify all three under a single geometric framework [19].

The shortest path between two covariance points  $P_1$  and  $P_2$  is given by:

$$\gamma(t) = P_1^{\frac{1}{2}} \left( P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}} \right)^t P_1^{\frac{1}{2}} \quad (10)$$

Its length is the *Riemannian distance*, and is given by

$$\delta_R(P_1, P_2) = \|\log(P_1^{-1} P_2)\|_F. \quad (11)$$

Between the points  $P_1$  and  $P_2$ , the Riemannian mean of  $m$  covariance matrices  $P_1$  to  $P_m$  can be calculated as following:

$$\mathcal{G}(P_1, \dots, P_m) = \arg \min_{P \in P(n)} \sum_{i=1}^m \delta_R^2(P, P_i) \quad (12)$$

In Riemannian geometry approaches, each EEG epoch is viewed as a point on the manifold of covariance matrices, with the assumption that trials from the same mental state cluster closely together. Traditional techniques like CSP, by contrast, treat these covariance matrices as if they lie in a flat Euclidean space, yet they actually lie on the curved manifold of Symmetric Positive-Definite (SPD) matrices, whose intrinsic geometry is Riemannian. Riemannian methods explicitly account for this curvature when comparing and averaging covariances.

The idea behind this is that, the spatial layout and power distribution of EEG sources remain roughly constant—information that is captured by the covariance structure. In practice, we apply a required geometric feature transform before feeding data into a classifier. Depending on the workflow, these steps can be arranged to perform classification in the tangent space (Euclidean) domain.

More information about the Riemannian approach can be found [19]. An illustration of how this approach works can be found at figure 5

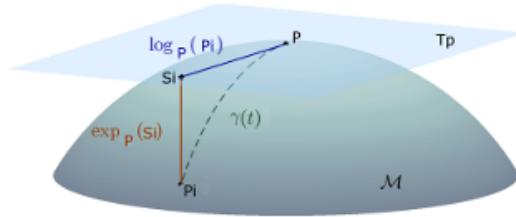


Figure 5: The tangent space at point  $F$  on manifold  $M$ , showing the tangent vector  $S_i$  at  $F$  and the *geodesic*  $\gamma(t)$  connecting  $P$  to  $P_i$ . Picture taken from [19]

### 2.4.3 Long short term memory, LSTM

Long Short-Term Memory (LSTM) networks are a special kind of recurrent neural network designed to overcome the vanishing-gradient problem and learn long range dependencies in sequential data. Historically LSTM networks has been used in areas like natural language processing, speech recognition and for machine translation. A picture of a typical LSTM cell can be found in the picture blow (6).

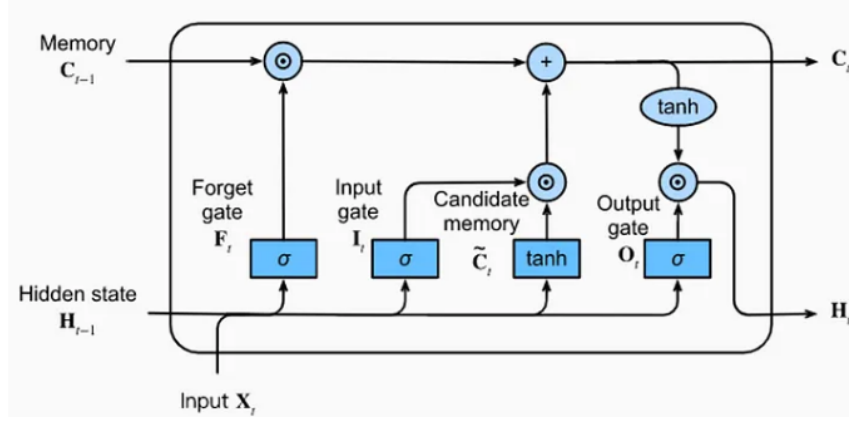


Figure 6: The LSTM cell architecture taken from [20]

LSTM cells consists of four feedforward neural networks. These neural networks contains an input layer and an output layer. Within these networks, each input neuron is linked to every output neuron. In reality, the LSTM unit consists of four fully connected layers.

Three out of the four feedforward neural networks are tasked with information selection. This are the forget gate, the input gate, and the output gate. These gates execute the usual memory management tasks. The forget gate removes information from memory, the input gate adds new information, and the output gate accesses existing information in memory.

An LSTM unit takes in three vectors as input, two comes from the LSTM itself, which were produced during the previous time step ( $t - 1$ ), which is the cell state (C) and the hidden state (H), and a third vector (X) coming from the input vector, which is introduced to the LSTM at time  $t$ . By utilizing these input vectors (C, H, X), the LSTM uses its gates to manage internal information flow, modifying the values of the cell state and hidden state vectors. These adjusted vectors will then serve as inputs for the time step ( $t + 1$ ).

At any time  $t$ , an LSTM takes an input vector  $X_{[t]}$ . In addition, it obtains the hidden state  $H_{[t-1]}$  and cell state  $C_{[t-1]}$  vectors generated in the preceding moment ( $t - 1$ ). The initial task performed by the LSTM unit is controlled by the **forget gate**. This gate determines, using the  $X_{[t]}$  and  $H_{[t-1]}$  vectors, which information should be removed from the cell state vector from time  $t - 1$ .

After removing information from the the incoming cell state ( $C_{[t-1]}$ ), we can introduce new data. This process is done by two independent neural networks, the **candidate memory and the input gate**. They both receive inputs in the form of vectors,  $X_{[t]}$  and  $H_{[t-1]}$ , which are combined into a single vector. The role of the candidate memory is to produce a candidate vector, which contains information proposed to be integrated into the cell state.

The **output gate** regulates the value of the hidden state generated by the LSTM at the current time step  $t$ , which is then passed as input to the LSTM at the time step  $t + 1$ .

---

The control of information flow is such that the cell state functions as a long-term memory, while the hidden state operates as a short-term memory. The hidden state established at time  $t$  also serves as the LSTM unit's output at time  $t$ , providing the output required for executing a given task, in our case the classification task. For our classification task we are only interested in the last output from all the LSTM cells.

Although LSTM has mostly been used for LLM and machine translations, there has been research in the use of LSTM to make classifications and predictions on MI tasks based on EEG. One such study with promising results was conducted by professor Elnaz Banan Sadeghian and Mr Navid Ayoobi and can be found here [\[21\]](#).

---

### 3 Literature review

As stated, this thesis builds on Daniel Nicolai Grønn's specialization project and the master's thesis of Kjersti Brynestad and Erlend Vatsvåg.

Kjersti Brynestad and Erlend Vatsvåg provided a codebase foundation by developing much of the functionality from signal preprocessing, pipelines and initial classification algorithms. Furthermore they also designed the complete data acquisition pipeline for the OpenBCI headset, which has been for great use in this project.

Daniel Nicolai Grønn then refined their codebase, improving its readability and overall structure. His implementation of a graphical user interface made the system far more accessible. It is worth noting that because of the use of IDUN, which is the super computer for NTNU, the GUI created by Daniel Grønn has become somewhat obsolete for the training of some models. This is because IDUN does not support the usage of GUI's in its interface. But the GUI has been greatly appreciated in other parts such as running the recording protocol, evaluating stored models, running the real life drone demos and connecting and communicating with the drones. Furthermore, I also greatly value the insightful discussions we've shared over the past months.

Much of our LSTM methodology is drawn from the work of Professor Elnaz Banan Sadeghian and Mr Navid Ayoobi. Their results show that an LSTM based approach performs well for MI task classification across different subjects. This success inspired us to adapt and extend their pipeline to our own dataset and research goals.

We plan to investigate how altering the temporal window used to extract EEG data segments for training and testing affects model performance. This idea is drawn from Erlend Skredsvig's specialization thesis, [22]. Which demonstrated that widening or narrowing the data window can have a dramatic impact on classifier accuracy.

---

## 4 Methodology

In this chapter we will explain the data acquisition process for this project. Then we will present the tools used throughout the project.

### 4.1 Data acquisition

We have primarily worked with data acquisition using the OpenBCI Ultracortex Mark IV headset with a CYTON board, to capture EEG at 250 Hz from up to eight Ag–AgCl dry comb electrodes, plus earlobe reference and ground, all streamed over Bluetooth. The headset can be viewed in figure 3a. The layout of the eight electrodes can be seen in the figure below 7. To ensure that each electrode had a good connection to the scalp the OpenBCI GUI was used. After connecting to your board you can press on CYTON Signals to check for the impedance on each channel.

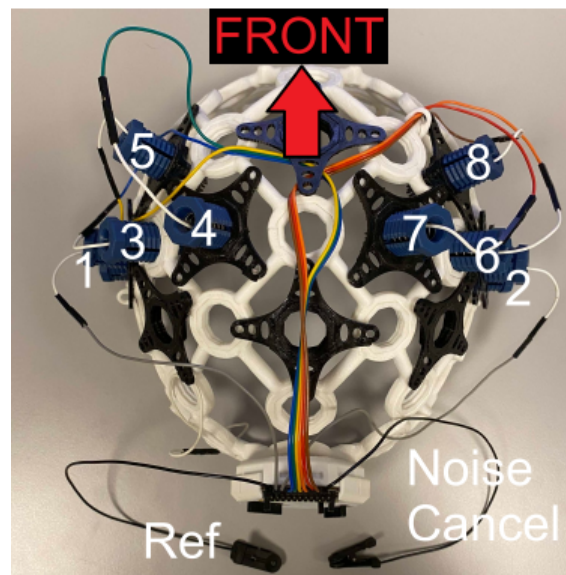


Figure 7: Electrode layout for the OpenBCI headset taken from [2]

To record the actual data, the protocol recorder developed by Kjersti Brynstad and Erlend Vatsvåg [3] was used. This protocol consisted of 2 arrows, one to the left and one to the right and a fixation point in the middle of the screen. During each trial, subjects sat facing a screen. Trials began with the cross alone for 0.5s, then one arrow turned red for 2s to cue a specific motor imagery task (e.g. left-hand or right-hand), and finally all arrows returned to black while the subject rested for 5s before the next trial. A picture of the recorder protocol can be found below 8.

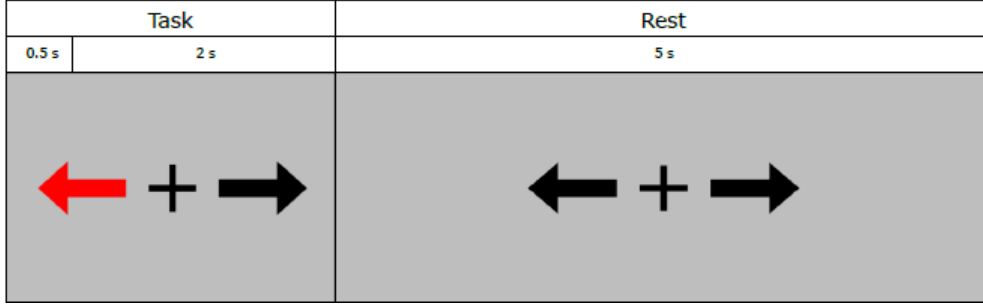


Figure 8: The record protocol, taken from [3]

Our dataset consisted of 17 participants, where 16 of the participants data was collected by [3]. The last participant was collected by me, and it is 15 sessions of the recorder protocol on my self. A table of the participants can be found at the table below 2.

Table 2: Participant demographics

| Subject | id | Sex | Age | Laterality | Sessions |
|---------|----|-----|-----|------------|----------|
| 1       | sp | F   | 23  | R          | 4        |
| 2       | pb | F   | 25  | R          | 4        |
| 3       | xd | F   | 23  | R          | 4        |
| 4       | kz | F   | 25  | R          | 4        |
| 5       | bq | M   | 25  | R          | 4        |
| 6       | hw | F   | 24  | L          | 4        |
| 7       | bw | F   | 24  | R          | 4        |
| 8       | rs | M   | 24  | L          | 4        |
| 9       | zz | F   | 23  | L          | 4        |
| 10      | wt | F   | 24  | R          | 4        |
| 11      | qs | M   | 27  | R          | 4        |
| 12      | kc | F   | 23  | R          | 4        |
| 13      | ll | M   | 27  | R          | 4        |
| 14      | dd | M   | 24  | R          | 4        |
| 15      | bc | M   | 23  | R          | 4        |
| 16      | gb | M   | 26  | R          | 5        |
| 17 (me) | tq | M   | 23  | R          | 15       |

During this project we also used datasets gathered by Erlend Skredsvig [22]. We used his datasets to compare how the model perform when trained and tested on different datasets. His datasets were produced using a 32-channel Mentalab Explore cap at 250 Hz. The headset used Ag/AgCl wet electrodes with a combined reference and ground placed on the right earlobe. Before starting the recording he ensured the impedance was under 20 K $\Omega$  at each channel. The data was streamed over Bluetooth using the Lab Streaming Layer (LSL). The layout of the electrodes can be shown in the figure below 9. The actual cap can be seen in figure 10. The dataset we got from the apple catcher protocol consists of four participants two of them performed over average and the remaining 2 were average.

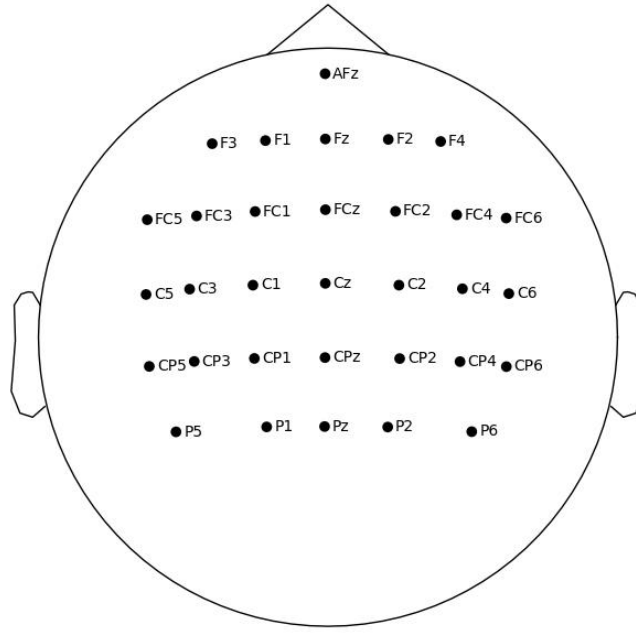
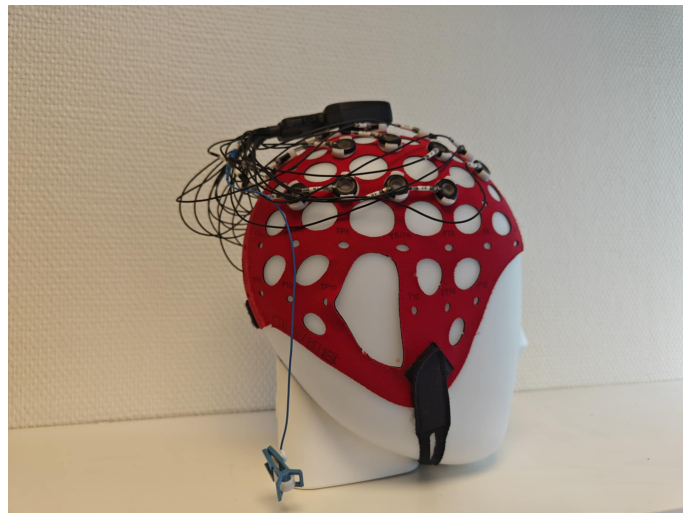


Figure 9: Mentalab electrode setup, taken from [\[22\]](#)



(a) Mentalab cap from the front



(b) Mentalab cap from the side

Figure 10: The Mentalab cap that were used

---

## 4.2 Tools

During this project a number of tools were used. I have used *Visual Studio Code* for the development environment. *Python* was the main language used throughout the project, and i have used *bash* to automate some parts of the project.

### 4.2.1 IDUN

We got access to IDUN during this project. IDUN is a cluster of high computational computers which provides a high performance compute platform for NTNU. More information about IDUN can be found [here](#). IDUN was mainly used for the training of different LSTM models.

### 4.2.2 Artificial intelligence, AI

Through out this project i have been using different artificial intelligence (AI) tools. I have used ChatGPT and google Gemini to facilitate me on produsing the code.

Although the work remains my own, the AI tools supported me with brainstorming, solution exploration, and the organization of ideas. ChatGPT was especially valuable for coding. It helped me with implementing and understanding various machine learning algorithms and generate scripts. ChatGPT were also used to troubleshoot and clarify code. I have used ChatGPT for troubleshooting and debugging for when i first started out with IDUN.

For the writing process, i drafted each section manually and then used ChatGPT to fine read and correct spelling errors. Its suggestions for phrasing, synonyms, and translations were used after review to ensure precision and coherence.

It it worth noting that AI tools were only use to help me fulfill my ideas, and that all ideas was produced by me and my supervisor.

---

## 5 Results

In this chapter I will showcase the results I gathered throughout the project.

### 5.1 Recreating the results from [3].

To make sure Erlend and Kjersti's original code was in working condition and to get familiar with the codebase, we wanted to recreated the results they reported in their master's thesis [3].

Table 3 below shows the accuracies for different sessions, participants, and pipelines. The first pipeline uses CSP for feature extraction followed by SVM for classification. The second uses Riemannian classification. We chose this 2 pipelines because they showed the most promising overall accuracies among the majority of the subjects. For both pipelines we applied a notch filter, a small highpass filter at 0.1 Hz, a bandpass filter from 8 Hz to 12 Hz, and a common average reference. The dataset is of the *task\_vs\_task* type that is, left hand vs. right hand tasks. We show the results for four different subject.

When we compare our numbers with those of Erlend and Kjersti, they match closely, with only a few outliers. These outliers may come from randomness in our training and testing of models, therefore we consider this recreation a success.

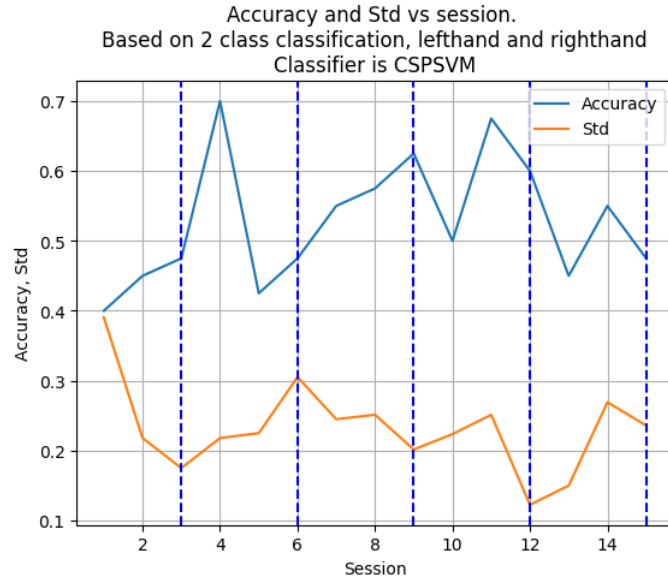
Table 3: Recreating the accuracies from Erlend and Kjersti

| Subject | Classifier | Session 1 | Session 2 | Session 3 | Session 4 |
|---------|------------|-----------|-----------|-----------|-----------|
| bc      | CSPSVM     | 0.717     | 0.717     | 0.933     | 0.933     |
| bc      | RIEMANN    | 0.700     | 0.767     | 0.933     | 0.917     |
| gb      | CSPSVM     | 0.667     | 0.700     | 0.650     | 0.650     |
| gb      | RIEMANN    | 0.667     | 0.667     | 0.600     | 0.617     |
| hw      | CSPSVM     | 0.533     | 0.633     | 0.733     | 0.783     |
| hw      | RIEMANN    | 0.567     | 0.650     | 0.767     | 0.767     |
| pb      | CSPSVM     | 0.867     | 0.750     | 0.767     | 0.750     |
| pb      | RIEMANN    | 0.900     | 0.800     | 0.800     | 0.833     |
| sp      | CSPSVM     | 0.600     | 0.750     | 0.617     | 0.733     |
| sp      | RIEMANN    | 0.650     | 0.750     | 0.600     | 0.650     |
| wt      | CSPSVM     | 0.700     | 0.800     | 0.733     | 0.667     |
| wt      | RIEMANN    | 0.533     | 0.750     | 0.717     | 0.617     |

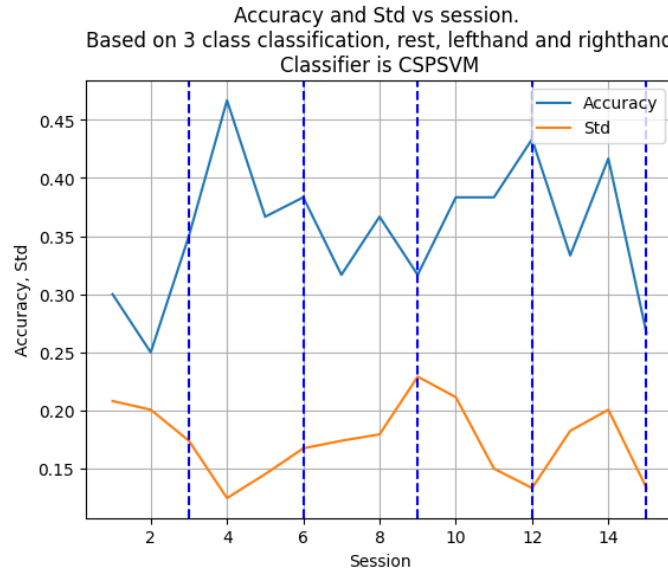
---

## 5.2 Neuro plasticity.

The plots below, [11](#) and [12](#) show how the classifier accuracy based on CSP+SVM and RIEMANNIAN methods fluctuate during fifteen session on one subject. The pipeline is identical to that described in section 5.1. Here we showcase four different cases; Two use a two class problem, which is righthand task vs. lefthand task. The other two use a three class setup, which is rest, lefthand task and righthand task. The dataset used here was fifteen sessions on myself, recording three sessions each day for five days over a span of three weeks. The purpose of the plots is to visualize possible neuroplastic changes throughout the fifteen sessions of recordings.

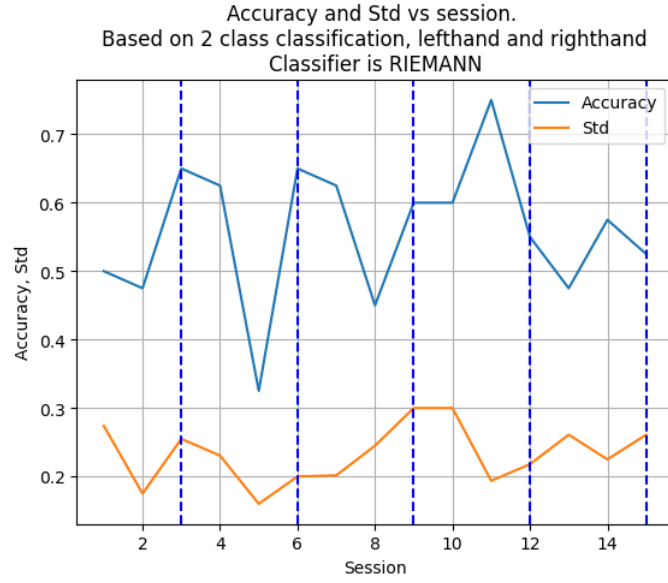


(a) Accuracies and standard deviations over 15 session using the 2 class problem with CSPSVM

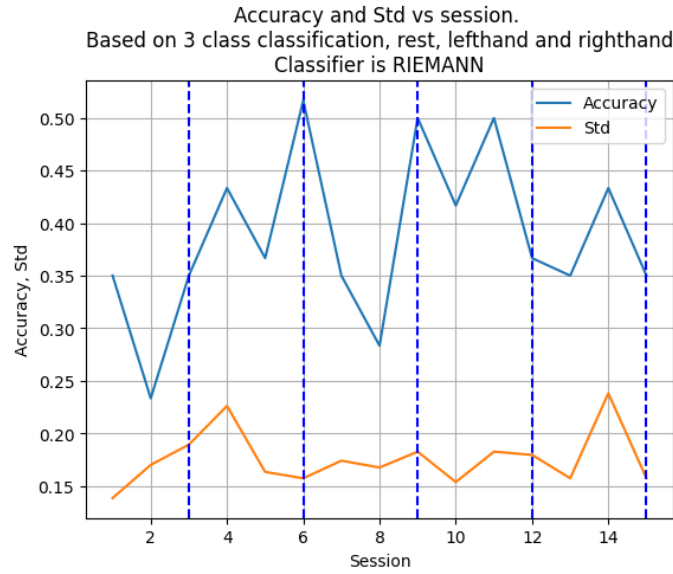


(b) Accuracies and standard deviations over 15 session using the 3 class problem with CSPSVM

Figure 11: Accuracies and standard deviations over 15 session using the 2 and 3 class problem with CSPSVM



(a) Accuracies and standard deviations over 15 session using the 2 class problem with RIEMANN



(b) Accuracies and standard deviations over 15 session using the 3 class problem with RIEMANN

Figure 12: Accuracies and standard deviations over 15 session using the 2 and 3 class problem with RIEMANN

---

### 5.3 The LSTM models.

We wanted to test out a new classification method in this lab, which is a recursive neural network based on LSTM (Long Short Term memory). This approach was based on the idea of Professor Elnaz Banan Sadeghian and Mr. Navid Ayoobi and can be found at [21]. For these experiments we used a bandpass filter on 6 to 13 hz, a notch filter and a Laplacian re-referencing, as stated in the paper.

This was the first time we used LSTM in the lab so we needed to do more preliminary testing of this approach. To do this we wanted to tune and find the best hyperparameters for the model. We did this by searching through the different combinations of the table below, 4.

The result of the best hyperparameters for each subject can be found in the table below, 5. Here we show the best hyperparameter for four different subjects with both the two class problem (right hand vs left hand) and the three class problem (right hand vs left hand vs rest).

Table 4: Hyperparameter grid used for tuning the LSTM model

| Hyperparameter   | Values              |
|------------------|---------------------|
| Hidden size      | 128, 256, 512, 1024 |
| Number of layers | 2, 4, 8, 10         |
| Dropout          | 0.1, 0.3            |
| Batch size       | 8, 16, 32           |
| Epochs           | 10, 20              |

Table 5: Best hyper-parameter configurations per subject and task type

| Subject | Task         | Acc.   | Hidden | Layers | Dropout | Batch | Epochs |
|---------|--------------|--------|--------|--------|---------|-------|--------|
| hw      | task_vs_task | 0.6000 | 128    | 10     | 0.1     | 8     | 20     |
| hw      | all_vs_all   | 0.5111 | 512    | 4      | 0.1     | 32    | 20     |
| bc      | task_vs_task | 0.7667 | 512    | 8      | 0.1     | 32    | 10     |
| bc      | all_vs_all   | 0.4333 | 512    | 4      | 0.3     | 16    | 20     |
| bw      | task_vs_task | 0.6500 | 256    | 8      | 0.3     | 8     | 20     |
| bw      | all_vs_all   | 0.4778 | 512    | 2      | 0.3     | 16    | 10     |
| zz      | task_vs_task | 0.5833 | 128    | 8      | 0.1     | 32    | 10     |
| zz      | all_vs_all   | 0.4222 | 128    | 8      | 0.1     | 8     | 20     |

### 5.4 Train the LSTM on one group of subjects and evaluate it on a different group (cross-subject).

We wanted to investigate the external validation of our data using the LSTM model. This mean we train one model based on some subjects and test it using the dataset of other, different subjects. We made the LSTM model based on the optimal hyperparameters we found from the previous chapter. The hyperparameters can be found in the table below, 6. To perform the external validation we sorted out six

---

participants. Three of them were marked as above average, based on how good accuracies they scored with traditional machine learning techniques. This three were subject **bc**, **hw** and **pb**. The remaining three was marked average and were subject **bw**, **ll** and **zz**. We trained and tested different LSTM models on various subsets in of this group, and the groups were mixed thorougher. For this we used the two class (right vs left) dataset for all the subjects. The results from this can be found ar table 7.

Table 6: Best overall hyperparameter for the LSTM.

| <b>Hyperparameter</b> | <b>Value</b> |
|-----------------------|--------------|
| Hidden size           | 512          |
| Number of layers      | 8            |
| Dropout               | 0.1          |
| Batch size            | 32           |
| Epochs                | 20           |

Table 7: Accuracies for the external validation.

| Training subjects | Test subjects | Accuracies |
|-------------------|---------------|------------|
| bc, hw, pb        | bw, ll, zz    | 0.503      |
| bw, ll, zz        | bc, hw, pb    | 0.489      |
| bw, hw, bc        | pb, ll, zz    | 0.500      |
| pb, bw, zz        | bc, hw, ll    | 0.512      |

---

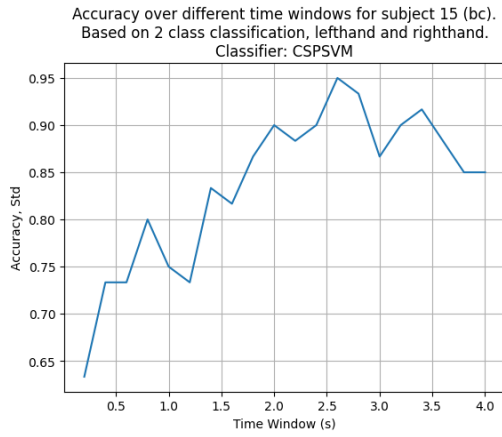
## 5.5 Temporal windowing.

Based on the work of Erlend Skredsvig's specialization thesis [22], we also wanted to investigate the effect the temporal window has on the classifiers.

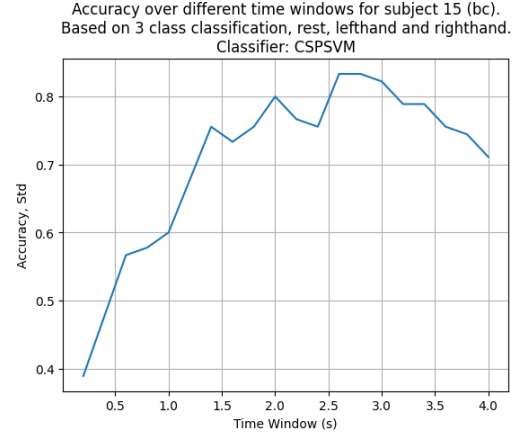
We ranged our window from 0.2 seconds to 4 seconds with a 0.2 second increment as seen in equation 13.

$$\text{time\_window} = [0.2, 0.4, 0.6, \dots, 3.8, 4.0] \quad (13)$$

The results was obtained with applying the time window to the datasets which altered the sample size. For this experiment we only used the last recorded session for each subject. We did this for two subjects, one above average (bc) and one average (zz). Furthermore we did this for both the two and three class problem and with both the CSPSVM and the RIEMANNIEN classifier. The results and description can be found in figure 13, 14, 15 and 16.

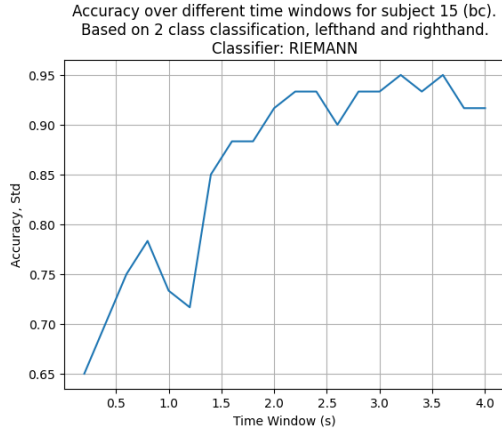


(a) Accuracy vs time window.

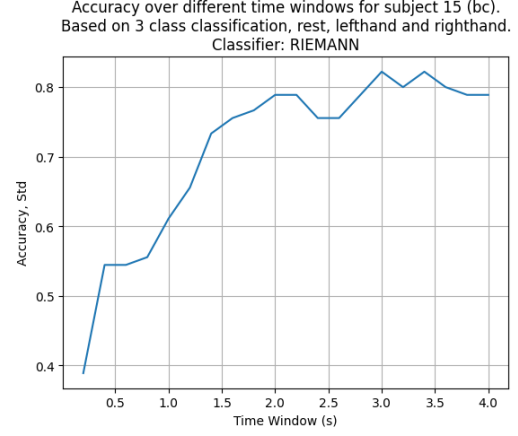


(b) Accuracy vs time window.

Figure 13: Accuracy vs time window for subject bc, using CSPSVM on the 2 and 3 class problem.

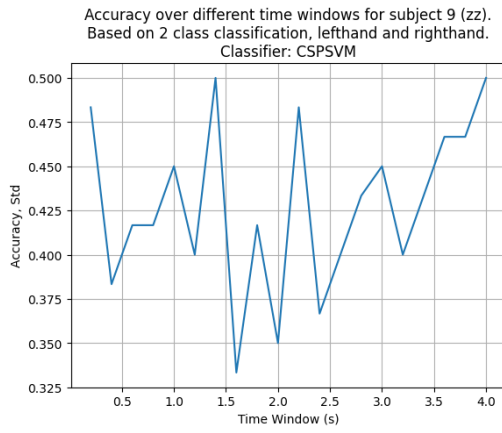


(a) Accuracy vs time window.

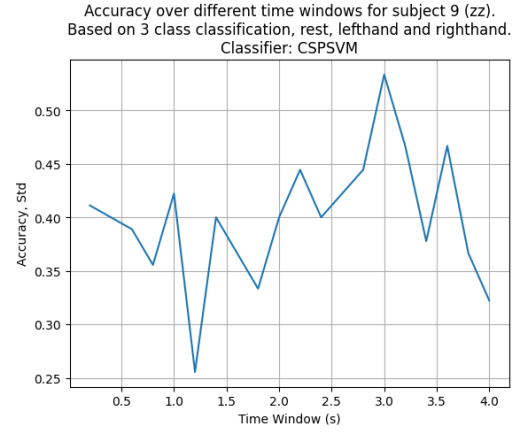


(b) Accuracy vs time window.

Figure 14: Accuracy vs time window for subject bc, using RIEMANN on the 2 and 3 class problem.

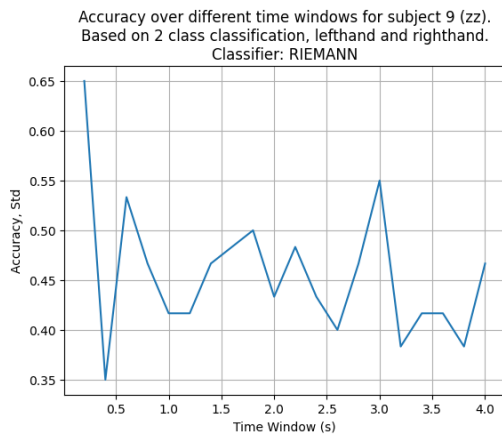


(a) Accuracy vs time window.

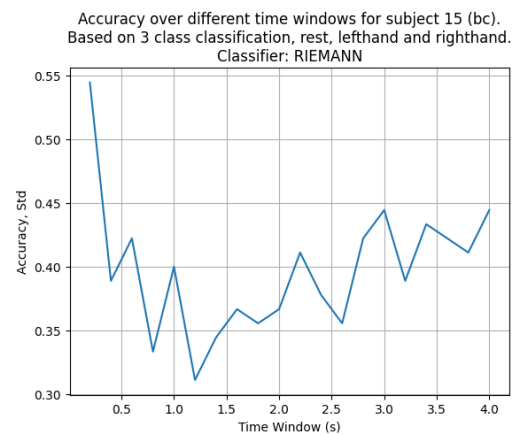


(b) Accuracy vs time window.

Figure 15: Accuracy vs time window for subject zz, using CSPSVM on the 2 and 3 class problem.



(a) Accuracy vs time window.



(b) Accuracy vs time window.

Figure 16: Accuracy vs time window for subject zz, using RIEMANN on the 2 and 3 class problem.

---

## 5.6 Comparison with the apple catcher game.

Lastly, we will present what the dataset collected with the Mentalab cap are able to achieve. These datasets were collected by Erlend Skredsvig [22] using his Apple Catcher game protocol. The dataset consists of four subjects where two of them performed above average (subject 15 and 22) and the remaining two were average (subject 36 and 38). These datasets are based on the two class problem, which is motor imagery tasks of the right hand vs the left hand. Here we can see four figures, 17, 18, 19 and 20, with different classifiers and their accuracies and standard deviations for each of the four subjects.

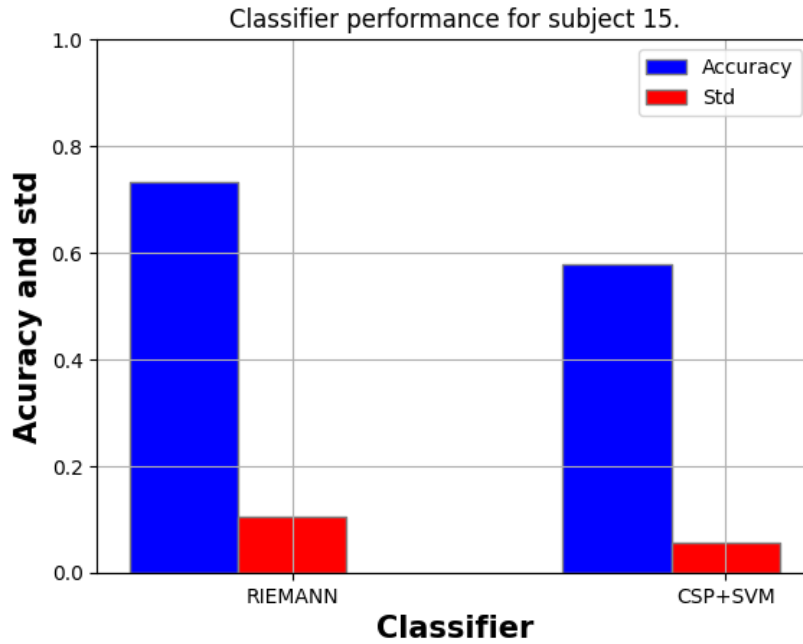


Figure 17: Different classifier accuracy and STD for subject 15.

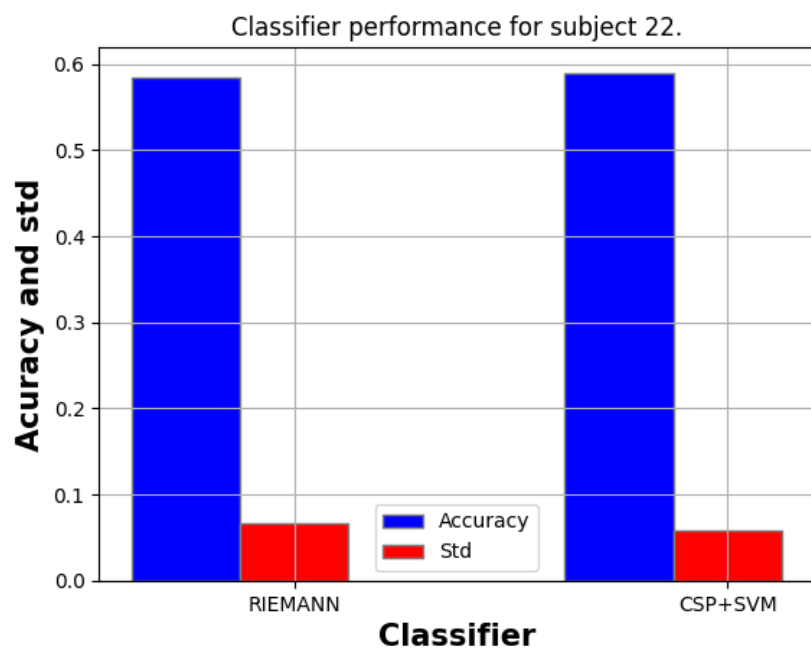


Figure 18: Different classifier accuracy and STD for subject 22.

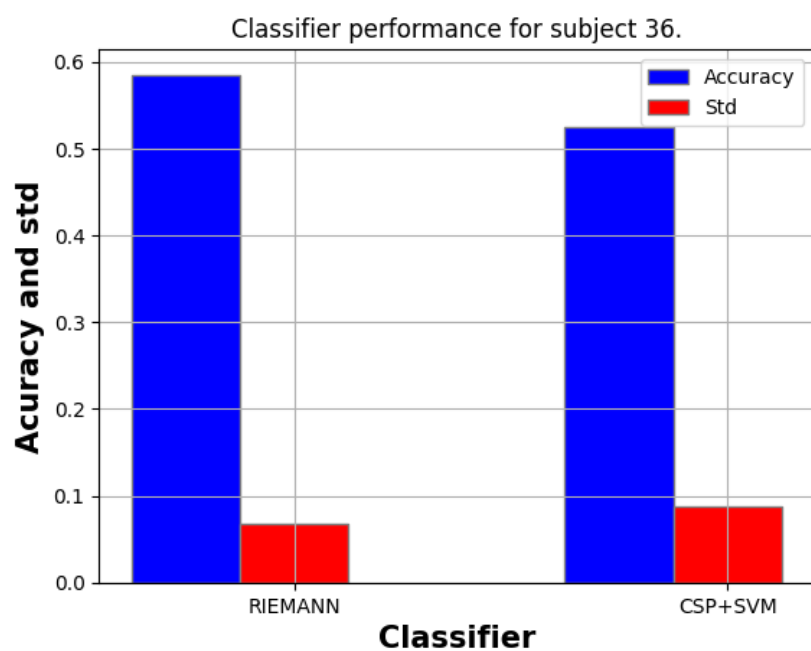


Figure 19: Different classifier accuracy and STD for subject 36.

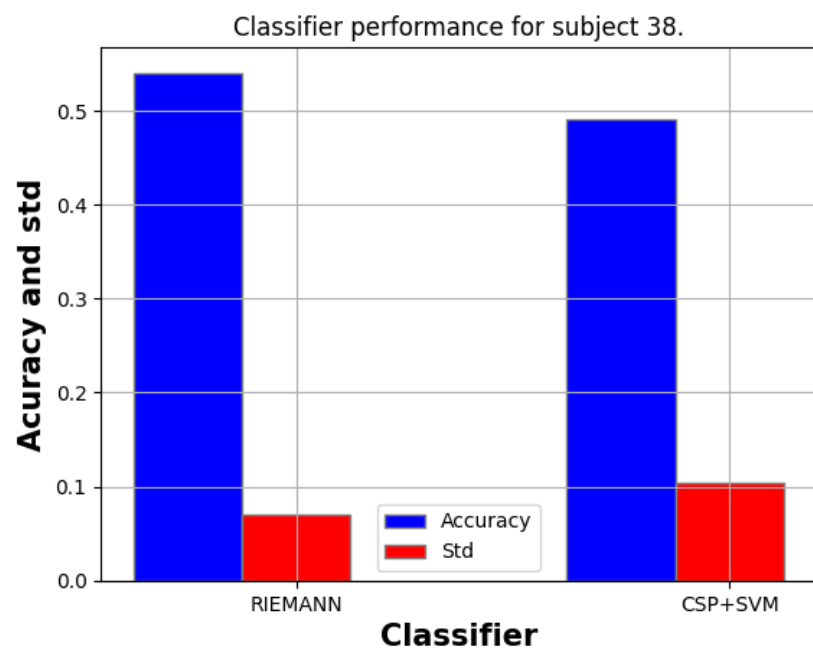


Figure 20: Different classifier accuracy and STD for subject 38.

---

## 6 Discussion

In this chapter I will discuss my findings from the result chapter.

### 6.1 Neuro plasticity.

During these experiments, we aimed to see accuracy rise session by session until it reached an optimal level, after which it would stabilize for the next sessions. We expected this effect based on articles like [23]. This is because after we perform the same task multiple times, the brain would "grow" new connections to perform better at those tasks. While analyzing Figure 11 and Figure 12, we can clearly see that this is not the case. For each session, we can see that the accuracies fluctuate, and that we never reach any level of stability.

We can see they fluctuate around 50% for the two-class problem and 33% for the three class problem, this means the classifier performance is random. This randomness can be due to BCI illiteracy, which we previously discussed, affects 15% to 30% of the population.

We can look at the accuracy trend for subject **bc** in figure 21. Here it clearly looks like the accuracies increase and stabilize after a few sessions. Ideally we should have done this neuroplasticity test on this subject, but unfortunately we only have access to these four recordings.

To better discuss the workings of neuroplasticity, we should have recorded more sessions for a larger number of subjects to better observe how the accuracies change over time.

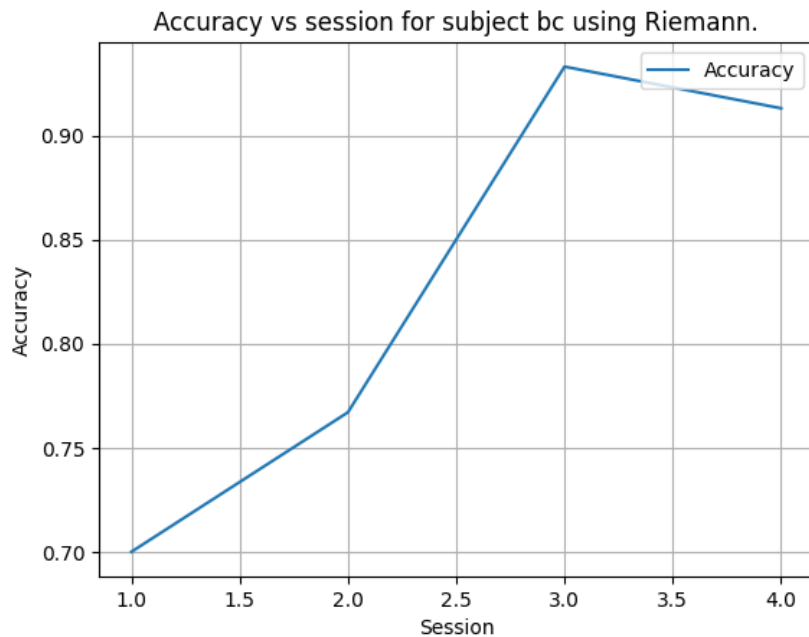


Figure 21: Accuracy trend for subject 15.

---

## 6.2 The LSTM models.

This experiment is very self explanatory but it is worth mentioning that we experimented with even higher values of the *Hidden size* and *Number of layers* but we quickly found out they made the performance worse while also having a very high computation time. Therefore we discarded them in the wider search of optimal parameters. The idea that a larger input size and more layers would worsen the performance is counterintuitive but we can see it happens in multiple different occasions based on [24]. The best hyperparameters can be found in table 6.

When we analyze the accuracies we can see that the performance is diminishing compared to the previous classifiers in table 3.

## 6.3 Cross subject.

As we can see in table 7, the accuracies cross subject were not very promising. Furthermore because this was based on the two class problem we can conclude this results to be random. This results make sense based on the article [25]. This article discuss how the brain works differently for every person. A direct quote from this article states:

“Like a person’s fingerprints, no two brains — not even those of identical twins — are exactly the same. Because of that, there’s no definition of “normal” capabilities for the human brain.”

Based on this it gives sense that models trained on one set of subjects and evaluated on a different set of subjects would give diminishing performance.

However the goal of using ML and AI models to predict MI tasks, so to make them learn a general pattern that occurs in the data. Based on previous studies this patterns such as ERD and ERS are phenomena that is present in every person. So this randomness may be conclude that our models has not generalized the data enough.

To make a stronger conclusion we could done a number of thing, such as making both the training and testing datasets consisting of good performing subjects, and the same for the average performing subjects. This would give us a better generalization based on how well subjects perform. We could also have used more subjects in this experiment, we had a pool of 16 subjects but we only experimented with six of them here.

## 6.4 Temporal Windowing.

As we can see in figure 13, 14, 15 and 16, different time windows will drastically affect the performance of the classifier. This affect is most noticeable for subject bc,

---

which already from the start performed above average. In figure 13 we can see the optimal time window for both the two class and three class problem is about 2.6 seconds when we are using the CSPSVM pipeline. For the Riemannian pipeline the optimal time window is between 3.2 - 3.4 seconds, 14.

For subject zz on the other hand it is not as trivial to draw a conclusion. In 15 and 16 we see that changing the time window can both increase and decrease the performance. For the CSPSVM classifier we get peaks at 1.4 second with the two class problem and 3 second with the 3 class problem. While for the Riemannian classifier we get peaks at 0.2 seconds for both the two and three class problem.

It looks like the optimal temporal windows is different based on each subject and on the different classifier we chose to use. To get a better estimate of how the temporal window affect the performance a wider set of subjects should be analyzed.

## 6.5 Appel catcher game comparison.

As a baseline, Erlend Skredsvig [22] got these results from his source reconstruction pipeline, 8.

Table 8: Accuracies from the source reconstruction pipeline.

| Subject | Accuracy |
|---------|----------|
| 15      | 0.90     |
| 22      | 0.70     |
| 36      | 0.50     |
| 38      | 0.58     |

From what we can see in figure 17 our Riemannian and CSPSVM pipelines got a worse performance then source reconstruction, but the results are better then random. Here the Riemannian pipeline performed better than the CSPSVM.

For subject 22 we see the same trend where our pipeline performed worse then the source reconstruction, 18.

For subject 36 we get a different result. Here our two pipelines performs better than the source reconstruction.

Subject 38 has the same trend, where our two pipeline are outperformed by the source reconstruction.

Moreover, the low standard deviations across all methods indicate stable classifier decisions (e.g.  $\sigma < 0.05$ ). Overall, methods exhibit consistent subject performance: subjects who perform well with one pipeline tend to do so with the others. Finally we see that the source reconstruction baseline is generally better than the other two.

---

## 7 Conclusion.

The objective of this project was to investigate the use of motor imagery as a brain computer interface for aerial drones. This investigation has been done by previous students but we wanted to further develop their work and bring in new ideas to enhance the performance.

To investigate this we have tested out perviously developed pipelines and experimented with new ones, such as LSTM models. We have checked for how neuroplasticity has an affect on how well we can predict MI tasks over time, and how a time window will effect the results. Furthermore we have experimented with external validation using our LSTM models. Lastly we compared the results from our dataset, based on the OpenBCI cap, against datasets generated using the mentalab cap.

Based on the table 3 and 5 we can see that the LSTM model did not perform as well as the classical machine learning protocols. The external validation of the LSTM model didn't do that well either. From table 7 we can see that the predictions from the classifier were random.

When investigating the effect of neuro plasticity we saw on figure 11 and 12 that the accuracies were fluctuating and did not diverge to a stable point. This can be caused by the phenomena of BCI illiteracy.

During the temporal window analysis we could clearly see that different window sizes effected the performance of the classifiers. The optimal window size was changing based on the subject and the classifier used.

Lastly we compared the dataset collected using the Apple Catcher protocol against our dataset. From this we saw that a pipeline using source reconstruction yield higher performance then the pipeline we have made for our datasets.

We are yet to find a classification pipeline that fits the general population and consistently gives predictions above random. This need to be investigated further.

### 7.1 Further work.

It is worth investigating the usage of the Mentalab cap combined with a pipeline using source reconstruction. Based on literature this approach have yielded higher performance then the current setup we are using.

Although the LSTM models did not perform as well as we hoped, it opened up the possibility to explore more deep neural networks in the future.

To make a stronger conclusion on the effect of neuroplasticity we should collect a larger amount of recordings on multiple subjects. Furthermore to find a better optimal time window that suits multiple subjects we should do the time window analysis on multiple subjects.

Automatic artifact detection and removal should also be investigated. We currently

---

record eye blink artifacts at the start of each recording session but we are not dealing with this information in any form.

I have already begun refactoring the code, but a deeper cleanup is still required. In its current state, the codebase is difficult to navigate. Function workflows are obscure, and many routines lack clear purpose. A more comprehensive reorganization breaking the pipeline into well-defined modules, adding descriptive documentation, and standardizing naming conventions will be necessary to make the code both maintainable and extensible.

---

## Bibliography

- [1] Wenzhe Liao et al. ‘Motor imagery brain–computer interface rehabilitation system enhances upper limb performance and improves brain activity in stroke patients: A clinical study’. In: *Frontiers in Human Neuroscience* 17 (2023). URL: <https://www.frontiersin.org/articles/10.3389/fnhum.2023.1117670/full>.
- [2] Daniel Nicolai Grønn. *Imagine and Fly: A Brain-Computer Interface for Motor Imagery Drone Control*. Specialization project in Cybernetics. Supervisor: Marta Molinas; Co-supervisor: Luis Alfredo Moctezuma Pascual. Norwegian University of Science and Technology, Department of Engineering Cybernetics, Dec. 2024.
- [3] Kjersti Brynstad and Erlend Vatsvåg. ‘An Asynchronous Motor Imagery-based Brain–Computer Interface for Two-dimensional Drone Control’. MA thesis. Norwegian University of Science and Technology, 2021.
- [4] Brain. *Brain* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 10/06/2025]. 2010. URL: <https://en.wikipedia.org/wiki/Brain>.
- [5] Johns Hopkins Medicine. *Brain Anatomy and How the Brain Works*. URL: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain> (visited on 11th June 2025).
- [6] Wikipedia. *Neuroplasticity*. <https://en.wikipedia.org/wiki/Neuroplasticity>. Retrieved June 13, 2025. 2025.
- [7] Anne L. Gordon and Annalisa di Maggio. ‘Rehabilitation for Children After Acquired Brain Injury: Current and Emerging Approaches’. In: *Pediatric Neurology* (2012), pp. 339–344. DOI: <https://doi.org/10.1016/j.pediatrneurol.2012.02.029>. URL: <https://www.sciencedirect.com/science/article/pii/S0887899412000835>.
- [8] Bernhard Graimann, Brendan Allison and Gert Pfurtscheller. ‘Brain–Computer Interfaces: A Gentle Introduction’. In: *Brain–Computer Interfaces*. Berlin, Heidelberg: Springer, 2009, pp. 1–27.
- [9] Gert Pfurtscheller et al. ‘The Graz Brain-Computer Interface’. English. In: *Brain-Computer Interfaces - Revolutionizing Human-Computer Interaction*. Springer, 2010, pp. 79–96.
- [10] Carmen Vidaurre and Benjamin Blankertz. ‘Towards a cure for BCI illiteracy’. In: *Brain Topography* (2010), pp. 194–198. URL: <https://doi.org/10.1007/s10548-009-0121-6>.
- [11] David Valentine. *Basic EEG electrophysiology*. <https://www.learningeeg.com/basic-eeg-electrophysiology>. Accessed: 2025-06-12. 2020.
- [12] Filter (signal processing). *Filter (signal processing)*. Last edited January 9, 2025. Retrieved June 12, 2025. URL: [https://en.wikipedia.org/wiki/Filter\\_\(signal\\_processing\)](https://en.wikipedia.org/wiki/Filter_(signal_processing)).
- [13] Ronald Gordon and Edward J. Rzempoluck and. ‘Introduction to Laplacian Montages’. In: *American Journal of Electroneurodiagnostic Technology* (2004), pp. 98–102. URL: <https://doi.org/10.1080/1086508X.2004.11079469>.

- 
- [14] George M. Parish, Sebastian Michelmann and Simon Hanslmayr. ‘How should I re-reference my intracranial EEG data?’ In: *Intracranial EEG: A guide for cognitive neuroscientists*. Ed. by N. Axmacher. Springer International Publishing, 2023, pp. 451–473. URL: [https://doi.org/10.1007/978-3-031-20910-9\\_28](https://doi.org/10.1007/978-3-031-20910-9_28).
- [15] GeeksforGeeks. *What is feature extraction?* <https://www.geeksforgeeks.org/machine-learning/what-is-feature-extraction/>. May 2025.
- [16] Rajesh P. N. Rao and Reinhold Scherer. ‘Statistical pattern recognition and machine learning in brain–computer interfaces’. In: *Statistical Signal Processing for Neuroscience and Neurotechnology*. Ed. by Karim Oweiss. Elsevier B.V., 2010, pp. 335–368.
- [17] GeeksforGeeks. *Getting started with classification*. <https://www.geeksforgeeks.org/machine-learning/getting-started-with-classification/>. 2025.
- [18] MLMath.io. *Math behind SVM (Support Vector Machine)*. <https://ankitnitjsr13.medium.com/math-behind-support-vector-machine-svm-5e7376d0ee4d>. 2019.
- [19] Alexandre Barachant et al. ‘Riemannian geometry applied to BCI classification’. In: *Latent Variable Analysis and Signal Separation (LVA/ICA 2010)*. Springer, Berlin, Heidelberg, 2010, pp. 629–636. DOI: [10.1007/978-3-642-15995-4\\_78](https://doi.org/10.1007/978-3-642-15995-4_78).
- [20] Ottavio Calzone. *An intuitive explanation of LSTM*. Medium. Apr. 2022. URL: <https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c>.
- [21] Navid Ayoobi and Elnaz Banan Sadeghian. ‘A self-paced BCI system with low latency for motor imagery onset detection based on time series prediction paradigm’. In: *arXiv preprint arXiv:2204.05450* (Apr. 2022). DOI: [10.48550/arXiv.2204.05450](https://doi.org/10.48550/arXiv.2204.05450). URL: <https://arxiv.org/abs/2204.05450>.
- [22] Erlend Skredsvig. *The Apple Catcher Game: A Novel Motor Imagery BCI for Hand Rehabilitation*. Specialization project. Norwegian University of Science and Technology, Department of Engineering Cybernetics, 2024.
- [23] Michael Rugnetta. *Cross-modal reassignment*. <https://www.britannica.com/science/neuroplasticity/Cross-modal-reassignment>. Encyclopedia Britannica. Retrieved June 15, 2025. n.d.
- [24] Data Science Stack Exchange. *External/internal validation performance?* [https://www.reddit.com/r/datascience/comments/1b00egf/externalinternal\\_validation\\_performance/](https://www.reddit.com/r/datascience/comments/1b00egf/externalinternal_validation_performance/). Retrieved June 15, 2025. Mar. 2013.
- [25] Cleveland Clinic. *Neurodivergent*. <https://my.clevelandclinic.org/health/symptoms/23154-neurodivergent>. Cleveland Clinic Health Library. Last reviewed June 2, 2022. Retrieved June 28, 2025. June 2022.
-