THINK & PLAY

CLASSIFYING LEFT AND RIGHT HAND MOTOR IMAGERY EEG SIGNALS BY USING DRY ELECTRODES FOR REAL-TIME BCI GAMING

HUNTER STERK

2022

16 June 2022

A thesis presented for the master Artificial Intelligence
at Utrecht University Faculty of Science

Hunter Sterk
Student number: 6981046
h.sterk@students.uu.nl

Research was supported by Sogeti (Vianen, Netherlands)

Under the supervision of
Dr. Leendert van Maanen (Utrecht University)
Dr. Martijn Mulder (Utrecht University)
Drs. Wouter de Nie (Sogeti)

Universiteit Utrecht

Abstract

Recent research on classifying motor imagery electroencephalography (EEG) signals has often shown that exceptionally high accuracy scores can be achieved. However, this is usually done with EEG caps with a lack of usability and by performing the classification on large time windows. When using a brain computer interface (BCI) for real-time gaming, it is important to keep the time windows as short as possible to minimise the response time in the game. The aim of this study is to investigate whether the performance of the most common and state-of-the-art methods for classifying motor imagery EEG signals is sufficient for real-time BCI gaming. The data is recorded with portable dry electrodes and is split into small windows of 512 ms.

Four different classifiers were evaluated: Logistic Regression (LR), Support Vector Machines (SVM), Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM). For the feature extraction, Common Spatial Patterns (CSP), Fast Fourier transform (FFT), Discrete Wavelet Transform (DWT) and statistical features are used. The LSTM model used another feature extraction method, in which each time window was broken down into eight segments and converted to features by applying a linear regression on each segment. The LSTM model outperformed the other models with a maximum accuracy score of 75.1%. The other three classifiers performed best with a combination of CSP, DWT and statistical features, but was not able to exceed a score of 57.6%.

When the LSTM model was applied on other datasets, the accuracy score dropped significantly. When using the model for BCI gaming, it is recommended to record the data and train the model right before each gaming session. With the methods used in this study, it was impossible to use a model that has been trained at a different moment or by another person. More research could be done into a technique like transfer learning, in order to reuse already trained models.

The conclusion of this study is that the performance achieved by the used methods is indeed much lower when used on data from dry electrodes with small window sizes. With proper mental tasks, it seems that a simple BCI game can be played with the evaluated models. But all the high results from recent studies give a distorted view of the possibilities for BCI gaming.

Contents

Abbreviations

BCI	Brain Computer Interface
CNN	Convolutional Neural Network
CSP	Common Spatial Patterns
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
EEG	Electroencephalography
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
LR	Logistic Regression
LSTM	Long Short-Term Memory
MI	Motor Imagery
ML	Machine Learning
SVM	Support Vector Machines
TCN	Temporal Convolutional Network

Introduction 1

Imagine controlling computers with your thoughts without using any physical movements. Something that is only possible in the far future or is it closer than we think? Communication between the human brain and computers has been a topic that intrigues researchers for a long time. The translation of brain signals to actions in computers or other devices is also known as a *brain computer interface* (BCI). The aim of this study is to evaluated the latest techniques and discover the opportunities and challenges in the field of BCI gaming.

In past years, many studies have been conducted on how BCIs can be used in the gaming industry [1]. By making it possible to control games with your thoughts, a whole new dimension will be introduced to today's gaming experience. The possibilities are endless. Besides gaming, there are many other fields in which BCIs can be beneficial. One example is letting paralysed people control their wheelchair by only using their mind [2]. For these reasons, the field of BCIs is a very interesting and promising research area.

This study was carried out within the company Sogeti (Vianen, The Netherlands), which started a new BCI project within their research and development department (SogetiLabs). Their goal is to build a proof-of-concept to inspire people within the company and to show clients their knowledge in the field of brain signal processing. In the end, they want to build a game – based on the classic game 'Breakout' from Atari – that can be played by just thinking left or right.

This study will focus on *non-invasive* BCIs, which means that the sensors are placed on the scalp and that surgery is not required. The measurement of electrical activity in the brain, is also known as *electroencephalography* (EEG). Within the field of non-invasive BCIs, there are two types of sensors that can be used for measurements: wet and dry electrodes. In the last decade, a lot of research has been carried out on EEG in combination with wet electrodes [3]. However, wet electrodes also have many disadvantages compared to dry electrodes [4]. Dry electrodes take much less time to prepare, are way more comfortable and can be taken anywhere. These will therefore be used in this study.

In previous research, *motor imagery* (MI) tasks were often used during the recording of the training data, because it provided the best results most of the time [5, 6]. MI tasks are mental tasks that are performed by the subjects during the EEG recording, which involve thinking about moving muscles, but not executing them. This resulted in exceptionally high accuracy scores. However, the problem with these studies are the large window sizes at which the predictions are made. When used for games, this means a higher response time, which reduces the gaming experience. Insufficient research has been done on the classification of small fragments of EEG data. This study will focus on the effect on the obtained accuracy when using dry electrodes in combination with a small window size.

This provides a better insight whether the current technologies are sufficient to allow one to play a simple video game by only using his mind.

The research question for this thesis will be as follows: How do the most common and state-of-the-art methods for classifying left and right hand motor imagery EEG signals perform using dry electrodes in combination with a small window size? And is this performance sufficient for playing a real-time BCI game? To answer the main questions, the following sub-questions must be addressed first:

1. What are the most important sensor locations for classifying motor imagery EEG data?
2. Which motor imagery tasks are most effective for predicting the participants actions?
3. What are the best feature extraction methods for motor imagery EEG data?
4. Which machine learning algorithm gives the highest accuracy at classifying motor imagery EEG data?

The most common and state-of-the-art methods are evaluated. The following methods are used for feature extraction: Common Spatial Patterns (CSP), Fast Fourier transform (FFT), Discrete Wavelet Transform (DWT) and statistical features are used. The following methods are used as classifiers: Logistic Regression (LR), Support Vector Machines (SVM), Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM). The LSTM model used another feature extraction method, based on the proposed approach of Wang et al. [7]. To optimise the used methods, a grid search is performed in order to find the optimal sensor selection, mental tasks, combination of methods and the hyperparameters.

This thesis starts with some background information of BCIs and EEGs in chapter 2. The third chapter is concerned with the methodology used for this study and describes the different methods used for preprocessing, feature extraction and classification. Chapter 4 gives an overview of all the findings of this study. The results, limitations and future research are discussed in chapter 5. The thesis ends with a brief conclusion in which the results and recommendations are summarised.

Literature review 2

This section provides a literature review within the relevant fields of this. It starts with a background about brain computer interfaces and electroencephalography, followed by its current applications. Next, the different feature extraction methods used in earlier EEG research will be discussed. It ends with the possibilities for classifying the data and a discussion about the different machine learning algorithms.

2.1 Background

A communication method between humans and computers is called a brain computer interface (BCI). There are two types of BCIs: invasive and non-invasive. With an invasive BCI, the sensors that measure brain activity are placed under the scalp by surgery. The advantage of this is a high signal-to-noise ratio, but on the other hand, such surgery carries risks and it is not possible to move the sensors afterwards to measure other brain areas [8]. Non-invasive BCIs make use of sensors that are placed on or very close to the scalp. This method does not require any form of surgery. In this study, the focus is on non-invasive BCIs, which will be explained in more detail in the following sections.

2.1.1 Electroencephalogram

Electroencephalography (EEG) is a non-invasive and portable way of measuring brain activity. The first research on EEG was done in 1929 by Berger [9]. Since then, research into EEG has continued to grow. In 1973, Vidal [10] tried to create a system that would allow people to communicate with computers through EEG. The term 'brain computer interface' was first coined. With his research, Vidal demonstrated the feasibility and potential added value of BCIs. BCIs now exists in many different forms and is used for various purposes, which will be discussed further in section 2.1.3.

To understand EEGs, we must delve a little deeper into the actual functioning of the brain. Britton et al. [11] explained this in their book about the basics of EEG. The brain activity measured by EEGs is produced by neurons inside a 2-5 mm thick layer of cells, also called the neocortex. In total, the human brain contains an estimated 86 billion neurons [12]. Two neurons communicate with each other in a so-called synapse. These neurons are then called the pre-synaptic and the post-synaptic neuron. The pre-synaptic neuron is located before the synapse and sends signals to the post-synaptic neuron. The signal transmission between the two neurons takes place through a chemical process, in which the cells communicate by passing on neurotransmitters [13]. The electrodes applied to the scalp each measure in a radius of approximately 6 cm^2, which contains about 100 cortical neurons [14]. Most of the measured data come from post-synaptic potentials of the pyramidal neurons. They are named after their triangular cell body, resembling a pyramid. Every

pyramidal neuron consists of about 7000 synaptic connections [15]. Signals from individual neurons are too small to be detected by the electrodes. So, a simultaneous signal transmission from around thousands of neurons simultaneously is needed to see activity in the data [14]. The waves in the EEG charts represent the intensity in voltages of the group of neurons measured by each sensor. Once the brain has to perform a task, the amplitude will be lower and waves of voltage will be smaller. This continues until the mental task is completed. The waves are measured in hertz, which means the number of cycles per second. The frequencies of brainwaves can be categorised into five different bandwidths, which are shown below. The list is adapted from Abhang et al. [16].

- Delta (0.5-4 Hz): Has a high amplitude. Occur during stage 3 of non-rem sleep, also called deep sleep.
- Theta (4-8 Hz): Registered in the hippocampus and are observed during physical activity as well as REM sleep. Occur when deeply relaxed or inwardly focused.
- Alpha (8-12 Hz): Arises from synchronous electrical activity in the thalamus. Occur at total relaxed or passive attention.
- Beta (12-35 Hz): Are measured especially when awake and conscious. Occur when anxious, active or relaxed, or at external attention.
- Gamma (>35 Hz): Occur during full concentration, such as studying or solving a difficult problem.

2.1.2 EEG electrode types

The main difference between different EEG setups is the way the electrodes are applied to the scalp. The different types of electrodes can be divided into wet and dry electrodes. The use of wet electrodes is considered a traditional method in EEG, due to it is relatively easy to implement a connection with a low amount of noise [3]. However, research on dry electrodes has grown in recent years due to several important advantages. Roberto et al. compared both methods and describes their advantages and disadvantages [4]. Wet electrodes do have a substance between the electrodes and the scalp, which is either liquid, gel or paste. The main advantage is that the recorded data from wet electrodes contain a small amount of noise, due to the low contact impedance between the electrodes and the scalp [17]. Each of the three different substances has its characteristics. Liquid penetrates well through the hair onto the scalp and is therefore good to use with longer hair. The disadvantage is that the liquid spreads easily and can create an unwanted connection between different electrodes. Gel can easily be applied to the desired spot with a syringe. Because of its more solid consistency, it does not spread as quickly to other electrodes and is therefore useful when using a large number of electrodes. On the other hand, the gel dries out quickly, which makes it less suitable for long-term sessions. The last type of substance is paste, which is more suitable for long sessions. The disadvantage is that the paste is less user-friendly. It takes a long time to apply and remove the substance from the scalp.

Dry electrodes are placed on the scalp without the use of any solution. In the last years, a lot of research has been done on these electrodes, because of their many advantages. To achieve the lowest possible resistance without the use of an electrode solution, the material of the electrodes is crucial. Where wet electrodes use solid metal electrodes, dry electrodes often consist of thin metal pins to reach the scalp through the hair [18].

Comparing the two different electrodes, we see that dry electrodes have a worse signal to noise ratio than wet electrodes. On the other hand, the preparation time for using wet electrodes is relatively long. The participant's scalp and hair must be cleaned extensively before the electrodes can be applied. Also, the scalp sometimes needs to be abraded, which can be experienced as an unpleasant process for the participant. This is a big difference with dry electrodes. A helmet with dry electrodes can be used almost immediately after donning and is often more comfortable to wear. In addition, the scalp does not have to be cleaned before and after using the helmet. Overall, it can be concluded that dry electrodes are much more user-friendly. This is an important aspect when it will be applied in for BCI gaming.

2.1.3 Current applications

Today, there are many different applications of BCIs. According to Nijholt [8], the applications can be divided into two different categories: control and monitor. The focus of control applications is on controlling devices by using your brain. With monitor applications, the aim is to find out the mental or emotional state of the user, which can then be used to control the user's environment or some other user interface. In this section, some promising examples from both categories will be described briefly.

BCIs can be used for diagnostic purposes, which therefore falls under the category of monitoring. Changes in brain activity can be used to recognise neurodegenerative disorders. For example, it is known that a decrease in intensity in the gamma frequency band can be directly linked to cognitive deterioration [19]. In addition, it is also known that Alzheimer's disease can be recognised by increasing activity in the delta and theta bands and decreasing activity in the alpha and beta bands [20]. These examples show that BCIs can play a major role in diagnosing many different neurological disorders at an early stage.

Another example of an application where EEG data is monitored, is analysing the learning process of students. High concentration of students can be recognised by an increase in frontal theta activity. A decrease in alpha activity suggests that information is being processed into long-term memory. With this information, mental fatigue of the students can be detected and taken into account. In addition, this makes it possible to offer a personal learning programme for each student, based on his or her strengths and limitations. Besides schools, monitoring EEG data can also be applied to the workplace. Stress and fatigue are common problems among employees, which can lead to lower performance and an increased risk of accidents at work. This can also be recognised by BCIs, so that action can be taken at an early stage. [4]

There are also many applications where BCIs are used as a way to control machines or computers. A well-known example is controlling a wheelchair [2]. This would be an excellent solution, especially for people with physical disabilities, who would no longer need to perform any muscular movements to control their wheelchair, but by just using their mind. A lot of research has also been done in the field of games and how BCIs can contribute to this. In 2007, the first BCI game called Mindflex was released. The game was developed by NeuroSky (California, USA) in cooperation with Mattel (California, USA) [4]. The goal of the game was to use your thoughts to guide a ball around obstacles. Since then, the development of BCI games has increased and many different versions of games have been made. However, technically and graphically, they are still far from being comparable to today's games [1]. This is therefore a field of research in which much development still needs to be done.

2.2 Feature extraction

Although only the data in the most relevant frequency range is used, a lot of noise can be found in these frequencies as well. To maximise the performance of the classifiers, it is important to work with the minimum data required to distinguish the different motor imagery actions. Due to the high sampling rate and the total of 16 electrodes, a large amount of data has to be handled with lots of noise included. The purpose of feature extraction is to extract discriminative information from the EEG data, which can be used as features for the classification process.

The most common methods of feature extraction for motor imagery tasks, can be divided into four groups: time-domain, frequency-domain, time-frequency analysis and spatial-domain [21]. The raw EEG data is in the time-domain, which represents the electrical activity of the neurons in voltage per unit of time. A representation of the data in the frequency domain, gives the amount of activity within a specific frequency range over a certain timeframe. Time-frequency analysis is a more complex method, which analyses the signal in both the time and frequency domain simultaneously. In the spatial domain, the locations of the sensors are also included in the analysis. This is in contrast to the previously mentioned techniques, where each sensor is analysed separately. In Table 2.1 an overview of the most commonly used methods for feature extraction is provided, including examples of studies in which they were applied.

Overview of feature extraction methods		
Time-domain	Autoregressive (AR)	[22, 23]
	Adaptive autoregressive (AAR)	[23, 24]
	Root-mean-square (RMS)	[25]
	Integrated EEG (IEEG)	[25]
Frequency-domain	Fast Fourier transform (FFT)	[26]
	Welch's method	[27]
Time-frequency domain	Short Time Fourier transform (STFT)	[28]
	Wavelet transform (WT)	[29]
	Discrete Wavelet transform (DWT)	[30]
Spatial-domain	Common spatial pattern (CSP)	[31, 32]
	Common spatial-spectrum pattern (CSSP)	[33]

Table 2.1 Overview of the most common feature extraction methods in BCI studies. [34, 35]

2.3 Classification

The purpose of this section is to provide an overview of the most commonly used and state-of-the-art machine learning classifiers in BCIs. Classifiers that are widely used in the field of BCIs are Support Vector Machines (SVMs), Linear Discriminant Analysis (LDA), Logistic Regression (LR), K-Nearest Neighbours (k-NN), Naive Bayes and Regression Trees [21]. SVMs and LDAs are most often found in previous research. The SVM classifier often performs best when compared to the other classifiers [32, 36, 37]. However, according to a comparison study by Ilyas et al. [38], Logistic Regression can get similar accuracy scores as SVMs. It also outperforms the other classifiers mentioned before.

Besides the traditional algorithms, Deep Neural Networks (DNNs) are increasingly making their way into the classification of EEG data [39–41]. Two types of DNNs that are often used are Convolutional Neural Networks (CNNs) [28, 42] and Recurrent Neural Networks (RNNs) [43]. Both seem to give better results than the SVM classifier according to previous studies [43, 44]. The advantage of these DNNs is that no feature extraction is required, so the raw EEG data can be used as input [44]. However, a DNN also has its disadvantages. There are many hyperparameters that can be tuned. When these are not set correctly, the performance will not be good either [45]. Therefore, it requires some time to apply a DNN in the right way. Another disadvantage is that the most important features of the data for the DNN, are often difficult to reason back to the original data afterwards [21]. For the purpose of classifying EEG data, it can be useful to know which information in the data is most relevant in order to adjust the recordings accordingly.

A selection of models from papers with high accuracy scores are listed by Garcia-Moreno et al. [46] (see Table 2.2). The purpose of this study is to evaluate methods that are most commonly used, as well as some state-of-the-art methods. Since SVM is the most widely used method and gives better results than LDA, this method was selected to use in this study. Logistic Regression has been used as baseline algorithm, due to its simplicity and good results as well. In addition, two state-of-the-art methods were evaluated. The first is a Long Short-Term Memory (LSTM) model. As shown in Table 2.2, this method tends to give the best results and is capable of achieving accuracy scores up to 98.9%. Even with multi-class problems, this algorithm scores higher than 90%. Another method that gave promising results – which is not mentioned by Garcia-Moreno et al. – is a Temporal Convolutional Network (TCN). Bai et al. showed in his comparison study [47] that TCNs can get similar results or even outperform LSTMs at classifying sequential data. Ingolfsson et al. applied this method to motor imagery EEG data and was able to achieve accuracy scores up to 97.4% on a 4-class problem [48]. The high accuracy scores from Table 2.2 were all achieved with large time windows. Ingolfsson et al. even used the whole task of 4.5 seconds as input sequence for the TCN model. For this reason, it is not expected that these results will be replicated in this study. The challenge is to achieve the highest possible accuracy score on a small window size.

Method	Dry electrodes	Subjects	Classes	Validation split	Accuracy
CNN + LSTM	Yes	Cross-subject: 4	Binary	90–10%	98.9%
CNN + LSTM	No	Cross-subject: 108	Five	75–25%	98.3%
LSTM	No	Intra-subject: 109	Five	5 × 5-fold	97.8%
SVM	Yes	Intra-subject: 8	Binary	4-fold	95.1%
CNN	No	Intra-subject: 20	Four	ICV	93.9%
LR	No	Intra-subject: 29	Three	50%–50%	90.5%
CNN	No	Intra-subject: 2	Binary	80–20%	86.41%
SVM	No	Intra-subject: 2	Binary	50–50%	82.14%
CNN + LSTM	Yes	Intra-subject: 1	Four	90–10%	80.13%
LSTM	No	Intra-subject: 9	Binary	5 × 5-fold	79.6%
CNN	No	Intra-subject: 9	Binary	60%–40%	78.44%
CNN + SAE	No	Intra-subject: 9	Binary	10 × 10-fold	77.6%
RLDA	No	Intra-subject: 9	Four	ICV	73.7%
CNN	No	Intra-subject: 9	Four	4-fold	~70%

Table 2.2. An overview of some state-of-the-art methods for classifying motor imagery EEG data. The list is sorted on the obtained accuracy score. CNN: Convolutional Neural Network, LSTM: Long Short-Term Memory, SVM: Support Vector Machine; SAE: Stacked autoencoder, RLDA: Regularized Linear Discriminant Analysis; ICV: Inner Cross Validation. The table is based on Garcia-Moreno et al. [46].

Method

3

This chapter will give a description of the used methodology in this study. Firstly, a description is given of how the EEG recordings were made. This includes a definition of the participants, the research environment and the equipment that was used, and the stimuli that were shown during the recordings. Lastly, the different methods of preprocessing, feature extraction and classification are explained.

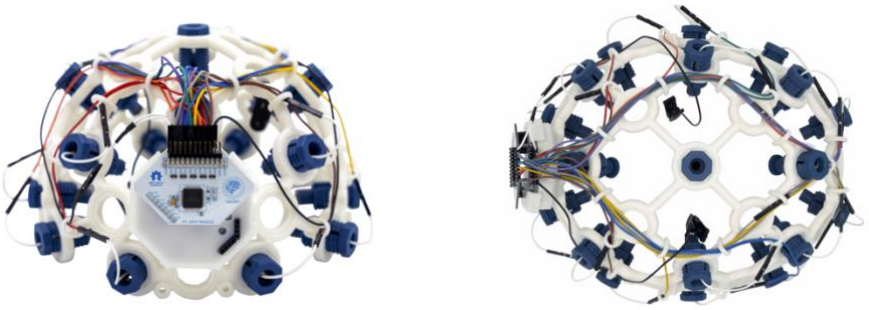
3.1 Data acquisition

3.1.1 Subjects

The goal of this project is to create an individual classification model based on one person, instead of a global model trained on multiple people. To ensure that the tasks during the recordings were performed properly, the researchers themselves served as participants. Therefore, extensive training in motor imagery tasks was not required. The participants included two male students, 25 and 26 years old. The participants performed a variety of mental tasks during the different recordings, which are described in subsection 'Stimuli' (3.1.3). Recordings of all different types of tasks were conducted by both participants.

3.1.2 Environmental setup

The EEG recordings were carried out in a controlled lab environment of Sogeti (Vianen, The Netherlands). EEG signals were recorded with the Ultracortex Mark IV from OpenBCI (Figure 3.1) in combination with the Brainflow library [49]. The Ultracortex Mark IV is able to record brain activity (EEG), muscle activity (EMG) and heart activity (ECG).



(a) Rear view *(b) Top view*

Figure 3.1. OpenBCI Ultracortex Mark IV, the 3D-printable EEG headset used for this study [50].

A total of 16 channels were sampled during the recordings. The locations of the sensors are based on the internationally accepted 10-20 system for EEG recordings (Figure 3.2). To make the headset capable of handling 16 sensors, a Cyton board in combination with a Daisy board was used [50]. The OpenBCI's WiFi shield was used to enable the headset to connect to the laptop via WiFi instead of Bluetooth. This ensured that a sampling frequency of 1000 Hz (samples per second per sensor) could be used.

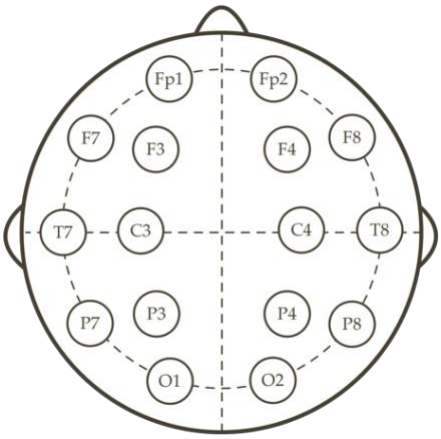


Figure 3.2. Electrode locations according to the internationally accepted 10-20 system for EEG recordings.

Before each recording could be started, the environment had to fulfil a number of requirements in order to keep the influencing factors as constant as possible. During the recordings, there was a maximum of two researchers in the room in order to limit the number of visual and auditory distractions. The location, desk, laptop and chair were exactly the same for all recordings. The laptop that has been used was running on the Windows Operating System with an Intel Core i5 processor and 16 GB of RAM. To minimise interference from unwanted muscle potentials, the participants were sitting as still as possible during the recording. Other interference is caused by the electrical grid of the building, which is around 50 Hz. It is assumed that this noise is constant as soon as the same location is maintained during the recordings.

3.1.3 Stimuli

Each recording session lasted approximately three minutes. To avoid fatigue and lack of concentration, a maximum of four sessions were conducted in sequence. Because it was unclear whether the EEG data varied over individual days, the total number of sessions has been spread over several days. At each recording, the participant performed one of the ten different mental tasks. All tasks contained left and right hand motor imagery, as these are the most commonly used for BCIs and have been proven to give good results [5, 6]. In addition, the same tasks were also performed physically – instead of only imagining it – to

analyse whether this gave better results. During the recording, a video was shown with a black background and a white bar at the bottom centre of the screen. So for example when the ball fell down on the left side, the participant should think about lifting his left arm for the 'Motor Imagery 1' task. The same applies to the right side. When the ball fell in the middle, the participant should do nothing. From now on, this is called the 'neither' task. An overview of the different mental tasks used in this study is given in Table 3.1.

Overview of the mental tasks	
Motor Imagery 1	MI task of lifting the left or right arm, when the bar should move to the corresponding direction. Nothing is done at the 'neither' task.
Motor Imagery 2	In addition to 'Motor Imagery 1', the MI task of lifting both legs is performed at the 'neither' task.
Motor Imagery 2b	Same as 'Motor Imagery 2', but the stimuli were shown at half speed.
Motor Imagery 3	In addition to 'Motor Imagery 1', the MI task of pressing the tongue to the roof of the mouth is performed at the 'neither' task.
Motor Imagery 3b	Same as 'Motor Imagery 3', but the stimuli were shown at half speed.
Physical 1	Same as 'Motor Imagery 1', but the tasks are actually executed here, instead of the movements only being imagined.
Physical 2	Same as 'Motor Imagery 2', but the tasks are actually executed here, instead of the movements only being imagined.
Physical 2b	Same as 'Physical 2', but the stimuli were shown at half speed.
Physical 3	Same as 'Motor Imagery 3', but the tasks are actually executed here, instead of the movements only being imagined.
Physical 3b	Same as 'Physical 3', but the stimuli were shown at half speed.

Table 3.1. Overview of the different mental tasks that were performed during the EEG recordings.

In the game 'Breakout', the player must rebound the falling ball by moving the bar. The setup of the recording was thus similar to the game. Research by Roc et al. showed that this could increase the performance of the BCI [51]. However, when the ball falls in a diagonal direction, it may be unclear to the participant in which direction the bar should be moved. This would then be at the expense of the purity of the training data. For this reason, the ball always falls vertically to the far left of the screen, to the far right, or exactly in the middle. In this way, the participant can immediately see which task he has to perform. Each task was shown 12 times per recording in random order. This means that

the participant performs a total of 36 tasks during one recording. A task consists of 1 second of black screen (rest state), followed by the falling ball for 4 seconds (Figure 3.3). Within these 4 seconds, after 1 second the bar automatically moves to the side of the ball. This contributes to making the stimuli look realistic, so that it resembles the final application. During the recording, it was important for the participants to be fully concentrated and avoid let themselves get distracted. Also, movements – such as head or eye movements – should be limited. This will reduce the noise in the data as much as possible.

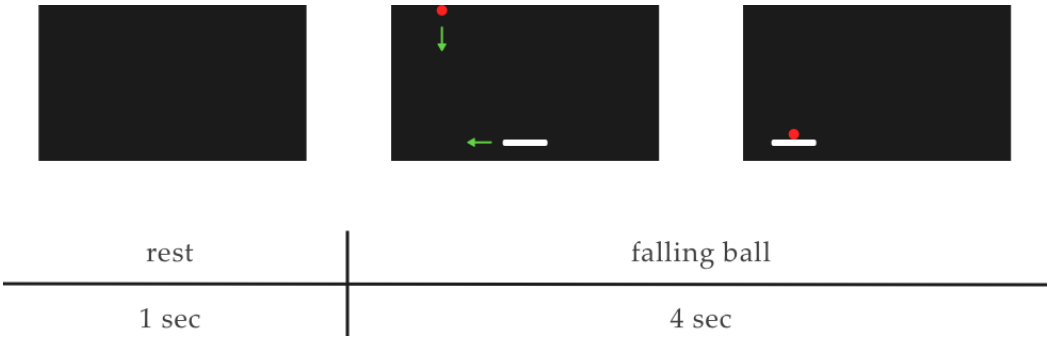


Figure 3.3. Stimulus presentation for the motor imagery tasks during the recording of EEG training data. Based on the position of the falling ball, the participant could see which mental task had to be performed. From the moment the ball is visible to the moment it has hit the bar, lasted exactly 4 seconds. In the recording sessions in which the stimuli were shown at half speed, each task lasted 8 seconds. Between each task the participant had a rest period of 1 second, in which a black screen was shown.

3.2 Preprocessing

3.2.1 Raw data

A visualisation of the raw EEG data can be seen in Figure 3.4. The EMG and heart rate (ECG) data that were recorded by the helmet as well, were not used in this study. The unfiltered EEG data has been converted to the frequency without applying any filters. The average amplitude of an entire task is displayed per frequency. The figure shows that most of the activity takes place in the frequency range 0-60 Hz. The peak around 50 Hz probably comes from the electricity grid at the recording location. There is probably also a lot of noise in the data at other frequencies. These little deviations in the data which are not generated by the brain are called artefacts [52]. Possible examples of artefacts could come from muscles in the head, heartbeats, eye movements or nearby electronic devices. In the *Wiley encyclopedia of biomedical engineering* by Bressler and Ding [53], all the possible artefacts in EEG data are described in more detail.

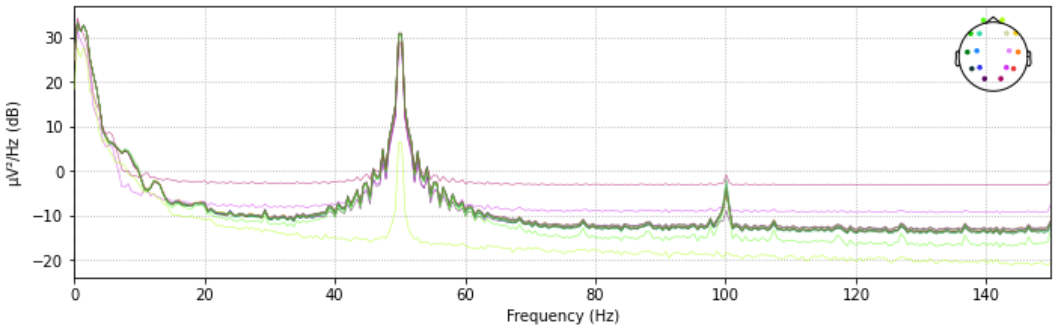


Figure 3.4. The raw EEG data shown on the frequency domain. The average amplitude of an entire task is displayed per frequency. Each colour corresponds to a specific sensor, which can be seen at the top-right corner.

3.2.2 Data preparation

At the beginning of each task, 800 ms has been removed from the data. This is because the data during the participant's reaction time when a new stimulus was shown, is unlikely to contain useful information. At the end of each task, 200 ms has been removed. This is because it is assumed that when the ball in the stimuli has almost hit the bar and the bar no longer needs to be moved, the participant stops performing the mental task.

The next step was to divide the tasks into a train, validation and test set. It was ensured that data from a specific task could never be present in multiple sets. This prevents the classifiers from predicting data from a task on which the model has already been trained. In addition, it was also ensured that the number of tasks of each class was balanced in a set in order to avoid bias.

As a final step, the tasks were split into overlapping time windows of 512 milliseconds each with an intermediate step of 50 data points. This means that each time window has, with respect to the previous time window, 50 new and 462 overlapping data points. A window size of 512 ms has been chosen because it is assumed to be short enough for gaming applications and gave good results in the study of Garcia-Moreno et al. [46].

By splitting the time windows after creating the train, validation and test set, it is ensured that the time windows of one specific task always come in the same set. As a result, the classifying models are always tested on data from tasks they have not seen before. The window size and window cut were initial values and were optimised at a later stage during the grid search, as described in section 4.

3.2.3 Channel selection

According to Kübler et al. [54], the most important brain area for classifying MI tasks is located at the sensorimotor cortex. This area is located in the centre of the scalp, as shown in Figure 3.5.

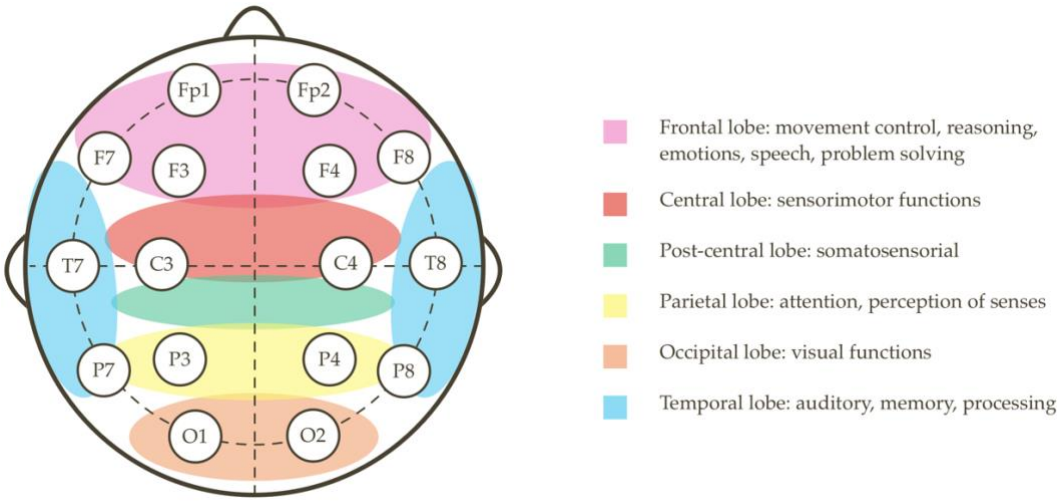
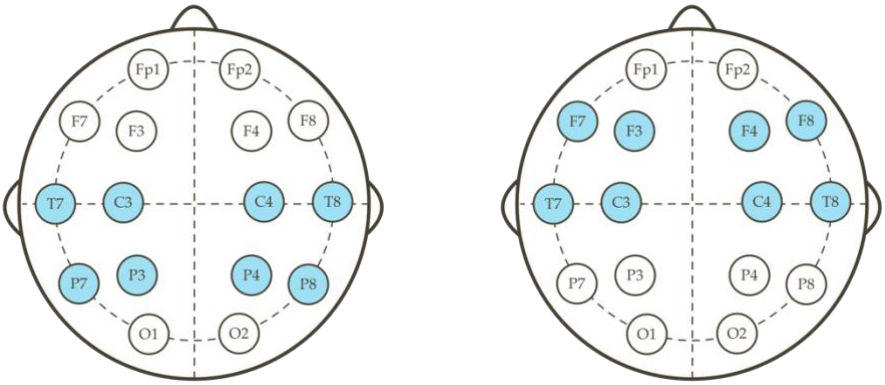


Figure 3.5. Sensor locations of the Ultracortex Mark IV helmet and the locations of the brain areas with their functions. The positions of the lobes are based on the paper of Garcia-Moreno et al. [46].

In this study, two different groups of sensors were compared. The first group was composed according to their distance from the parietal lobe while wearing the EEG headset. The sensors closest to this area were *C3*, *C4*, *P3*, *P4*, *P7*, *P8*, *T7* and *T8*. The second group of sensors is based on previous research. Bousseta et al. [55] and Garcia-Moreno et al. [46] used sensors that were closer to the frontal lobe and more widely distributed around the head. Based on the sensors used in these papers, a group was composed for this experiment with the sensors of the EEG headset that were the most closely located: *C3*, *C4*, *F3*, *F4*, *T7*, *T8*, *F7* and *F8*. The two different sets of sensors are shown in Figure 3.6.



(a) Parietal lobe *(b) Frontal lobe*

Figure 3.6. The locations of the two sensor sets that were compared in the first experiment, based on the international 10-20 system. Sensor set (a) is located above the parietal lobe of the cerebral cortex. The second sensor set (b) is located above the frontal lobe. The sensors that were used, are highlighted in blue.

3.2.4 Band-pass filter

A Finite Impulse Response (FIR) filter has been applied to the raw EEG data. This is a band-pass filter that cuts all values outside a certain range. A band-pass filter is one of the most commonly used methods for filtering out the frequency bands containing irrelevant information [34]. It consists of a low-pass filter and a high-pass filter. The former reduces all values after a certain frequency and the latter does the opposite, which can be seen in Figure 3.7. In this study, the band-pass filter has been employed with a range of 7-40 Hz. This contains the Mu, Beta and Gamma frequency bands, which are known to be associated with motor-related tasks [56].

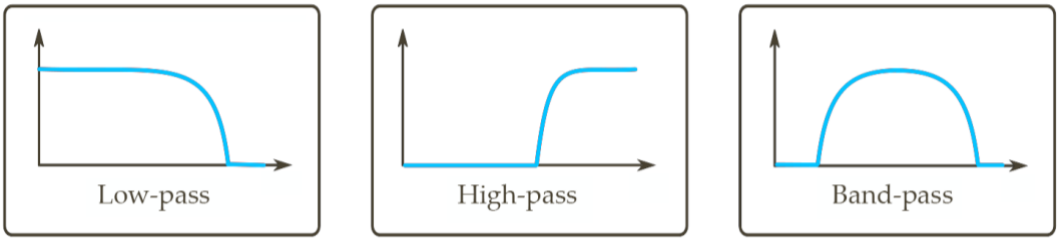


Figure 3.7. Three different kinds of filters with the frequency on the X-axis and the amount of EEG activity (voltage) that is let through on the Y-axis [57].

In Figure 3.8, the band-pass filtered EEG data is shown on the frequency domain. The dotted lines indicate the frequencies on which the low-pass and high-pass filters were applied. As can be seen in the figure, the amplitude of the data slowly decreases the further

it gets from the applied filters. The peak around 50 Hz is therefore no longer as prominent in the data.

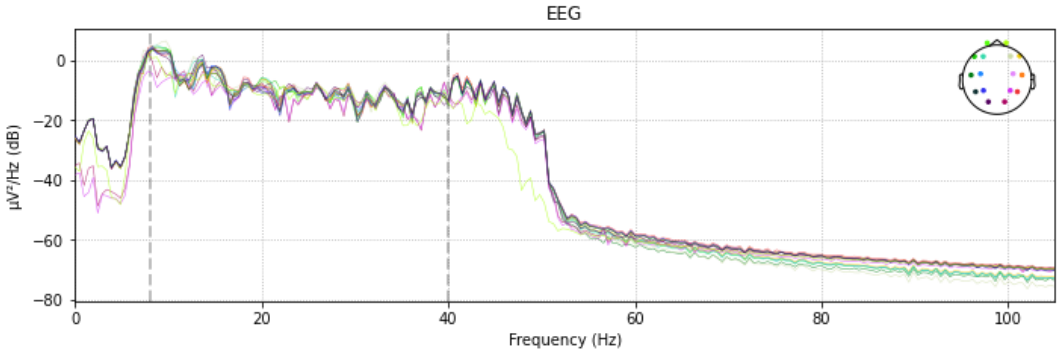


Figure 3.8. The raw EEG data shown on the frequency domain after applying a band-pass filter with a range of 7-40 Hz. The dotted lines indicate the frequencies on which the low-pass and high-pass filters were applied. The average amplitude of an entire task is displayed per frequency. Each colour corresponds to a specific sensor, which can be seen at the top-right corner.

3.3 Feature extraction

Feature extraction is a technique to reduce the dimensionality of the data. The result is a feature vector of a certain size, which then can be used for classification. Many different feature extraction methods can be used to classify EEG signals. The most common methods from previous research has been used in this study. The used methods are described in this section. It starts with a description of Common Spatial Patterns. Second, the Fast Fourier Transform, followed by its alternative called Discrete Wavelet Transforms. And lastly, the conversion of the results of these methods to statistical features.

3.3.1 Common Spatial Patterns

As discussed before, dry electrodes have a low signal-to-noise ratio. This can be a problem when the relevant signals are weak, while other signals within the same frequency band are stronger and thus dominant, such as muscle artefacts [58]. To tackle this problem, Common Spatial Patterns (CSP) has been used, which analyses multichannel data with multiple classes. For this study, the theoretical framework that has been used was proposed by Blankertz [59]. The goal of CSPs is to maximise the variance under one class and minimise it for the other class. A set of CSP filters will be estimated in the filter matrix $W \in \mathbb{R}^{C \times C}$, where C is the number of channels. A traditional CSP works with a binary problem, while the data in this study contains three different classes. A general way to extend CSPs to multiclass problems is to perform the algorithm on a set of binary subproblems. So with three classes, three different filter matrices W will be computed (for each binary pair).

The first step of the CSP algorithm is to compute a covariance matrix of each class with concatenated time-windows. For the sake of simplicity, the computations are explained for a case with two classes. The estimated covariance matrixes of the band-pass filtered data are denoted as $\sum^{(+)} \in \mathbb{R}^{CxC}$ and $\sum^{(-)} \in \mathbb{R}^{CxC}$. A detailed explanation of how these matrices are computed is given by Blankertz. The second step is simultaneous diagonalisation. In other words, it decomposes the covariance matrices into eigenvectors with corresponding eigenvalues:

$$W^{\intercal}\sum^{(+)}W = \Lambda^{(+)}, \quad W^{\intercal}\sum^{(-)}W = \Lambda^{(-)}, \quad (\Lambda^{(c)\text{diagonal}}) \tag{1}$$

where $\Lambda^{(c)}$ is the eigenvalue, which corresponds to one class with minimum variance and the other one with maximum variance. Determining W has the condition such that $\Lambda^{(+)} + \Lambda^{(-)} = I$, where I is the identity matrix. This means that the sum of the eigenvalues of two classes has to be 1. Equation 2 can be achieved by solving the generalized eigenvalue problem:

$$\sum^{(+)}w = \lambda\sum^{(-)}w \tag{2}$$

For applying CSP to multiple classes, Approximate Joint Diagonalisation (AJD) has been used. The technique is based on the Pham's algorithm, which he describes in his paper [60].

The last step is to sort the eigenvectors from high to low based on the corresponding eigenvalues. The higher the eigenvalue, the more variance between the two classes. This principle is explained by Koles [61]. The eigenvectors are also called CSP components. One can then determine the x number of best CSP components to use as features. The number of components used in this study was determined by a grid search, see section 4.4.

3.3.2 Fast Fourier transform

The raw EEG data gives a representation of the brain activity per unit of time. This is also called the time domain. Since the EEG signal is made up of many waves – each with its own frequency – it can be difficult for machine learning models to classify the raw data. Fast Fourier transform (FFT) is a method that converts each time window from the time domain to the frequency domain. This means that for each frequency value, a summary of the activity is given for the entire time window. This is illustrated in Figure 3.9. The implementation of FFT in this study is based on the book of Brigham [62].

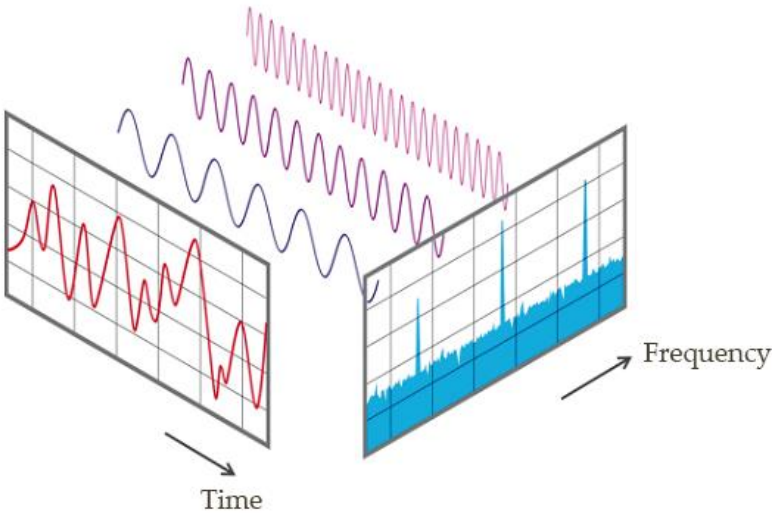


Figure 3.9. Illustration of the Fourier transform, which converts the EEG data from the time domain to the frequency domain [63].

3.3.3 Discrete Wavelet Transform

The main drawback of the previously mentioned Fourier transform is that all information from the time domain will be lost. For example, after the transform it is no longer possible to see how the intensity of the frequencies changed during the time window. This is a problem because there is probably not only relevant information in the frequency of a wave, but also the order in which the different waves occurred during a time window. For example, the previously mentioned classifiers such as LSTM and TCN are specifically designed for sequential data that comes from the time-domain. Both domains are therefore important for the classification process. In 1946 Gabor proposed a method to solve this problem, called Short Time Fourier Transform [64]. This method again splits the time windows into several segments, after which a Fourier transform is performed on each segment. In this way, information from both the frequency and time domain is retained. The problem of this method is that the time resolution – the extent to which information from the time domain is retained – is equal for both the high and low frequencies. Higher frequencies often change faster than lower frequencies and thus contain more information in the same time frame. Wavelet transforms offer a solution by increasing the time resolution for higher frequencies and increasing the frequency resolution for lower frequencies [65].

Figure 3.10 illustrates the time and frequency resolutions of the described methods. The size and direction of the blocks indicate the level of precision with which features can be extracted from the two different domains. The raw data (time series) has a maximum resolution in the time domain and a minimal resolution in the frequency domain. This means that features related to changes in time can be extracted very accurately, but features related to frequency cannot be obtained. For the Fourier transform, the exact opposite is

true. With the Short Time Fourier transform, the resolutions for the frequency and time domain are both medium-sized. With the Wavelet transform, the level of accuracy at which features can be extracted depends on the frequency band. The lower the frequency band, the more information the features contain that can be extracted from the time domain. The opposite counts for the frequency domain.

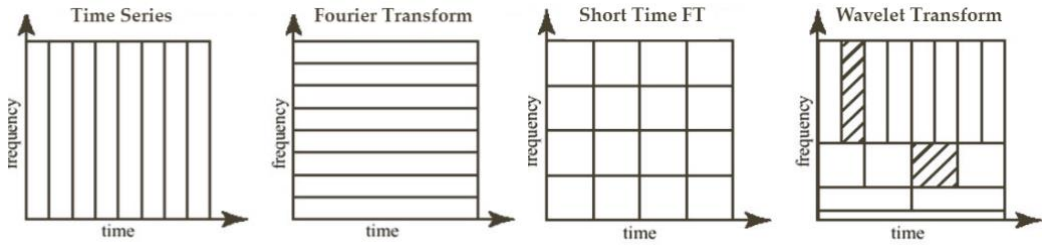


Figure 3.10. A schematic representation of the different transformations compared to the raw data (time series). The size and direction of the blocks indicate the size of resolution of the time and frequency domain. Figure is adapted from Baratchi [66].

A wavelet transform works by finding the correlation between a wavelet and the original data using the inner product [67]. The more the wavelet matches the data, the higher the output. Different frequencies can be analysed per time step by using scaling and shifting the wavelet. The real breakthrough of the Wavelet transforms came in 1988 when Ingrid Daubechies developed the Discrete Wavelet transform (DWT) [68]. The difference is that only a certain number of scales are used for the wavelet. The DWT formula is denoted by

$$W_n(s) = \sum_{n'=0}^{N-1} X_{n'}\, \Psi^* \left[\frac{(n'-n)\delta t}{s}\right]$$

where $W_n(s)$ is the transformed input data, $x_{n'}$ is the original input data, Ψ^* is the wavelet function and s is the scale. Daubechies developed a lot of different wavelets, of which the Daubechies 4 (db4) wavelet is the most commonly used [69]. Therefore, the db4 wavelet has been used in this study with a level 7 decomposition where only the frequency bands around 7-40 Hz were considered as features. The implementation of DWT is further explained by Wirsing [68].

3.3.4 Statistical features

One of the tested feature extraction methods was converting the features obtained from previously described techniques, to statistical features. This is a method mentioned in the review by Alimardani et al. [70]. The features that have been extracted are: the variance, standard deviation, mean, the 5th, 25th, 75th and 95th percentile, the entropy and root

mean square. In addition, the skewness and area under the curve were also used, as these were relevant features according to Zhang et al. [71].

3.4 Classification

Based on previous literature, four machine learning classifiers have been chosen for this study. As discussed in section 2.3, Logistic Regression has been used as baseline algorithm, Support Vector Machines as most commonly used method and Temporal Convolutional Networks and Long Short-Term Memory networks as state-of-the-art methods. These four algorithms are explained in the following subsections.

3.4.1 Logistic Regression

Despite the simplistic approach of Logistic Regression (LR), it is still widely used for classifying EEG data [21]. LR is very similar to Linear Regression. The difference between the two algorithms is that Linear Regression is able to predict continuous values, while Logistic Regression can be used for predicting discrete (binary) values. LR has a sigmoid shaped (s-shaped) logistic function, which translates each input value to a value between 0 and 1. See Figure 3.11. This value is interpreted as the probability that an input belongs to a certain class. The parameters of the logistic function are fitted based on the maximum-likelihood method, which is explained by Bishop [72]. The probability that a feature vector $\boldsymbol{\varphi}$ belongs to class C_1 can be written as

$$p(C_1|\varphi) = y(\varphi) = \sigma(\mathbf{w}^{\mathbf{T}}\varphi)$$

where $\mathbf{w}$ is the *weight vector* and with the constraint $(C_2|\varphi) = 1 - p(C_1|\varphi)$. It is basically a logistic sigmoid applied to a linear function. $\sigma(a)$ is the *logistic sigmoid* function, which is defined by

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

A feature vector φ is classified as class C_1 if $y(\varphi) \geq 0.5$ and to class C_2 if $y(\varphi) < 0.5$. Although the initial design of LR is built for classifying two categories, it can also be used for a multinominal problem. In order to do this, the model is fitted three times with a *one-vs-rest strategy*. For each class, it generates new data in which the two other classes are combined. The probability of the class is then calculated in relation to the other two classes. The class with the highest probability of the three new classification problems, will be used as prediction.

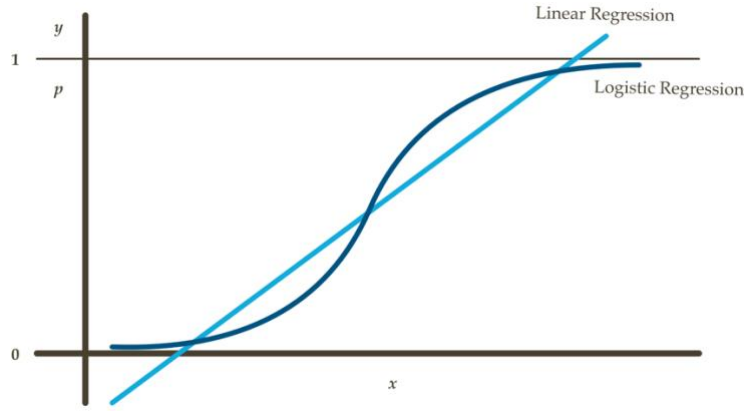


Figure 3.11. Illustration of Linear Regression (light blue) and Logistic Regression (dark blue). For the Logistic Regression, the y-axis indicates the probability for a certain class.

3.4.2 Support Vector Machine

The second classifier that has been used in this study is Support Vector Machine (SVM). This method classifies the data by fitting a linear hyperplane that separates the classes. This is done by maximizing the margin to both sides. See Figure 3.12. So in essence, SVM is a linear classifier. However, it often occurs that the data cannot be separated linearly. SVM addresses this by applying a hyperplane to the data in a higher dimensional space. This is done by transforming the input data with a kernel function, which is explained by Bishop [73]. There are several kernel functions that can be used within SVMs. The ones used in this research are called *Linear*, *Polynomial*, *Radial* and *Sigmoid*.

Like LR, SVM is designed for classifying two classes. However, SVMs can also be used for problems with multiple classes. In contrast to LR, the *one-vs-one strategy* is used to accomplish this. A new classification problem occurs for each pair of classes, in which one class is predicted. The most predicted class of all classification problems, will be used as final prediction.

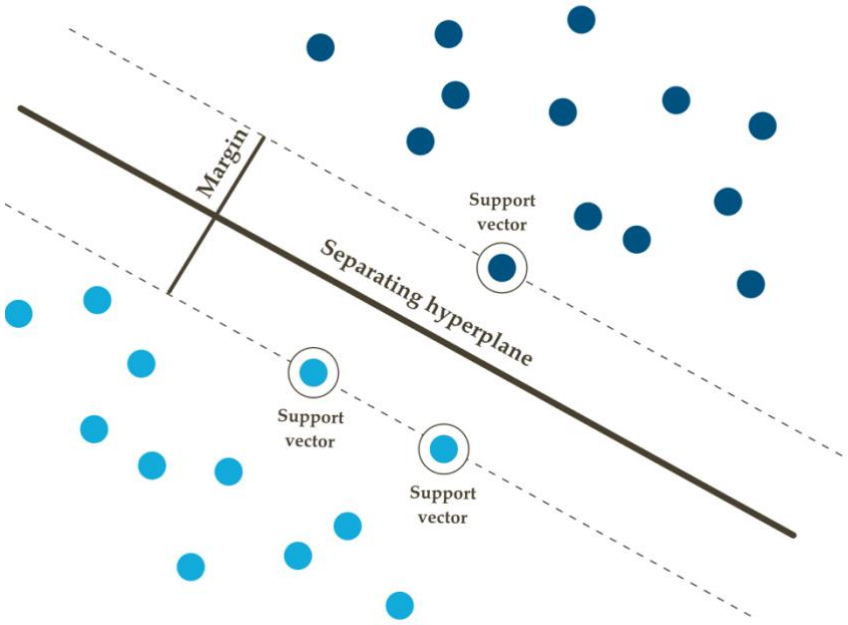


Figure 3.12. Illustration of a Support Vector Machine (SVM) model. It shows the separating hyperplane with a maximum margin such that it separates the two classes. The support vectors are the closest datapoints to the hyperplane that affects its position and orientation. The figure is based on the paper of Sharma et al. [74].

3.4.3 Temporal Convolutional Network

A Temporal Convolutional Network (TCN) is a neural network that gave promising results on EEG data in previous studies. The model used in this study has been adapted from the paper of Bai et al. [47]. The proposed implementation was designed by combining the best practices of various modern convolutional networks into a new architecture. According to Bai, the results are often better than the traditional convolutional networks.

A TCN uses *causal convolutions*, which means that an element in the output sequence is only affected by elements that occurred in the past (see Figure 3.13). In order to avoid extremely long convolutional filters, TCNs make use of *dilated convolutions*. This means that it increases the distance between the input elements that affects one single element of the output sequence. This is in contrast to a conventional convolutional layer, which can be seen as a 1-dilated layer since the input elements affecting one output element are adjacent. According to Bai, the dilated approach makes it possible to have deep networks with large filters, as well as a long effective history size. In other words, it can look far back into the past to make a prediction. Since the player is unlikely to switch mental tasks every fraction of a second, this can be very useful for classifying EEG data.

The set of elements of the input that affects one single element of the output, is called the *receptive field*. In the ideal situation, the receptive field has the same size as the input length. This means that a single element of the output depends on all previous elements of the

input. This is also called *full history coverage*. Instead of using a simple one-dimensional causal convolutional layer, Bai et al. proposed an approach with *residual blocks* which consists of two layers of dilated causal convolution. The output of these layers will be used as input for the next residual block. Adding more residual blocks increases the receptive field twice as much, compared to a basic causal layer, since each block consists of two such layers. The total size of the receptive field with two layers in each residual block, is determined by Equation 1, where b is the dilation base, k is the kernel size and n the number of residual blocks.

$$receptive\ field = 1 + 2 \cdot (k-1) \cdot \frac{b^n - 1}{b - 1} \tag{1}$$

The main advantage of a TCN is that it is capable to exponentially extend its receptive field size with only a linear increase in the number of parameters. In contrast to RNNs, it does not suffer from vanishing and exploding gradient issues while dealing with long input sequences [75]. At a vanishing gradient the weights of the lower layers of a neural network will change very slow or not at all. This causes a stagnating learning process of the model at an early stage. At an exploding gradient the models' weights grow exponentially, which results in an unstable model. This is important for the EEG data used in this study. The TCN model probably works best with sequential EEG data as input, such as the output of the CSP feature extraction method. Since in this study an EEG helmet is used with a relatively high sampling rate of 1000 samples per second per sensor, the input sequences of the sequential data will be quite long as well. Therefore, a vanishing or exploding gradient could become a problem.

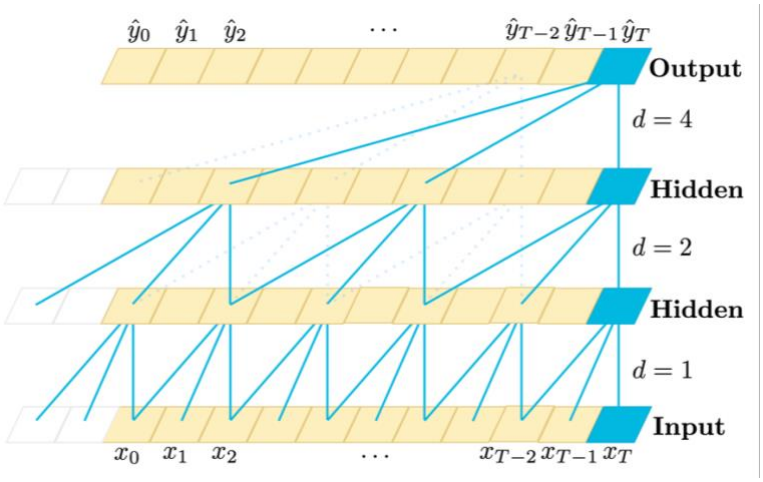


Figure 3.13. An example of a dilated causal convolution, with dilation factors d = 1, 2, 4 and filter size k = 3. The receptive field of this model has a full history coverage, since a single element of the output is affected by all previous elements of the input sequence. The figure is adapted from the paper of Bai et al. [47].

3.4.4 Long Short-Term Memory network

Another solution to the vanishing and exploding gradient issue could be a Long Short-Term Memory network (LSTM). The basic idea of LSTMs is that they hold relevant information through the whole processing of the input sequence and discard all information that is not required for the classification. Each element of the input sequence goes through a cell, as shown in Figure 3.14. The output of the cells evolves based on memory of the past. The first step in each cell is to determine what information will be let through. This decision is based on the input element (X_t) and the output of the previous cell (h_{t-1}). Then for each element in the cell state, a sigmoid layer outputs a number between 0 and 1, which determines the extent to which the values of the previous cell state (C_{t-1}) must be memorised. This layer is also called the *forget gate* (f_t). Together with the 'input gate' (i_t), it will be determined whether it forgets the previous cell state or updates the current cell state (C_t) with the new information. The exact computations within an LSTM cell are explained in detail by Zhang et al. [71].

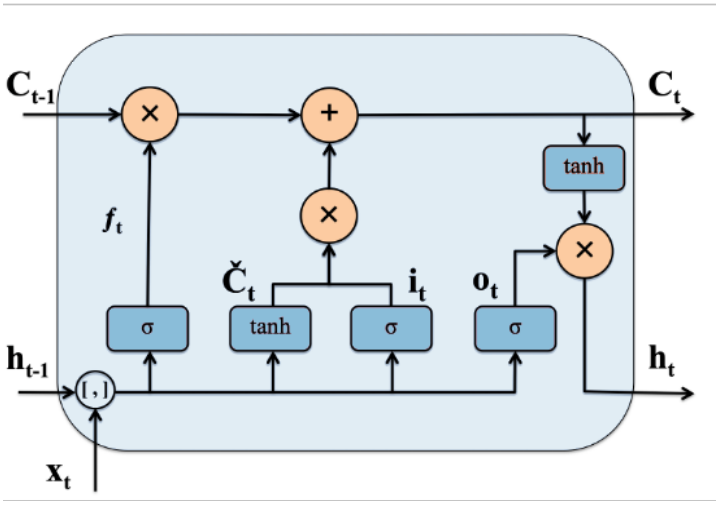


Figure 3.14. Illustration of a LSTM cell. The figure is adapted from the paper of Zhang et al. [71].

Wang et al. proposed a new classification framework based on the LSTM for classifying motor imagery EEG tasks [7]. Due to the promising results, this method was replicated and used in this study. This method uses a different approach for the preprocessing and feature extraction. The time windows of the EEG data are split into eight different sections. On each section, a linear regression analysis has been applied. The mean of the EEG data and the slope of the regression line will be used as Input A and Input B, as shown in Figure 3.15. A dense layer optimises the weighting coefficients for selecting the most relevant sensors, which will serve as input for the LSTM model. The output of the model will be merged and converted to a prediction by applying another dense layer.

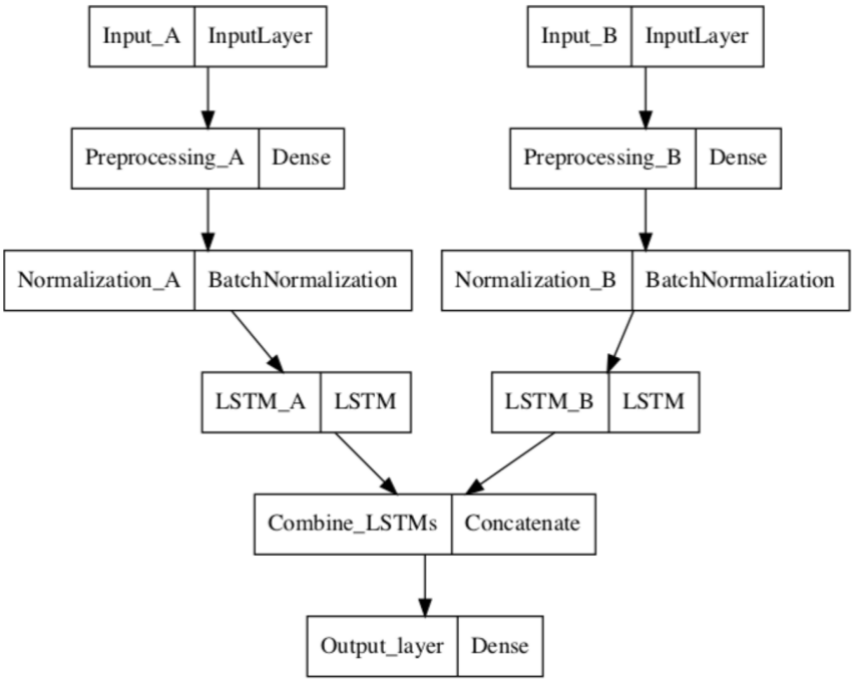


Figure 3.15. The LSTM architecture that has been used in this study. During the preprocessing, each time window is split into eight segments. A linear regression analysis is performed on each segment. The mean of each segment and the slope of each regression lines will be used as Input A and Input B.

3.4.5 Evaluation metrics

In this study, the different tasks are divided into three classes: left, right and neither. The number of tasks for each class is exactly the same. According to Thomas et al. [76], accuracy is commonly used as a measure of the classification process. Which is the ratio between correctly classified time windows and the total number of time windows inside the test set.

Besides the accuracy, it is also important within this study that the computation time of the classifier is as short as possible. While playing the game, the reaction time of the actions should be minimised for the best experience. Next to accuracy, this will also be used as an evaluation metric when comparing the different classifiers.

Results

4

In order to find the highest possible accuracy of the preprocessing and machine learning methods used, a grid search was performed. All techniques mentioned in chapter 3 have been tested step by step with different combinations and hyperparameters to obtain the optimal classification pipeline. The grid search is divided into several types of experiments. In each experiment, the emphasis is on optimising a small set of hyperparameters. The hyperparameters that provides the highest performance are used as constant values in the subsequent experiments. The way the grid search is structured is described in the following sections.

4.1 Sensor selection

The aim of the first experiment was to find out which sensors work best for the problem of this study. Two groups of sensors have been compared, which are described in section 3.2.3. For this experiment, a selection of three types of mental tasks was made. It has been taken into account that the mental task must contain at least two recordings, performed by the same participant on the same day. An overview of all recording sessions can be found in Table A3 of the appendix. The three recording sessions that have been used in this experiment are:

- Motor Imagery 2b (01-11-2022, subject 1)
- Motor Imagery 3 (12-08-2021-12-08, subject 2)
- Physical 1 (11-02-2021, subject 1)

At the first mentioned, the MI task was to lift both legs at the rest task, while the stimuli were played at half speed. At the second, the MI task was to press the tongue to the roof of the mouth at the rest task. The last task involved actually executing the tasks, rather than just imagining them. Here, the participant did not have to do anything during the rest task.

A window size of 512 samples was used. The preprocessing methods used were CSP with 8 components, followed by DWT. Finally, the outcome of the DWT was converted into statistical features, which were used as input for the classifiers. SVM and LR were used as classifiers for this experiment, as they are relatively fast to train compared to the complex neural networks. The list of parameters that are used for the SVM and LR models is shown in Table A1 in the appendix.

Figure 4.1 shows, per mental task, the accuracies of the SVM and LR classifiers obtained on the validation set. The results show that the frontal lobe sensor set scores better most of the time. This set contains the following sensors: C3, C4, F3, F4, T7, T8, F7 and F8. These will be used in subsequent experiments. See Table A2 in the appendix for the full results.

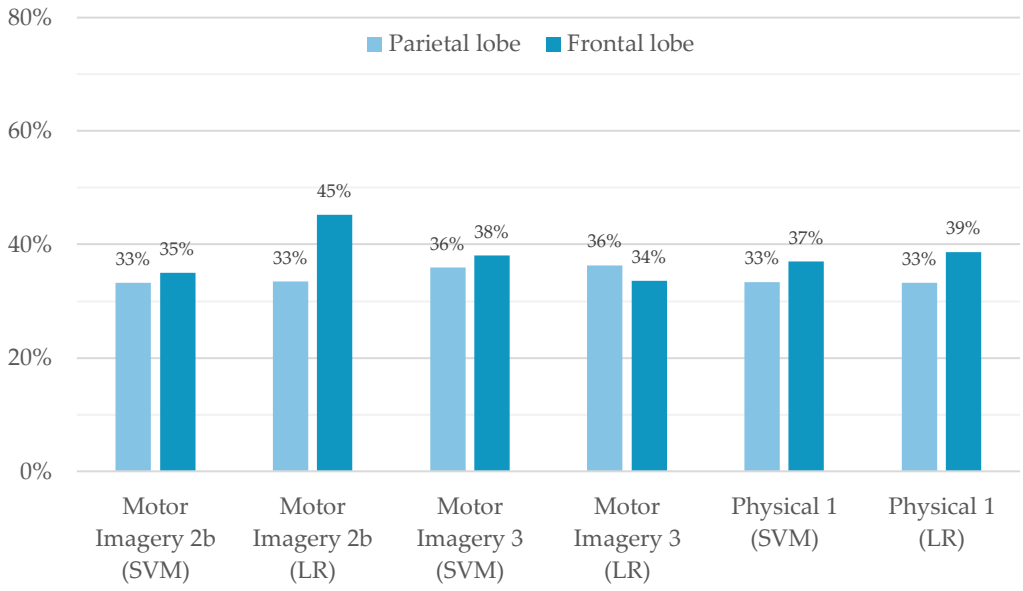


Figure 4.1. Comparison of the performance of the two sensor sets. One sensor set is located above the parietal lobe and the other one above the frontal lobe of the brain. Three different kinds of mental tasks are used for the comparison. The figure shows the accuracy scores on the validation set obtained with Support Vector Machines (SVM) and Logistic Regression (LR).

4.2 Mental task

The second experiment was intended to find out which recordings could be classified most accurately. The most important variables in a recording are the participant that did the recording, the moment it was made and the mental task that was performed. Zhang et al. research [71] demonstrated that the predictability of different recordings can vary significantly. In his study, Zhang decided to remove certain poorly predictable tasks from the data. It is assumed that the use of the best predictable data in the subsequent experiments makes the differences in accuracy more noticeable. This makes it easier to identify the best performing preprocessing and machine learning techniques. Also in this experiment, only LR and SVM were used as classifiers with the same constant values as in the previous experiment. The sensor set with the highest accuracy from the previous experiment was used to achieve the goal of this experiment: finding the best predictable mental tasks.

In total, 18 recording sessions were done with a total of 54 individual recordings. For each mental task, the performance of both SVM and LR was considered. The results of the other recordings can be found in Table A3 of the appendix. One of the two best predictable tasks was 'Motor Imagery 2b', in which the participant had to perform the MI task of lifting both legs at the rest task and where the stimuli were shown at half speed. The other mental task was 'Physical 2', in which the same tasks as in the previous one had to be actually executed, rather than just being imagined. In this task, the stimuli were played at normal speed. As

can be seen in the figure, the 'Motor Imagery 2b' task has a better average accuracy on the validation set. Hence, this mental task has been used in the subsequent experiments.

4.3 Pipeline combination

The result from the previous experiment – the best predictable mental task – has been used to identify the optimal combination of preprocessing and machine learning methods. As shown in Figure 4.2, seven different combinations of methods were tested. In contrast to the previous experiment, this time the TCN has been used as well, to see if there is a difference between the more 'simple' algorithms and a deep neural network. In the CSP, eight components are used as features in the classification process and no regularisation was used for the covariance estimation. Since the weights of the TCN model are initialised randomly, multiple runs were performed to minimize coincidence. The model with the highest accuracy on the validation set was used in the comparison of this experiment. See Table A1 in the appendix for the full list of parameters used in the TCN model.

The results (see Figure 4.2) showed that the combination of CSP, DWT and statistical features gave the best accuracy for all three classifiers. This combination of preprocessing and machine learning methods was therefore used in the subsequent experiments. See Table A4 in the appendix for the full results.

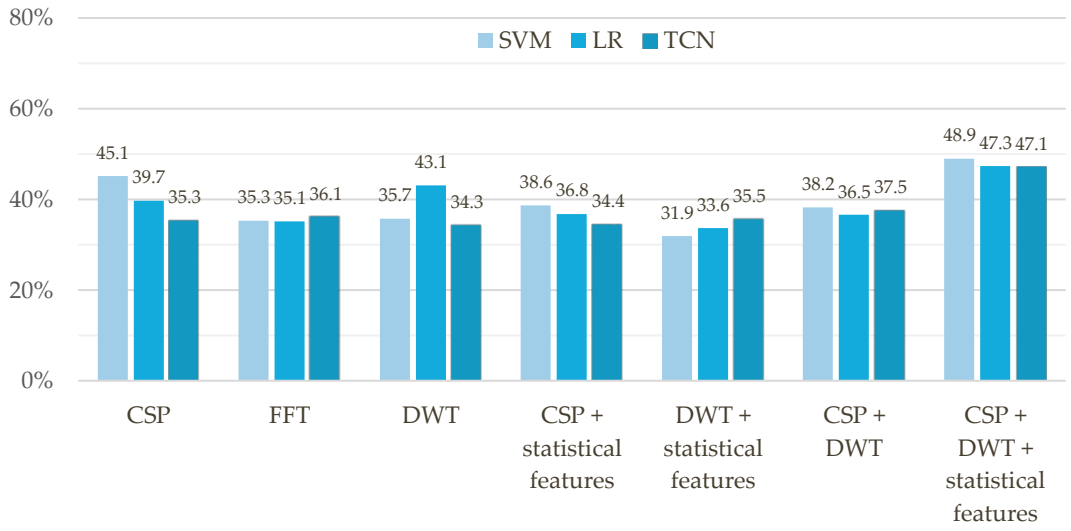


Figure 4.2. Shows the accuracy scores on the validation set of different preprocessing and machine learning combinations obtained with Support Vector Machines (SVM), Logistic Regression (LR) and Temporal Convolutional Network (TCN).

4.4 CSP optimisation

Because it was not clear from previous research which CSP parameters would work best for the problem of this study, it was decided to optimise the performance of this method using a grid search. Again, for the sake of simplicity, this was done only with SVM and LR. The first step was to find out whether the use of regularisation in the estimation of the covariance leads to better performance. Four different regularisation methods were tested and compared to the approach without regularisation. As can be seen in Figure 4.3, the Diagonal Fixed and PCA methods perform significantly lower compared to the rest. The remaining methods do not have significant differences for both classifiers. The Empirical method has a slightly better performance, compared to the other methods. The Empirical method was therefore used in the subsequent experiments. See Table A5 in the appendix for the full results.

Within the used MNE package for the CSP, the number of components can be determined for the output. The default value is equal to the number of sensors, in this case 8. In a subsequent experiment, it was tested whether the number of components used affected the performance of the classifiers. A range of 1 to 8 has been considered. As can be seen in Figure 4.4, reducing the number of components has a limited effect on the accuracy of the validation set. Only with 1 or 2 components, the accuracy is significantly lower. For the sake of dimensionality reduction, it was decided to keep the number of components as low as possible. A number of 4 components has been chosen since it is the lowest number which performs well on both SVM and LR.

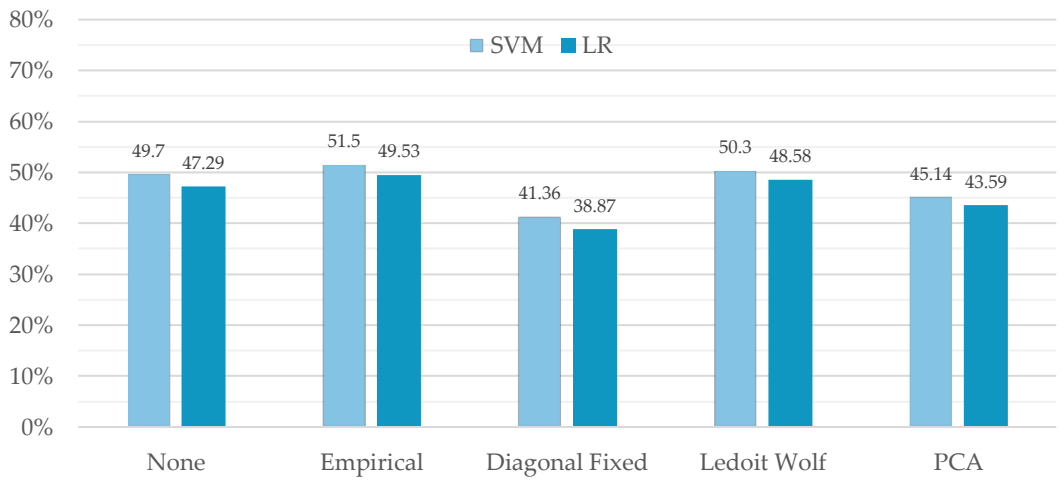


Figure 4.3. Comparison of the different regularisation methods for the Common Spatial Patterns (CSP) model. The figure shows the accuracy scores that are obtained on the validation set. PCA: Principal Component Analysis.

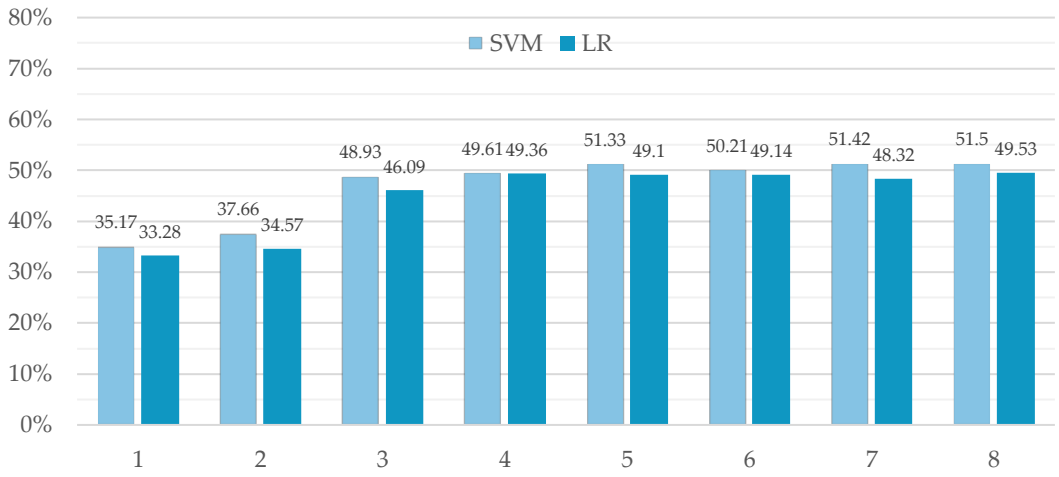


Figure 4.4. Comparison of the different number of components for the Common Spatial Patterns (CSP) model. The figure shows the accuracy scores that are obtained on the validation set.

4.5 Hyperparameter optimisation

The goal of this experiment was to tune the hyperparameters of the different classifiers. The Scikit-learn Python library is used for both the SVM and LR models. For the SVM, the different kernels and gamma types were considered first. All possible combinations were tested with [*1.0, 0.1, 0.01, 0.001*] as values for the learning rate (α) and [*100, 500, 1000*] for the number of iterations. The value for α and the number of iterations with the best accuracy are shown in Table 4.1. The model with 'poly' as kernel and 'scale' as gamma type gave the highest accuracy on the validation set.

Kernel	Gamma	α	Iterations	Train	Val	Test
linear	scale	0.01	1000	53.20	49.61	55.70
	auto	0.01	1000	53.20	49.61	55.70
poly	scale	1.0	1000	52.01	49.70	57.25
	auto	0.1	1000	55.12	49.61	55.70
rbf	scale	1.0	1000	58.89	49.01	56.09
	auto	1.0	1000	70.50	47.55	55.96
sigmoid	scale	0.1	1000	48.72	45.57	56.35
	auto	1.0	1000	39.16	40.93	34.84

Table 4.1. Comparison of the different kernels and gamma types used in the Support Vector Machines (SVM) model. All possible combinations were tested with [1.0, 0.1, 0.01, 0.001] as values for the learning rate (α) and [100, 500, 1000] for the number of iterations. Only the combinations with the highest accuracy score on the validation set are shown. The results of the best kernel and gamma type, based on the validation set, are marked in blue.

For the LR, the different solvers and penalties were considered first. Like before, all possible combinations were tested with [*1.0, 0.1, 0.01, 0.001*] as values for the learning rate (α) and [*100, 500, 1000*] for the number of iterations. The value for α and the number of iterations with the best accuracy are shown in Table 4.2. The model with 'saga' as solver and 'Elasticnet' as penalty gave the highest accuracy on the validation set. Therefore, these parameters have been used in the subsequent experiments of the optimisation process.

Solver	**Penalty**	**α**	**Iterations**	**Train**	**Val**	**Test**
lbfgs	None	0.01	500	56.89	46.35	55.18
	L2	1.0	1000	53.75	48.84	55.57
newton-cg	None	0.001	500	57.37	46.09	53.89
	L2	1.0	500	53.75	48.84	55.57
sag	None	0.001	500	55.45	47.12	54.79
	L2	0.1	1000	53.71	49.10	55.96
saga	None	0.1	500	54.72	48.15	54.66
	L2	1.0	500	53.75	49.18	55.96
	L1	1.0	1000	53.85	49.36	56.09
	Elasticnet	1.0	1000	51.79	49.61	57.25

Table 4.2. Comparison of the different solvers and penalties used in the Logistic Regression (LR) model. All possible combinations were tested with [1.0, 0.1, 0.01, 0.001] as values for the learning rate (α) and [100, 500, 1000] for the number of iterations. Only the combinations with the highest accuracy score on the validation set are shown. The results of the best solver and penalty, based on the validation set, are marked in blue.

The next step was to optimise the value for α and the number of iterations for both SVM and LR. All combinations tested with SVM are shown in Table 4.3, which shows the accuracy on the validation set. As can be seen, a value for the maximum iterations equal to or less than *500*, does not significantly exceed the probability of random guessing in any of the cases. A value of *1000, 1250 and 1500* iterations in combination with a learning rate of *0.1*, gave the best results on the validation set. An amount of *1000* iterations with a learning rate of *0.1* is chosen as the optimal combination for the SVM model because it costs less time to train compared to the other models. See Table A7 in the appendix for the full results, including the accuracy scores on the train and test set.

	Learning rates (α)				
Iterations	**2.0**	**1.0**	**0.1**	**0.01**	**0.001**
250	27.77	30.27	33.28	33.36	33.36
500	30.44	32.16	33.28	33.36	33.28
750	41.53	46.43	33.28	33.28	33.28
1000	47.72	48.75	49.70	42.56	33.45
1250	48.24	48.75	49.70	42.56	33.45
1500	48.32	48.84	49.70	42.56	33.45

Table 4.3. Comparison of the different combinations of learning rates (α) and the number of iterations used in the Support Vector Machines (SVM) model. The table shows the accuracy scores that are obtained on the validation set. The results of the best learning rates (α) and the number of iterations are marked in blue.

All combinations tested with LR are shown in Table 4.4, which again shows the accuracy on the validation set. As can be seen, a value for α equal to *0.001* does not significantly exceed the probability of random guessing in any of the cases. Other changes in the learning rate or the number of iterations do not cause major differences in performance. However, it can be concluded that a model with *500* iterations in combination with a learning rate of *2.0* performs slightly better than the other models. Therefore, this model is chosen as the optimal combination for the LR model. See Table A8 in the appendix for the full results, including the accuracy scores on the train and test set.

	Learning rates (α)				
Iterations	**2.0**	**1.0**	**0.1**	**0.01**	**0.001**
250	51.53	51.49	51.28	50.52	33.87
500	51.97	51.89	51.15	50.52	33.87
750	51.76	51.78	51.40	50.52	33.87
1000	51.63	51.66	51.59	50.56	33.87
1250	51.55	51.61	51.55	50.67	33.87
1500	51.83	51.76	51.59	50.86	33.87

Table 4.4. Comparison of the different combinations of learning rates (α) and the number of iterations used in the Logistic Regression (LR) model. The table shows the accuracy scores that are obtained on the validation set. The results of the best learning rates (α) and the number of iterations are marked in blue.

The next step was to tune the hyperparameters of the TCN model. First, the optimal learning rate (α) was determined with [*1.0, 0.1, 0.01, 0.001*] as possible values. The highest average accuracy on the validation set after multiple runs was 43.19%, obtained with $\alpha = 0.1$. The other three values obtained a significantly lower accuracy of around 33%.

Secondly, the optimal parameters with regard to the receptive field were tested. These include the kernel size, the amount of dilation blocks (stacks) and the number of dilations in each block. The following values were considered: [*4, 5, 6*] as the number of dilations, [*2, 3, 4*] as kernel size and [*1, 2*] as the number of stacks. In total, 150 runs have been carried out with a random combination of the mentioned values. Table 4.5 shows the top 10 models with the highest accuracy on the validation set. It can be concluded that most of the models in the top 10 do not have 'full history coverage', which means that all 144 features are being used in the TCN model. The model with the highest accuracy has been chosen for the further experiments.

Rank	Dilations	Kernel Size	Stacks	Receptive Field	Train	Val
1	4	3	2	61	52.70	50.39
2	5	2	2	63	53.24	50.13
3	5	3	2	125	52.08	49.44
4	6	3	2	253	53.75	49.36
5	4	3	2	61	49.19	49.27
6	5	4	1	94	55.05	49.18
7	5	3	2	125	53.09	49.18
8	5	2	2	63	52.15	49.10
9	5	2	1	32	52.44	49.10
10	5	2	2	63	51.97	49.10

Table 4.5. The top 10 models of the results, in which different combinations of the number of dilations, the kernel size and number of stacks were tested with the Temporal Convolutional Network (TCN). The ranking is based on the highest accuracy on the validation set.

To determine the best normalisation method, four different methods were tested with parameters from the previous experiments (see Figure 4.5). Batch, weight and layer normalisation were considered. In the fourth method, no normalisation was used. Layer normalisation has been chosen as the best method, since it shows the best accuracy at a later stage and therefore has a higher potential compared to the other methods.

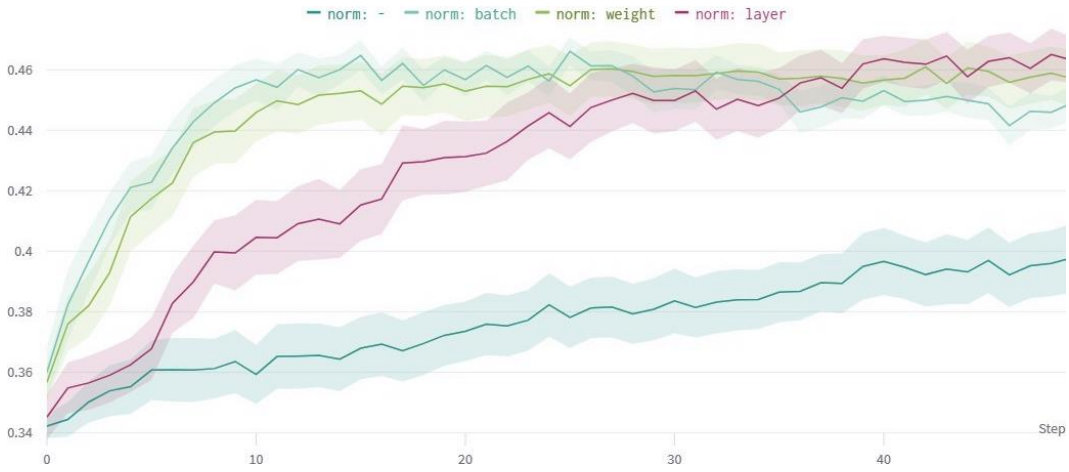


Figure 4.5. Shows the four different normalisation methods used in the Temporal Convolutional Network (TCN). The figure shows the accuracy scores that are obtained on the validation set over 50 epochs.

Finally, the best batch size, learning rate and optimiser were determined. The following values were considered: [*10, 50, 100, 200, 300*] as batch size, [*1.0, 0.1, 0.01, 0.001*] as learning rate and *adam* or *sgd* as optimiser. In total, 300 runs have been carried out with a random combination of the mentioned values. As can be seen in Table 4.6, it can be concluded that a batch size of 10, a learning rate of *0.01* and *sgd* as optimiser give the best results. Therefore, these parameters were considered as best performing TCN model.

Rank	Batch size	Learning Rate	Optimiser	Train	Val
1	10	0.01	sgd	51.32	50.82
2	10	0.01	sgd	50.34	50.04
3	300	0.1	sgd	51.14	49.87
4	10	0.01	sgd	51.07	49.79
5	100	0.1	sgd	45.02	49.61
6	50	0.01	sgd	51.32	49.27
7	50	0.1	sgd	49.37	49.10
8	10	0.001	sgd	51.39	49.01
9	50	0.01	Adam	50.63	49.01
10	100	0.1	sgd	51.25	48.93

Table 4.6. The top 10 models of the results, in which different combinations of batch size and optimisers were tested with the Temporal Convolutional Network (TCN). The ranking is based on the highest accuracy on the validation set.

4.6 Model comparison

After tuning all the models, the accuracy scores of the four machine learning classifiers have been compared. Because the weights of TCN and LSTM are being randomly initialised before each run, these models were repeatedly trained 20 times on the same dataset. The model with the highest accuracy on the validation set has been used in the comparison. The used hyperparameters of the LSTM model are given in Table A1 and are replicated from the paper of Wang et al. [7]. The results are shown in Figure 4.6. Only the validation set is used to optimise the models. The test set is therefore used to compare the different classifiers. The figure shows that the LSTM model obtained an accuracy score of 75.12% on the test set, which is significantly higher compared to the other algorithms.

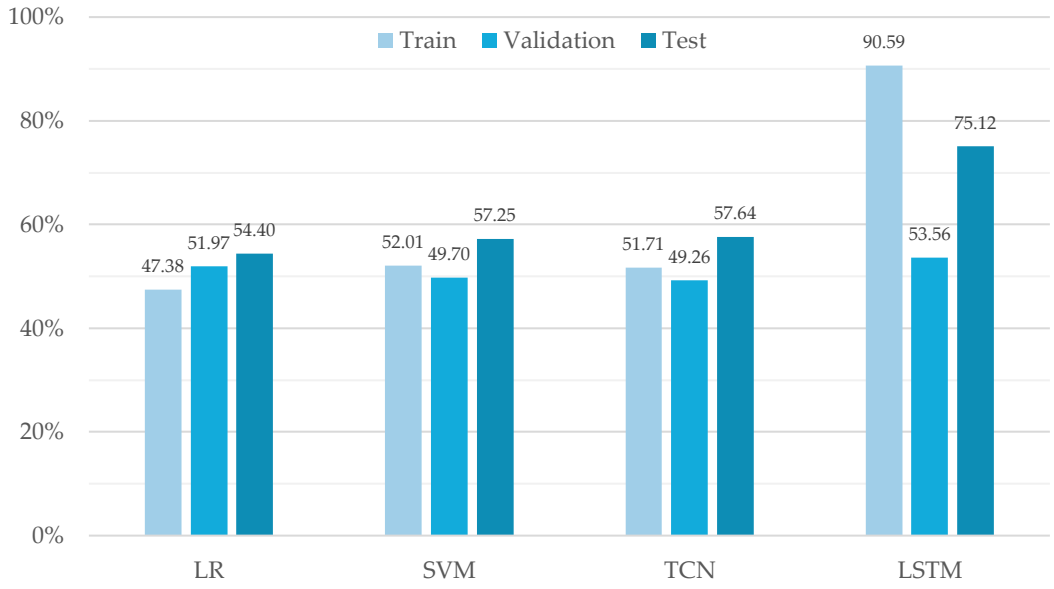


Figure 4.6. Comparison of the four used classifiers. The scores are obtained on the 'Motor Imagery 2b' tasks.

To evaluate the transferability of the tuned models, it was tested on other datasets as well. In total, the models were tested on four datasets: the two best datasets with 'Motor Imagery' tasks and the two best datasets with 'Physical' tasks. Figure 4.7 shows that the LSTM model performs significantly better than the other models for all datasets. Particularly in the 'Physical 2' task, LSTM scores with 62.3% considerably higher, while the other models are not able to obtain a score above the random chance of 33.3%.

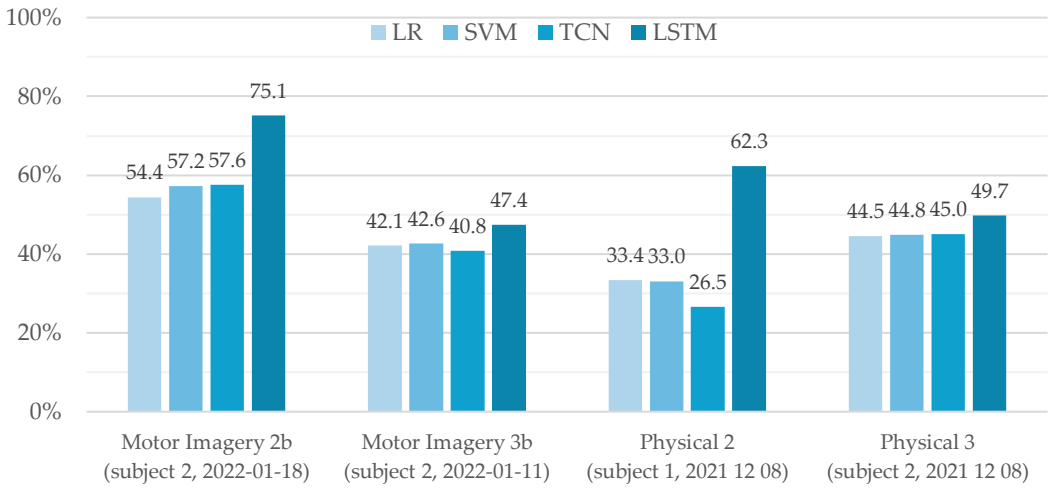


Figure 4.7. Comparison of the four used classifiers on different datasets. The classifiers are tuned on the 'Motor Imagery 2b' recordings from subject 2 on 18-01-2022.

Another important evaluation metric is computation time. This is especially important for playing the game, where the reaction time should be as short as possible. Therefore, the computation time of all the different preprocessing and classifying methods has been compared (see Table 4.7). Each method was performed 10 times on 100 time windows. After this, the average computation time per time window was calculated. As can be seen in Table 4.7, extracting the statistical features and the mean and slope for the LSTM model are the most time-consuming. The latter lasts considerably longer than the other methods with a computation time of 41 ms. In terms of the classifiers, the differences are minimal. Only Logistic Regression outperforms the other methods in the field of computation time.

	Method	**Computation time**
Preprocessing	Bandpass filter	0.008 ms
	CSP	0.047 ms
	FFT	0.421 ms
	DWT	0.284 ms
	Statistical features	6.761 ms
	Mean & slope (LSTM)	41.214 ms
Classifiers	LR	0.003 ms
	SVM	0.494 ms
	TCN	0.481 ms
	LSTM	0.446 ms

Table 4.7. Comparison of the computation time of each used preprocessing method and classifier. The table shows the average computation time per time window.

4.7 Error analysis

The overall accuracy score is a good indication of the performance of an algorithm. However, it does not tell everything about how well the game will be playable. To get a better understanding of the performance of the classifying models, the predictions are analysed in more detail in this section. As in the previous experiment, this analysis was carried out on the four different datasets. First, the confusion matrices of the LSTM model are evaluated (see Figure 4.8). Here, the *recall* and *precision* of the three different classes are considered in particular. The recall is the percentage of all windows of one class that were predicted correctly. The precision, on the other hand, only looks at the actually predicted classes and gives the percentage that was correct. The removal of one of the classes is also discussed, as in some cases this can significantly improve the performance. The disadvantage of this is that the game can then only be played with two actions.

Looking at the first confusion matrix (Figure 4.8a), it can be concluded that the overall accuracy is probably high enough for playing a game. However, this model does have a deviation to the right, which causes the bar in the 'Breakout' game to often move to the right while the player wants it to stay in the middle. It can be decided to take out the right class, since the other two classes have a much higher precision.

The second dataset, confusion matrix (Figure 4.8b), has the lowest accuracy score. However, it can be seen that the recall for the right class has a score of 100%, which means that all windows of this class are predicted correctly. The left class is predicted correctly zero times, but as soon as it is combined with the right class, a recall of 98.3% can be achieved. This means that almost all times that a certain direction is imagined, it can be distinguished well from the 'neither' class. Still, during the execution of the 'neither' class, a wrong prediction is made too often, which makes it difficult to use for gaming.

In the third confusion matrix (Figure 4.8c) it can be observed that the 'left' and 'right' classes are difficult to distinguish from each other. On the other hand, the 'neither' class can be distinguished very well from the other two classes. When the 'left' and 'right' classes are combined, the overall accuracy will probably be high enough to play the game.

Also with the last dataset, confusion matrix (Figure 4.8d), the game could probably be played properly. The model has a very high precision for both directions. Although the model often predicts the 'neither' class, this is not a problem for the game. In practice, this will mean that the bar in the 'Breakout' game will sometimes remain in the middle when the player wants to go in a certain direction. But the bar will almost never move in the opposite direction than the player wants. So in this case it is probably not even necessary to leave out one of the classes, in order to achieve a good gaming experience.

(a) Motor Imagery 2b (subject 2, 2022-01-18)

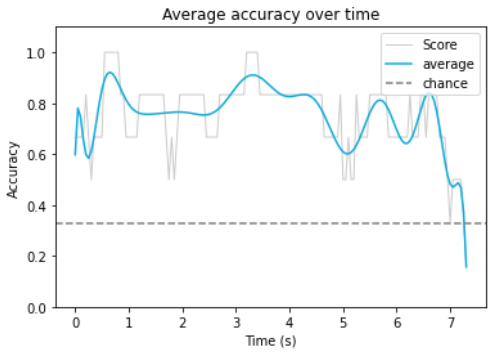
(b) Motor Imagery 3b (subject 2, 2022-01-11)

(c) Physical 2 (subject 1, 2021-12-08)

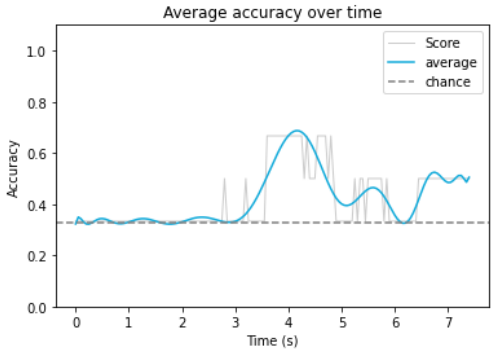
(d) Physical 3 (subject 2, 2021-12-08)

Figure 4.8. Comparison of the confusion matrices of the LSTM model on different datasets. The model is tuned on the 'Motor Imagery 2b' recordings from subject 2 on 18-01-2022.

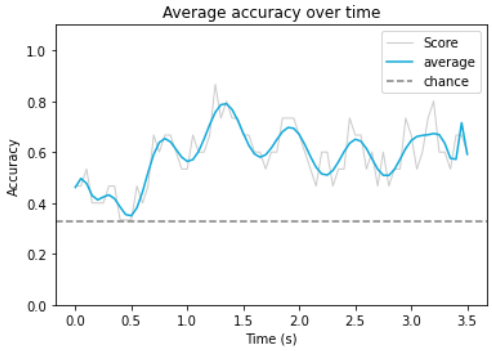
In Figure 4.9 the average accuracy is plotted during a task for each of the four datasets. This gives a good picture of the moments during a task when the brain signals can be classified well. For this experiment, the entire data of a dataset was used, so no data was subtracted at the beginning or the end. As can be seen in the different figures, the models usually have trouble classifying the data for about the first 0.5 seconds. This could be related to the reaction time as previously assumed. At the first data set (Figure 4.9a), there is also a significant decrease in accuracy at the end of the task. This could be caused by the fact that the participant stopped performing the task when the ball in the stimuli almost hit the bar. Therefore, it was beneficial to remove a part of the data at both the beginning and the end of the task.



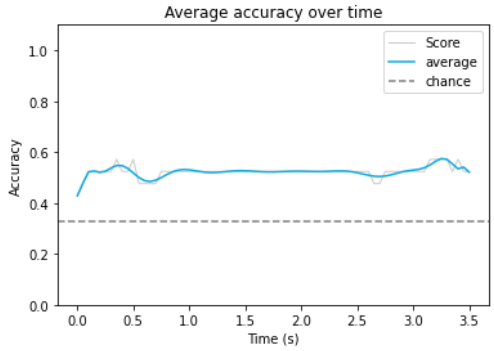
(a) Motor Imagery 2b (subject 2, 2022-01-18)



(b) Motor Imagery 3b (subject 2, 2022-01-11)



(c) Physical 2 (subject 1, 2021-12-08)



(d) Physical 3 (subject 2, 2021-12-08)

Figure 4.9. Comparison of the obtained accuracy of the LSTM model during a task on different datasets. Each plot shows the average obtained accuracy of all windows. The x-axis indicates the start time of each predicted time window, which has a total duration of 0.512 seconds. The model is tuned on the 'Motor Imagery 2b' recordings from subject 2 on 18-01-2022.

Discussion 5

5.1 Best methods

In this study, the most common and state-of-the-art methods for classifying motor imagery EEG signals are evaluated. Logistic Regression (LR) was used as baseline method, due to its simplicity and frequent use in recent research. Support Vector Machines (SVM) was selected as second method, because it is the most commonly used method. For the state-of-the-art methods, Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM) were selected, due to its promising results. The EEG data used in this study was recorded with dry electrodes and was split into relatively small time windows – compared to other studies – for the classification process. The research question of this study is how these methods perform on such data with small time windows.

The Logistic Regression as baseline method achieved a maximum accuracy score of 54.4%. The SVM could slightly improve this with a score of 57.3%. This was to be expected, as the performance of these two models did not differ much in other studies either. A remarkable result was the fact that the TCN model – which is considered to be a state-of-the-art method – obtained a maximum accuracy score of 57.6%, which is not significantly higher than the SVM model. One reason for this could be that the TCN model, proposed by Bai et al. [47], is especially designed for the classification of sequential data. In this thesis, the TCN is tested both the time and frequency domain. It was expected that this model would perform best in combination with Common Spatial Patterns (CSP) as a feature extraction method. The CSP method maximises the variance between the different classes, but keeps the data in the time domain. Nevertheless, TCN turned out to be the best performer in the frequency domain, using the Discrete Wavelet Transform (DWT) and statistical features. It even achieved a growth of almost 12% accuracy compared to the time domain. A possible reason for this can be found by looking at the results of the TCN model with regard to the receptive field. It shows that not all features contain the same amount of relevant information. Models with a lower receptive field often performed better compared to models with 'full history coverage'. This can be caused by the model training on features that contains too much noise, which can lead to overfitting.

The LSTM model was able to achieve an accuracy score of 75.1%, which was much higher than the other models. One reason for this could be that the LSTM uses a different feature extraction method compared to the other models. The approach was adapted from the paper of Wang et al. [71], in which the time windows were split into eight different sections. A linear regression was performed on each section and the means and slopes were used as features. As with DWT, these features give an overall view of change in activity during the time window. The difference is that with the DWT the segments are also split into different frequency bands. Since they are two comparable feature extraction methods, it can be

concluded that it is particularly the LSTM model that is more suitable for classifying the EEG data than the other models.

5.2 Data acquisition

Different combinations of mental tasks were tested in this study. It turned out that it is important to perform a mental task during the 'neither' task as well. Recordings in which nothing was done during this task appeared to be difficult to classify. If we compare the two mental tasks used for the 'neither' task, we see that lifting the legs works better than pushing your tongue against the roof of your mouth.

In addition, the duration of the tasks was evaluated as well. Each motor imagery task was played at two different speeds. At the high speed the tasks lasted exactly 4 seconds and at lower speed 8 seconds. The difference in the durations of the tasks gave varying results. It is therefore hard to say which of the two works better. It probably has both advantages and disadvantages. An advantage might be that less data is lost to the participant processing the stimuli and converting it into the correct action. A disadvantage might be that in longer tasks, the participant loses concentration more quickly and has difficulty performing the mental task for the entire period.

Furthermore, during the recordings it was noticed that the helmet has a lack of user-friendliness. The time it took to prepare the helmet for each use was too long. The fact that the helmet was heavy and that the tiny pins of the sensors were penetrated into the scalp, made a session of more than half an hour very unpleasant. So there is still a lot of room for improvement when it comes to the wearability and user-friendliness of the helmet. Therefore, this helmet is perfect for research purposes, but is not suitable for extensive gaming sessions. If this is the goal, it is recommended to do more research on the user-friendliness of different helmets.

5.3 Ready for gaming?

The second research question of this study is about whether the performance of the above-mentioned methods is sufficient for real-time BCI gaming. Real-time gaming refers to games in which the player's actions must be instantly translated into actions in the game. This is not the case for example in turn-based games, where the speed of the actions is less important. In real-time gaming, the response time of the game must therefore be as small as possible. A factor that influences this is the computation time of the used preprocessing and classifying methods. The results show that the preprocessing method for the LSTM model consumes the most time, with an average computation time of 41.2 ms for one time window. The next method that takes the longest to execute is the calculation of the statistical features, with an average of 6.8 ms per time window. All other preprocessing and classifying methods take less than 0.5 ms. These are such low values that the differences can be neglected. The durations of the two more time-consuming methods are

probably small enough for playing a simple real-time BCI game. When the developments of EEG classification are such that more complex games can be played, these times can become a drawback. In that case it is recommended to do further research to keep the computation times as low as possible.

The maximum accuracy score of 75.1% achieved by the LSTM model is assumed to be more than enough for playing a simple game. However, when the same model was applied to other datasets, the accuracy score dropped significantly. After analysing the predictions, it can be concluded that in most cases it is necessary to remove one of the three classes or combine it with another. It was often difficult to distinguish between the 'left' and 'right' classes. This might be due to the fact that these were the same type of mental tasks that had to be performed with either the left arm or the right arm. However, tasks performed with the legs could be more easily distinguished from those performed with the arms. Therefore, it is recommended to use mental tasks that are not too similar. One can also choose an approach in which two easily distinguishable tasks are used and the probability of the predictions determines the actions in the game. In the 'Breakout' game, this could work by making the bar stand still until a certain probability threshold for an action is reached. The action is then only activated when the algorithm is certain enough. This way you still have three different actions in the game by using two different mental tasks.

5.4 Limitations

The data was recorded by people without any experience in performing motor imagery tasks. Research by Roc et al. and Alimardani et al. shows that these kinds of tasks require a certain kind of training in order to be executed properly [51, 70]. According to these researchers, this can play a major factor in the accuracy of the classifiers. Yet with an eye on the end goal – playing a video game – the ideal scenario is that also untrained people can play the game straight away. In future research, a solution should be found that allows users to be instructed efficiently, while at the same time keeping the learning curve as low as possible.

Another limitation, which has to do with the Ultracortex Mark IV helmet used in this study, was that the sensors might not have made proper contact with the scalp. A sign that this was indeed the case was that some sensors in the data returned zero values over a longer period. Moreover, it was not always the same sensor where this happened. It varied over the different recordings, which makes it hard for the classification models to deal with.

In addition, an assumption was made that the sensor locations from the literature were at exactly the same location for the helmet used in this study. This may have caused problems since the two sensor selections tested were based on previous research. There is also a real possibility that the sensors were not exactly at the same location between the different subjects or the different recording sessions. This is because it could not be guaranteed that the helmet would fit on the head the same way for each session.

Another risk lies in the design of the grid search. At each step of the grid search, a certain set of variables was optimised. The variables with the best performance were then used in the following experiment to optimise a new set of variables. With this approach, there is a chance that a certain set of parameters, which has already been ruled out in earlier experiments, would have given the best performance in the end. In other words, the outcome is a local optimum where not all possible combinations have been tested. In addition, it is assumed that a window size of 512 ms is the maximum for a good gaming experience. In future research, other window sizes can be explored to find the ideal balance between accuracy and experience. It could also be examined whether other values for the window cut, currently 800 ms at the beginning and 200 ms at the end, give an improvement in performance.

A final limitation of this study is that no form of feature selection was applied. With such method it can be determined which features have the most relevant value for predicting the EEG data. By omitting the irrelevant information, overfitting can be prevented which can increase the accuracy score.

5.5 Future research

Whenever dry electrodes are used for real-time BCI gaming applications, the following things should be taken into account. The first thing is the location of the sensors. In recent research, many different theories are used. In this study, the sensors placed around the frontal lobe gave better results than those around the parietal lobe. It also turned out that the trained models were difficult to transfer to other data sets. This also confirms the theory by Zhang et al. [71] that the EEG data of different days or persons, can vary significantly. Therefore, it is important to train a model using only data recorded by one person and preferably carried out in one consecutive session. When the intention is to play a game, it is recommended to record the data and train the model right before each gaming session. If one wants to use data from different people or recording moments, Transfer Learning can offer a solution [77]. With this method, a trained model can be reused for subject-to-subject or sessions-to-session purposes.

The LSTM model in combination with the preprocessing method proposed by Zhang et al. [71], gave significantly higher results than the other models and is therefore recommended for use as a classifier. In the future, research can be done into the performance of TCN with similar data used in the LSTM model. Future research might also explore the approach of combining the features from the different preprocessing methods, in order to get a higher accuracy score.

There are also two other promising techniques – which are not used in this study – that could improve the results. The first technique is called Adaptive Learning [34]. This is a method in which the model gives live feedback to the user during the recording. This way, the user actually sees what the model predicts while thinking about left or right. This

enables the user to learn how the model works, instead of just the other way around. In addition, the model is constantly updated based on the new data received during the recordings. So it is an on-the-spot learning process for both the model and the user. The second technique that could be considered as well, is called Ensemble Learning [78]. This method combines different types of classifiers in order to achieve a better prediction. All models make a prediction simultaneously based on the input. All predictions are then combined into one prediction using an aggregating model. This is usually a linear model, such as logistic regression for binary classification. In this way, the model learns which classifiers perform best in certain situations. When these techniques produce a higher accuracy score, the effect on the Logistic Regression and Support Vector Machines can be examined as well. In this case, it is recommended to use CSP, DWT and statistical features as preprocessing methods, as these gave the best results in this study.

Conclusion 6

The aim of this study was to investigate whether the most common and state-of-the-art methods for classifying motor imagery EEG signals can be used for real-time BCI gaming. For this purpose, only EEG data recorded with dry electrodes was used. The EEG data was split up into small time windows of 512 ms to minimise the reaction time of the algorithm. Four different classifying algorithms were evaluated: Logistic Regression (LR), Support Vector Machines (SVM), Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM). LR was used as baseline method and achieved a maximum accuracy score of 47.3% in the first place. After further enhancement of the model by performing a subject-specific hyperparameter search, it did improve with 7.1% and achieved a maximum accuracy of 54.4%. SVM obtained without enhancement of the model an accuracy score of 48.9%. After the hyperparameter search, it did increase with 8,4% and achieved a maximum accuracy of 57.3% The TCN model is one of the state-of-the-art methods and obtained an accuracy score of 47.1% at first. After enhancing the model, it was able to improve by 10.5% and obtained a maximum score of 57.6%. All three models performed best with Common Spatial Patterns (CSP), Discrete Wavelet Transform (DWT) and statistical features as feature extraction methods.

Regarding the recordings, this study showed that the type of mental tasks can affect the results as well. Besides the motor imagery task of lifting the left or right arm, lifting both legs worked better as third task than pushing your tongue against the roof of the mouth. In addition, with some datasets the model had difficulties distinguishing the 'left' and 'right' classes. In such situations, it could be beneficial to remove one of the three classes or combine it with another one. One can choose an approach in which two easily distinguishable tasks are used, but only activates an action in the game when a certain probability threshold is reached. Regarding the sensor locations, there are various theories from recent research about the optimal locations. In this study, it appeared that the sensors placed around the frontal lobe gave better results than those around the parietal lobe.

The LSTM model is the other state-of-the-art method used in this study. The approach of this method is adapted from Wang et al. [71], in which each time window is broken down into eight segments and converted to features by applying a linear regression on each segment. This method replaced the previously mentioned feature extraction methods. The LSTM model outperformed the other models with a maximum accuracy score of 75.1%. However, the accuracy score dropped significantly after applying the same model on other datasets. This means that the data coming from different people or recording sessions can vary a lot. Therefore, it is recommended to record the data and train the model right before each gaming session.

The conclusion of this study is therefore that the performance achieved by the most common and state-of-the-art methods is indeed much lower when used on data from dry electrodes with small window sizes. However, as in previous studies, the LSTM model appears to have the highest accuracy compared to the other models. The next question is whether this performance is good enough for playing a real-time BCI game. With proper mental tasks, it seems that a simple BCI game can be played. The LSTM can also be further optimised for even better performance. A note here is that the player has to train a new model for each gaming session. Playing with a model that has been trained at a different moment or by another person is not possible without applying techniques intended for this purpose, such as transfer learning. One thing that is clear for sure, is that all the exceptionally high results from recent studies give a distorted view of the development and possibilities of using BCIs for gaming. The progress is on the right track, but not yet as far as it seems.

Appendix

Classifier	Parameter	Value
LR	Penalty	l2
	Solver	lbfgs, saga
	Learning Rate	1, 0.1, 0.01, 0.001
	Maximum Iterations	100, 500, 1000
SVM	Kernel	rbf
	Learning Rate	1.0, 0.1, 0.01, 0.001
	Maximum Iterations	100, 500, 1000
TCN	Epochs	50
	Learning Rate	1.0, 0.1, 0.01, 0.001
	Batch Size	100
	Optimiser	Adam
	Dilations	6
	Kernel Size	4
	Stacks	1
	Normalisation	Layer
	Dropout Rate	0.001
	Filters	3
LSTM	Epochs	100
	Learning Rate	0.001
	Batch Size	64
	Optimiser	Adam
	Normalisation	Batch
	LSTM-cell Dropout Rate	0.4
	Loss Function	Cross entropy

Table A1. The used initial values of the hyperparameters for each classifier before tuning. In case of multiple values, all values were tested individually. The model with the highest accuracy on the validation set has been used for comparison during the experiments.

Classifier	Mental task	Parietal lobe			Frontal lobe		
		Train	Val	Test	Train	Val	Test
SVM	Motor Imagery 2b	35.30	33.26	33.21	38.24	35.02	34.40
	Motor Imagery 3	59.12	35.95	33.81	59.93	33.67	40.05
	Physical 1	45.91	33.36	32.17	56.63	37.04	41.18
LR	Motor Imagery 2b	45.98	33.42	31.61	55.40	45.23	37.85
	Motor Imagery 3	56.57	36.34	34.07	62.43	33.59	40.57
	Physical 1	57.00	33.20	32.89	55.41	38.68	39.84

Table A2. Comparison of the sensor sets, in which three different kinds of mental tasks are used to compare their performances. One sensor set is located above the parietal lobe and the other one above the frontal lobe of the brain. The table shows the accuracy scores on the train, validation and test set obtained with Support Vector Machines (SVM) and Logistic Regression (LR).

Subject	Mental task	Date	*n* recordings	SVM			LR		
				Train	Val	Test	Train	Val	Test
1	Motor Imagery 1	11-02-2021	3	51.96	29.03	34.78	52.53	33.33	37.77
	Motor Imagery 1	11-26-2021	18	34.12	33.02	33.38	37.34	33.01	33.17
	Motor Imagery 2	12-08-2021	3	-	-	-	-	-	-
	Motor Imagery 2b	01-11-2022	2	38.24	35.02	34.40	55.40	45.23	37.85
	Motor Imagery 2b	01-18-2022	1	60.80	38.75	38.14	61.81	38.40	35.70
	Motor Imagery 3	01-11-2022	2	57.34	30.59	38.05	61.41	30.72	39.92
	Physical 1	11-02-2021	3	56.63	37.04	41.18	55.41	38.68	39.84
	Physical 2	12-08-2021	2	71.38	50.26	43.08	74.03	45.82	45.54
	Physical 2	01-18-2022	4	48.17	39.90	38.90	58.98	44.70	41.56
2	Motor Imagery 1	11-10-2021	2	54.59	35.92	29.71	56.93	37.37	27.43
	Motor Imagery 1	12-08-2021	1	67.88	45.32	23.86	75.41	41.61	25.16
	Motor Imagery 2	01-11-2022	2	58.68	43.99	35.46	59.22	41.30	36.51
	Motor Imagery 2b	01-11-2022	2	34.90	38.32	30.05	51.47	40.64	29.22
	Motor Imagery 2b	01-18-2022	1	64.78	48.93	54.79	65.36	47.29	52.72
	Motor Imagery 3	12-08-2021	2	59.93	33.67	40.05	62.43	33.59	40.57
	Motor Imagery 3b	01-11-2022	1	64.38	43.86	45.88	65.18	42.09	44.97
	Physical 1	11-10-2021	2	58.86	37.67	37.12	59.48	37.67	37.25
	Physical 2	01-18-2022	3	64.90	38.75	38.14	71.64	36.19	37.75
	Physical 3	12-08-2021	2	61.85	43.27	37.24	65.38	43.79	34.79

Table A3. Comparison of the performance of the recording sessions. For the dataset where no values are shown, the Common Spatial Patterns (CSP) model was not able to converge. The two datasets of both the Motor Imagery and physical tasks with the highest minimal accuracy score on the validation set, have been marked in blue.

Preprocessing methods	SVM			LR		
	Train	Val	Test	Train	Val	Test
CSP	90.12	45.14	46.63	56.06	39.72	23.58
CSP + statistical features	66.86	38.64	35.71	68.52	36.81	37.52
FFT	65.69	35.25	38.86	42.85	35.08	35.23
DWT	79.84	35.68	46.37	42.42	43.08	34.84
DWT + statistical features	47.12	31.90	44.82	54.43	33.62	40.67
CSP + DWT	85.49	38.18	43.13	45.75	36.54	26.81
CSP + DWT + statistical features	64.78	48.93	54.79	65.36	47.29	52.72

Table A4. Comparison of the different preprocessing methods. The table shows the accuracy scores on the train, validation and test set obtained with Support Vector Machines (SVM) and Logistic Regression (LR). The experiment was carried out on the 'Motor Imagery 2b 2022-01-18' recording session. The highest accuracy scores on the validation set, have been marked in blue.

Regularisation method	SVM			LR		
	Train	Val	Test	Train	Val	Test
None	64.78	48.93	54.79	65.36	47.29	52.72
Empirical	55.59	51.50	55.96	61.53	49.53	54.53
Diagonal Fixed	45.67	41.36	36.53	58.41	38.87	50.78
Ledoit Wolf	55.05	50.30	54.79	64.24	48.58	53.37
PCA	51.00	45.14	51.30	58.96	43.59	53.24

Table A5. Comparison of the different regularisation methods for the Common Spatial Patterns (CSP). The table shows the accuracy scores on the train, validation and test set obtained with Support Vector Machines (SVM) and Logistic Regression (LR). The experiment was carried out on the 'Motor Imagery 2b 2022-01-18' recording session. The highest accuracy scores on the validation set, have been marked in blue.

***n* components**	SVM			LR		
	Train	Val	Test	Train	Val	Test
1	34.38	35.17	37.18	40.17	33.28	44.95
2	34.85	37.66	28.37	48.06	34.57	44.95
3	55.16	48.93	54.79	53.28	46.09	53.37
4	53.20	49.70	55.70	53.89	49.36	55.83
5	54.14	50.33	56.48	56.75	49.10	55.44
6	60.69	50.21	55.96	58.31	49.14	53.37
7	55.74	49.42	57.12	63.48	48.32	54.15
8	55.59	51.50	55.96	61.53	49.53	54.53

Table A6. Comparison of the different number of components in the Common Spatial Patterns (CSP). The table shows the accuracy scores on the train, validation and test set obtained with Support Vector Machines (SVM) and Logistic Regression (LR). The experiment was carried out on the 'Motor Imagery 2b 2022-01-18' recording session.

α	Max iter	Train	Val	Test
0.001	250	32.75	33.36	33.42
	500	32.79	33.28	33.42
	750	32.79	33.28	33.42
	1000	34.60	33.45	33.16
	1250	34.60	33.45	33.16
	1500	34.60	33.45	33.16
0.01	250	32.75	33.36	33.42
	500	32.75	33.36	33.42
	750	32.75	33.28	33.55
	1000	37.86	42.56	44.82
	1250	37.86	42.56	44.82
	1500	37.86	42.56	44.82
0.1	250	32.75	33.28	33.55
	500	32.75	33.28	33.55
	750	33.55	33.28	34.59
	1000	52.01	49.70	57.25
	1250	52.01	49.70	57.25
	1500	52.01	49.70	57.25
1.0	250	33.04	30.27	33.68
	500	33.01	32.16	34.33
	750	55.99	46.43	56.22
	1000	59.10	48.75	57.51
	1250	59.61	48.75	56.48
	1500	59.57	48.84	56.74
2.0	250	34.02	27.77	35.88
	500	34.27	30.44	36.66
	750	55.19	41.53	46.89
	1000	62.83	47.72	53.50
	1250	63.55	48.24	55.44
	1500	63.55	48.32	56.22

Table A7. Comparison of the different combinations of learning rates (α) and the number of iterations used in the Support Vector Machines (SVM) model. The table shows the accuracy scores obtained on the train, validation and test set. The highest accuracy scores on the validation set, have been marked in blue.

α	Max iter	Train	Val	Test
0.001	250	33.28	33.87	33.16
	500	33.28	33.87	33.16
	750	33.28	33.87	33.16
	1000	33.28	33.87	33.16
	1250	33.28	33.87	33.16
	1500	33.28	33.87	33.16
0.01	250	49.10	50.52	57.12
	500	48.93	50.52	57.12
	750	48.93	50.52	56.99
	1000	48.93	50.56	57.12
	1250	49.18	50.67	56.74
	1500	49.27	50.86	56.61
0.1	250	48.24	51.28	55.31
	500	47.72	51.15	55.57
	750	48.15	51.40	55.70
	1000	48.07	51.59	55.70
	1250	47.81	51.55	55.70
	1500	47.81	51.59	55.96
1.0	250	47.21	51.49	54.92
	500	47.21	51.89	54.53
	750	47.03	51.78	54.40
	1000	46.60	51.66	55.05
	1250	46.26	51.61	55.05
	1500	46.52	51.76	55.05
2.0	250	47.21	51.53	54.92
	500	47.38	51.97	54.40
	750	46.95	51.76	54.40
	1000	46.43	51.63	55.05
	1250	46.17	51.55	54.92
	1500	46.52	51.83	54.92

Table A8. Comparison of the different combinations of learning rates (α) and the number of iterations used in the Logistic Regression (LR) model. The table shows the accuracy scores obtained on the train, validation and test set. The highest accuracy scores on the validation set, have been marked in blue.

References

[1] B. Kerous, F. Skola, and F. Liarokapis, "EEG-based BCI and video games: a progress report," *Virtual Reality 2017 22:2*, vol. 22, no. 2, pp. 119–135, Oct. 2017, doi: 10.1007/S10055-017-0328-X.

[2] T. I. Voznenko, E. v. Chepin, and G. A. Urvanov, "The Control System Based on Extended BCI for a Robotic Wheelchair," *Procedia Computer Science*, vol. 123, pp. 522–527, Jan. 2018, doi: 10.1016/J.PROCS.2018.01.079.

[3] A. J. Casson, "Wearable EEG and beyond," *Biomed Eng Lett*, vol. 9, no. 1, pp. 53–71, Feb. 2019, doi: 10.1007/S13534-018-00093-6.

[4] R. Portillo-Lara, B. Tahirbegi, C. A. R. Chapman, J. A. Goding, and R. A. Green, "Mind the gap: State-of-the-art technologies and applications for EEG-based brain-computer interfaces," *APL Bioengineering*, vol. 5, no. 3, Jul. 2021, doi: 10.1063/5.0047237.

[5] H. Yuan and B. He, "Brain-computer interfaces using sensorimotor rhythms: current state and future perspectives," *IEEE Trans Biomed Eng*, vol. 61, no. 5, p. 1425, 2014, doi: 10.1109/TBME.2014.2312397.

[6] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "EEG-based discrimination between imagination of right and left hand movement," *Electroencephalography and Clinical Neurophysiology*, vol. 103, no. 6, p. 642, Dec. 1997, doi: 10.1016/S0013-4694(97)00080-1.

[7] P. Wang, A. Jiang, X. Liu, J. Shang, and L. Zhang, "LSTM-based EEG classification in motor imagery tasks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 11, pp. 2086–2095, Nov. 2018, doi: 10.1109/TNSRE.2018.2876129.

[8] M. L. Martini, E. K. Oermann, N. L. Opie, F. Panov, T. Oxley, and K. Yaeger, "Sensor Modalities for Brain-Computer Interface Technology: A Comprehensive Literature Review," *Clinical Neurosurgery*, vol. 86, no. 2, pp. E108–E117, Feb. 2020, doi: 10.1093/NEUROS/NYZ286.

[9] H. Berger, "Über das Elektrenkephalogramm des Menschen," *Archiv für Psychiatrie und Nervenkrankheiten*, vol. 87, no. 1, Dec. 1929, doi: 10.1007/BF01797193.

[10] J. J. Vidal, "Toward direct brain-computer communication.," *Annu Rev Biophys Bioeng*, vol. 2, pp. 157–180, 1973, doi: 10.1146/ANNUREV.BB.02.060173.001105.

[11] J. W. Britton *et al.*, "Appendix 1. The Scientific Basis of EEG: Neurophysiology of EEG Generation in the Brain," in *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*, American Epilepsy Society, 2016.

[12] F. A. C. Azevedo *et al.*, "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain," *The Journal of Comparative Neurology*, vol. 513, no. 5, Apr. 2009, doi: 10.1002/cne.21974.

[13] P. Olejniczak, "Neurophysiologic Basis of EEG," *Journal of Clinical Neurophysiology*, vol. 23, no. 3, pp. 186–189, Jun. 2006, doi: 10.1097/01.wnp.0000220079.61973.6c.

[14] A. F. Jackson and D. J. Bolger, "The neurophysiological bases of EEG and EEG measurement: A review for the rest of us," *Psychophysiology*, vol. 51, no. 11, pp. 1061–1071, Nov. 2014, doi: 10.1111/PSYP.12283.

[15] J. Hawkins and S. Ahmad, "Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex," *Frontiers in Neural Circuits*, vol. 10, p. 23, Mar. 2016, doi: 10.3389/FNCIR.2016.00023/BIBTEX.

[16] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, "Chapter 2: Technological Basics of EEG Recording and Operation of Apparatus," *Introduction to EEG- and Speech-Based Emotion Recognition*, pp. 19–50, Jan. 2016, doi: 10.1016/B978-0-12-804490-2.00002-6.

[17] E. Habibzadeh Tonekabony Shad, M. Molinas, and T. Ytterdal, "Impedance and Noise of Passive and Active Dry EEG Electrodes: A Review," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 14565–14577, Dec. 2020, doi: 10.1109/JSEN.2020.3012394.

[18] Y. M. Chi, T. P. Jung, and G. Cauwenberghs, "Dry-contact and noncontact biopotential electrodes: Methodological review," *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 106–119, 2010, doi: 10.1109/RBME.2010.2084078.

[19] D. V. P. S. Murty *et al.*, "Gamma oscillations weaken with age in healthy elderly in human EEG," *Neuroimage*, vol. 215, no. 116826, Jul. 2020, doi: 10.1016/J.NEUROIMAGE.2020.116826.

[20] N. Houmani *et al.*, "Diagnosis of Alzheimer's disease with Electroencephalography in a differential framework," *PLOS ONE*, vol. 13, no. 3, Mar. 2018, doi: 10.1371/JOURNAL.PONE.0193607.

[21] N. Padfield, J. Zabalza, H. Zhao, V. Masero, and J. Ren, "EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges," *Sensors*, vol. 19, no. 6, p. 1423, Mar. 2019, doi: 10.3390/S19061423.

[22] D. J. Krusienski, D. J. McFarland, and J. R. Wolpaw, "An evaluation of autoregressive spectral estimation model order for brain-computer interface applications," *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pp. 1323–1326, 2006, doi: 10.1109/IEMBS.2006.259822.

[23] A. Schloegl, K. Lugger, and G. Pfurtscheller, "Using adaptive autoregressive parameters for a brain-computer-interface experiment," *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, vol. 4, pp. 1533–1535, 1997, doi: 10.1109/IEMBS.1997.757002.

[24] G. Rodŕiguez-Beŕmudez and P. J. Gárcia-Laencina, "Automatic and Adaptive Classification of Electroencephalographic Signals for Brain Computer Interfaces," *Journal of Medical Systems 2012 36:1*, vol. 36, no. 1, pp. 51–63, Nov. 2012, doi: 10.1007/S10916-012-9893-4.

[25] M. Hamedi, S. H. Salleh, A. M. Noor, and I. Mohammad-Rezazadeh, "Neural network-based three-class motor imagery classification using time-domain features for BCI applications," *2014 IEEE Region 10 Symposium*, pp. 204–207, Jul. 2014, doi: 10.1109/TENCONSPRING.2014.6863026.

[26] V. P. Oikonomou, K. Georgiadis, G. Liaros, S. Nikolopoulos, and I. Kompatsiaris, "A Comparison Study on EEG Signal Processing Techniques Using Motor Imagery EEG Data," *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 781–786, Jun. 2017, doi: 10.1109/CBMS.2017.113.

[27] A. Liu, K. Chen, Q. Liu, Q. Ai, Y. Xie, and A. Chen, "Feature Selection for Motor Imagery EEG Classification Based on Firefly Algorithm and Learning Automata," *Sensors 2017, Vol. 17, Page 2576*, vol. 17, no. 11, p. 2576, Nov. 2017, doi: 10.3390/S17112576.

[28] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *Journal of Neural Engineering*, vol. 14, no. 1, p. 016003, Nov. 2016, doi: 10.1088/1741-2560/14/1/016003.

[29] Y. Wang, X. Li, H. Li, C. Shao, L. Ying, and S. Wu, "Feature extraction of motor imagery electroencephalography based on time-frequency-space domains," *Journal of Biomedical Engineering*, vol. 31, no. 5, pp. 955–961, Oct. 2014, Available: https://europepmc.org/article/med/25764703.

[30] B. G. Xu, A. G. Song, and S. M. Fei, "Feature extraction and classification of EEG in online brain-computer interface," *Acta Electronica*, vol. 39, no. 5, pp. 1025–1030, 2011, Available: https://en.cnki.com.cn/Article_en/CJFDTotal-DZXU201105007.htm.

[31] Y. Zhang, Y. Wang, J. Jin, and X. Wang, "Sparse Bayesian learning for obtaining sparsity of EEG frequency bands based feature vectors in motor imagery classification," *International Journal of Neural Systems*, vol. 27, no. 2, Mar. 2017, doi: 10.1142/S0129065716500325.

[32] S. Kumar, A. Sharma, and T. Tsunoda, "An improved discriminative filter bank selection approach for motor imagery EEG signal classification using mutual information," *BMC Bioinformatics*, vol. 18, no. 16, pp. 125–137, Dec. 2017, doi: 10.1186/S12859-017-1964-6/FIGURES/6.

[33] S. Lemm, B. Blankertz, G. Curio, and K. R. Müller, "Spatio-spectral filters for improving the classification of single trial EEG," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 9, pp. 1541–1548, Sep. 2005, doi: 10.1109/TBME.2005.851521.

[34] A. Singh *et al.*, "A Comprehensive Review on Critical Issues and Possible Solutions of Motor Imagery Based Electroencephalography Brain-Computer Interface," *Sensors*, vol. 21, no. 6, p. 2173, Mar. 2021, doi: 10.3390/S21062173.

[35] A. Al-Saegh, S. A. Dawwd, and J. M. Abdul-Jabbar, "Deep learning for motor imagery EEG-based classification: A review," *Biomedical Signal Processing and Control*, vol. 63, Jan. 2021, doi: 10.1016/J.BSPC.2020.102172.

[36] M. Z. Baig, N. Aslam, H. P. H. Shum, and L. Zhang, "Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG," *Expert Systems with Applications*, vol. 90, pp. 184–195, Dec. 2017, doi: 10.1016/J.ESWA.2017.07.033.

[37] V. P. Oikonomou, K. Georgiadis, G. Liaros, S. Nikolopoulos, and I. Kompatsiaris, "A Comparison Study on EEG Signal Processing Techniques Using Motor Imagery EEG Data," *Proceedings - IEEE Symposium on Computer-Based Medical Systems*, pp. 781–786, Nov. 2017, doi: 10.1109/CBMS.2017.113.

[38] M. Z. Ilyas, P. Saad, M. I. Ahmad, and A. R. I. Ghani, "Classification of EEG signals for brain-computer interface applications: Performance comparison," *Proceedings of 2016 International Conference on Robotics, Automation and Sciences, ICORAS 2016*, Mar. 2017, doi: 10.1109/ICORAS.2016.7872610.

[39] D. Cheng, Y. Liu, and L. Zhang, "Exploring Motor Imagery EEG Patterns for Stroke Patients with Deep Neural Networks," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, pp. 2561–2565, Sep. 2018, doi: 10.1109/ICASSP.2018.8461525.

[40] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2015-November, pp. 2620–2623, Nov. 2015, doi: 10.1109/EMBC.2015.7318929.

[41] N. Lu, T. Li, X. Ren, and H. Miao, "A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 566–576, Jun. 2017, doi: 10.1109/TNSRE.2016.2601240.

[42] Z. Tang, C. Li, and S. Sun, "Single-trial EEG classification of motor imagery using deep convolutional neural networks," *Optik (Stuttg)*, vol. 130, pp. 11–18, Feb. 2017, doi: 10.1016/J.IJLEO.2016.10.117.

[43] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, Nov. 2015, Available: https://arxiv.org/abs/1511.06448v3.

[44] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017, doi: 10.1002/HBM.23730.

[45] A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 427–436, 2015.

[46] F. M. Garcia-Moreno, M. Bermudez-Edo, J. L. Garrido, and M. J. Rodríguez-Fórtiz, "Reducing Response Time in Motor Imagery Using A Headband and Deep Learning," *Sensors (Basel)*, vol. 20, no. 23, pp. 1–18, Nov. 2020, doi: 10.3390/S20236730.

[47] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," Mar. 2018, doi: 10.48550/arxiv.1803.01271.

[48] T. M. Ingolfsson, M. Hersche, X. Wang, N. Kobayashi, L. Cavigelli, and L. Benini, "EEG-TCNet: An Accurate Temporal Convolutional Network for Embedded Motor-Imagery Brain-Machine Interfaces," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2020-October, pp. 2958–2965, May 2020, doi: 10.48550/arxiv.2006.00622.

[49] A. Parfenov, "BrainFlow," 2021. https://brainflow.org/.

[50] "All-in-One Biosensing R&D Bundle – OpenBCI Online Store." https://shop.openbci.com/collections/frontpage/products/all-in-one-biosensing-r-d-bundle?variant=13043151994952.

[51] A. Roc *et al.*, "A review of user training methods in brain computer interfaces based on mental tasks," *J Neural Eng*, vol. 18, no. 1, Feb. 2020, doi: 10.1088/1741-2552/ABCA17.

[52] M. Sazgar and M. G. Young, "EEG Artifacts," in *Absolute Epilepsy and EEG Rotation Review*, Cham: Springer International Publishing, 2019, pp. 149–162.

[53] S. L. Bressler and M. Ding, "Event-Related Potentials," *Wiley Encyclopedia of Biomedical Engineering*, Apr. 2006, doi: 10.1002/9780471740360.EBS0455.

[54] A. Kübler, D. Mattia, R. Rupp, and M. Tangermann, "Facing the challenge: bringing brain-computer interfaces to end-users," *Artif Intell Med*, vol. 59, no. 2, pp. 55–60, Oct. 2013, doi: 10.1016/J.ARTMED.2013.08.002.

[55] R. Bousseta, I. el Ouakouak, M. Gharbi, and F. Regragui, "EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought," *IRBM*, vol. 39, no. 2, pp. 129–135, Apr. 2018, doi: 10.1016/J.IRBM.2018.02.001.

[56] N. Lu, T. Yin, and X. Jing, "A temporal convolution network solution for eeg motor imagery classification," *Proceedings - 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019*, pp. 796–799, Oct. 2019, doi: 10.1109/BIBE.2019.00148.

[57] M. Aasvik, "Simple High-pass, Band-pass and Band-stop Filtering," Mar. 10, 2016. https://www.norwegiancreations.com/2016/03/arduino-tutorial-simple-high-pass-band-pass-and-band-stop-filtering/.

[58] P. L. Nunez *et al.*, "EEG coherency. I: Statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales," *Electroencephalogr Clin Neurophysiol*, vol. 103, no. 5, pp. 499–515, 1997, doi: 10.1016/S0013-4694(97)00066-7.

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

[60] D. T. Pham, "Joint Approximate Diagonalization of Positive Definite Hermitian Matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 4, pp. 1136–1152, Jul. 2006, doi: 10.1137/S089547980035689X.

[61] Z. J. Koles, "The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG," *Electroencephalography and Clinical Neurophysiology*, vol. 79, no. 6, pp. 440–447, Dec. 1991, doi: 10.1016/0013-4694(91)90163-X.

[62] E. O. Brigham, "The fast fourier transform (fft)," in *The fast Fourier transform and its applications*, Prentice Hall, 1988, pp. 131–164.

[63] O. Trekhleb, "Playing with Discrete Fourier Transform Algorithm in JavaScript - DEV Community," Aug. 17, 2018. https://dev.to/trekhleb/playing-with-discrete-fourier-transform-algorithm-in-javascript-53n5.

[64] D. Gabor, "Theory of communication. Part 3: Frequency compression and expansion," *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 445–457, Nov. 1946, doi: 10.1049/JI-3-2.1946.0076.

[65] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Mathematische Annalen 1910 69:3*, vol. 69, no. 3, pp. 331–371, Sep. 1910, doi: 10.1007/BF01456326.

[66] M. Baratchi, "Urban Computing - Processing Time-series Data." p. 10, Available: https://urbancomputing2020.github.io/docs/L2/UC-slides-2.pdf.

[67] I. Daubechies, “Ten Lectures on Wavelets,” *Ten Lectures on Wavelets*, Jan. 1992, doi: 10.1137/1.9781611970104.

[68] K. Wirsing, “Time Frequency Analysis of Wavelet and Fourier Transform,” *Wavelet Theory*, Feb. 2021, doi: 10.5772/INTECHOPEN.94521.

[69] M. F. Mridha, S. C. Das, M. M. Kabir, A. A. Lima, M. R. Islam, and Y. Watanobe, “Brain-Computer Interface: Advancement and Challenges,” *Sensors 2021, Vol. 21, Page 5746*, vol. 21, no. 17, p. 5746, Aug. 2021, doi: 10.3390/S21175746.

[70] M. Alimardani, S. Nishio, and H. Ishiguro, “Brain-Computer Interface and Motor Imagery Training: The Role of Visual Feedback and Embodiment,” *Evolving BCI Therapy - Engaging Brain State Dynamics*, Oct. 2018, doi: 10.5772/INTECHOPEN.78695.

[71] G. Zhang, V. Davoodnia, A. Sepas-Moghaddam, Y. Zhang, and A. Etemad, “Classification of Hand Movements from EEG Using a Deep Attention-Based LSTM Network,” *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3113–3122, Mar. 2020, doi: 10.1109/JSEN.2019.2956998.

[72] C. M. Bishop, “ Linear Models for Classification,” in *Pattern Recognition and Machine Learning*, Springer New York, 2006, pp. 179–224.

[73] C. M. Bishop, “Sparse Kernel Machines,” in *Pattern Recognition and Machine Learning*, Springer New York, 2006, pp. 325–358.

[74] M. Sharma, P. V. Achuth, D. Deb, S. D. Puthankattil, and U. R. Acharya, “An automated diagnosis of depression using three-channel bandwidth-duration localized wavelet filter bank with EEG signals,” *Cognitive Systems Research*, vol. 52, pp. 508–520, Dec. 2018, doi: 10.1016/j.cogsys.2018.07.010.

[75] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Mar. 2015, doi: 10.1109/TNNLS.2016.2582924.

[76] E. Thomas, M. Dyson, and M. Clerc, “An analysis of performance evaluation for motor-imagery based BCI,” *Journal of Neural Engineering*, vol. 10, no. 3, p. 031001, May 2013, doi: 10.1088/1741-2560/10/3/031001.

[77] D. Wu, Y. Xu, and B.-L. Lu, “Transfer Learning for EEG-Based Brain–Computer Interfaces: A Review of Progress Made Since 2016,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 1, pp. 4–19, Jul. 2020, doi: 10.1109/TCDS.2020.3007453.

[78] Z. Unnisa, S. Zia, U. M. Butt, S. Letchmunan, and S. Ilyas, “Ensemble Usage for Classification of EEG Signals A Review with Comparison,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12196 LNAI, pp. 189–208, 2020, doi: 10.1007/978-3-030-50353-6_14.