FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme/specialisation: Masters of Science and Engineering / Signal processing, Robot -and Health Technology	Spring 2023 Open
Author: Martin Gjerde	
Course responsible: Tom Ryen Supervisor(s): Trygve Christian Eftestøl, Kolbjørn Kallesten Brønnick	
Title: Signal processing of EEG data and AI assisted classification of emotional responses based on visual stimuli Credits: 30	
Keywords: Machine Learning, EEG, Cognitive Behaviour, Python, Neuroscience, Brain, Signal Processing	Pages: 82 + appendices and bibliography Stavanger June 15. 2023

Acknowledgements

I would like to dedicate this section to thank the University of Stavanger for providing me the opportunity to spend my last semester of this masters degree in dwelling into topics that are of highly personal interest.

I would also like to say thank you to my supervisor Trygve Eftestøl for good technical guidance, feedback and discussions during the past months. An additional thanks to Kolbjørn Kallesten Brønnick for additional help when it comes to the use of the cognitive lab at UiS and elaborating about necessary background information about cognitive behaviour and use of EEG.

A special thanks goes to all my fellow students, that have become close friends, for creating a great academic environment for intellectual conversations and social activities. You have made the years at UiS unforgettable.

Abstract

This report outlines the research conducted to explore on the topic of classification of human neurological data using machine learning models. The primary objective was to investigate alternative approaches for efficiently interpreting EEG data and test the possibilities for predicting human emotions. During the study, data was collected by recording the brain activity of volunteering respondents using electroencephalography.

These participants were exposed to visual stimuli in the purpose of provoking specific neural activity as a result of emotional responses in the brain. The collected data underwent traditional signal preprocessing techniques typically employed in EEG data analysis.

Subsequently, the filtered data was subjected to wavelet transformation, both with and without synchrosqueezing. Principal components analysis was used to perform dimensionality reduction on the resulting features extracted from the data. The final model achieved a prediction accuracy of 32% when classifying between eight different classes of emotional responses based on training data from three respondents.

Contents

Chapter 1

Introduction

This chapter gives an overview of the project's motivation, background and general project description.

1.1 History and common use of EEG

Electroencephalography, EEG for short, was first introduced to the public through the study papers of the German psychiatrist Hans Berger in 1929. In 1924, Berger performed the first EEG on a 17 year old boy that was under a neurosurgical operation. In his publication he defined the observations as alpha and beta waves. It was not until later years that these observations were specifically defined as neural activity by American EEG scientists, as Berger previously only dismissed these as artefacts. These findings were a great breakthrough for the use of advanced technology in understanding the human brain and people's cognitive behaviour [1].

EEG is more commonly used today by psychiatrists and other medical personnel for diagnosing and correctly treating patients with brain disorders such as; brain tumour, sleeping disorders, epilepsy and stroke [2].

Scientists and researchers are today exploring how to utilise emerging technology to both process and analyse signals from EEG. The different explo-

ration topics varies from use of artificial neural networks to augment signals in EEG data [3] and also use of features from EEG to predict cognitive behaviour and learning in brain development [4].

In this project, exploration has been done to see how a (machine learning)ML model will perform in an attempt to classify different EEG responses based on human emotions processed through the brain. Future professionals in neuroscience and psychology would benefit from a model with these characteristics, as it can serve as a helpful tool for interpreting a patient's EEG data.

1.2 Project's vision and motivation

As new technology is becoming more frequently observed as an application for physical medical purposes and gaining attention in media [5], this research project's objective will be to achieve more insight in the use of technology for challenges related to psychology and psychiatry. Today there is currently more focus on research in medical technology related to direct physical illnesses or use of other brain activity measurement methods that are; more time consuming, less versatile and far more expensive.

EEG data used during this project will be collected through the EEG equipment located in the cognitive lab at the University of Stavanger. Since all the data will be anonymised this publication will be defined as an open report. This will increase the amount of neural activity data available to be used for future research projects. Additionally, all signal processing methods and scripts will also be available through a public repository so it can be used for further educational purposes at UiS. Hopefully, the use of EEG can be more prominent in the future educational curriculum of signal processing subjects at the university and eventually contribute to a more interdisciplinary work environment across different study programmes.

Increasing the use and interest of the equipment accessible in the cognitive lab may lead to more research topics that can help solve issues or improve the process of diagnosing patient's disorders related to the human brain.

1.3 Project description

This project will be a part of research to determine if it is possible to make use of machine learning models to classify human's emotional responses based on visual stimuli. With use of EEG, data will be collected from volunteering respondents that will be exposed to images from a standardised image set. When a stimulus evokes an emotional response or generates arousal, it is reflected as changes in large-scale neural networks in the brain. Electroencephalography can be used to measure the brain's activity by measuring the electric charge between synapses when the neurons are stimulated. The main achievement from the results of this project will be to see if it is possible to classify people's EEG responses and if they are finding what they look at appealing to them or not.

One of the challenges in this thesis will be to make sufficient use of known signal processing methods to be able to extract the correct information based on the literature available. Then the signal will be segmented, dividing the different emotional responses from one another. By later examining the resulting filtered data in more detail, hopefully a pattern in the EEG segments will be recognized and make it possible to correlate to the specific themed image that the respondent is looking at. These patterns will then be extracted and hopefully be predicted by a trained machine learning model [6].

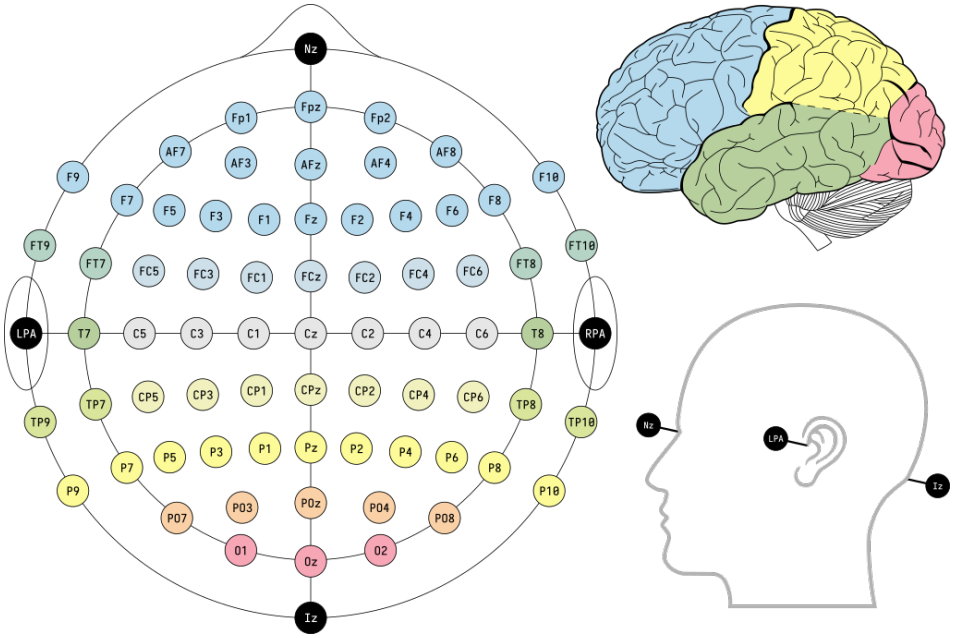


Figure 1.1: Example of EEG electrode protocol map and the four main lobes of the cerebral hemisphere; occipital lobe coloured in red, parietal lobe in yellow, frontal lobe in blue and temporal lobe in green [7].

The people used to collect data for the use of training the machine learning model will be exposed to images from open database libraries. Their reaction coming from the monitored EEG signals giving an indication of the electrical potential will be stored and used in the training of the machine learning model. For this project, code will be developed in Python to do the signal processing of the electrical signals coming from the EEG. This data will then be processed to extract specific features and fed into a machine learning algorithm to train an AI model that will be developed using open source resources, such as Sklearn.

Several methods are to be tested out to train the model varying on the range of the most tangible ones to find the most suitable to solve the classification problem. Examples of different classification algorithms can be random forest; based on decision trees, k-nearest neighbours or naïve bayes algorithm to determine if what the person is seeing in the image is of high or low valence [8].

Chapter 2

Theoretical study

This chapter will elaborate about the basic theory of the different topics relevant to this thesis project. First, the brain's anatomy and physiology in processing visual stimuli is discussed. The electroencephalography sensory system is explained in rough detail to get an understanding of what it is and how it works. Moreover, the signal processing pipeline of EEG signals is discussed on a mathematical basis. Lastly, the feature extraction and classification processes are explained on a certain level to better understand the results later in this report.

2.1 The brain's anatomy and physiology

2.1.1 The brain's anatomy and physiology

The brain is known to be a very complex system. The outermost layer of the brain is called the cerebral cortex, which has two hemispheres that can be divided into four lobes; the frontal lobe, the parietal lobe, the temporal lobe and the occipital lobe. These can be observed in 1.1. All of the lobes have responsibility for different functions, whereas the occipital lobe, located at the posterior part of the brain, is responsible for processing vision [9]. The way that the brain sends signals to different parts of the brain

is done through chemical reactions within and between neurons. A neuron is a cell that is found in the body's nervous system. The nervous system can be divided into two parts, the central nervous system(CNS) and the peripheral nervous system(PNS) [10]. Neurons receive nerve impulses, an electrical charge, through signal receptors connected to the nucleus of the cell called dendrites. This electrical charge occurs due to the difference in the cells' polarity [11]. To transfer nerve impulses further to other neurons, the neuron has one specific long fibre called an axon. When a signal reaches the end of the axon, it results in a stimulus that releases vesicles full of neurotransmitters that are being emitted through a gap from the presynaptic neuron to the postsynaptic neuron, called a synaptic terminal or synapse [12] [13] [14].

2.1.2 Neurological signals - brain's wavelengths

Electrical charge being transmitted between neurons differs between 6 main frequencies, based on the physiological functions, whereas low frequencies indicate a very relaxed state while higher frequencies indicate a more aware or distressed state. These states are named for their frequency ranges as described in Table 2.1 below [15].

Name	Frequency range [Hz]	Physiological function
Slow wave	0.1-1	NREM deep sleep
Delta	1-4	NREM deep sleep
Theta	4-10	Acquisition of navigation tasks
Alpha	8-15	Relaxed and passive attention
Beta	15-30	Wakefulness and REM sleep
Gamma	30-90	Associative learning

Table 2.1: Overview of the different brainwaves frequency ranges and physiological functions [15].

A representation of the most commonly analysed signals from an EEG recording separated into the 5 different frequency ranges is illustrated in Figure 2.1 [12]. Each signal in the figure reflects a typical EEG recording with the frequency ranges stated from Table 2.1.

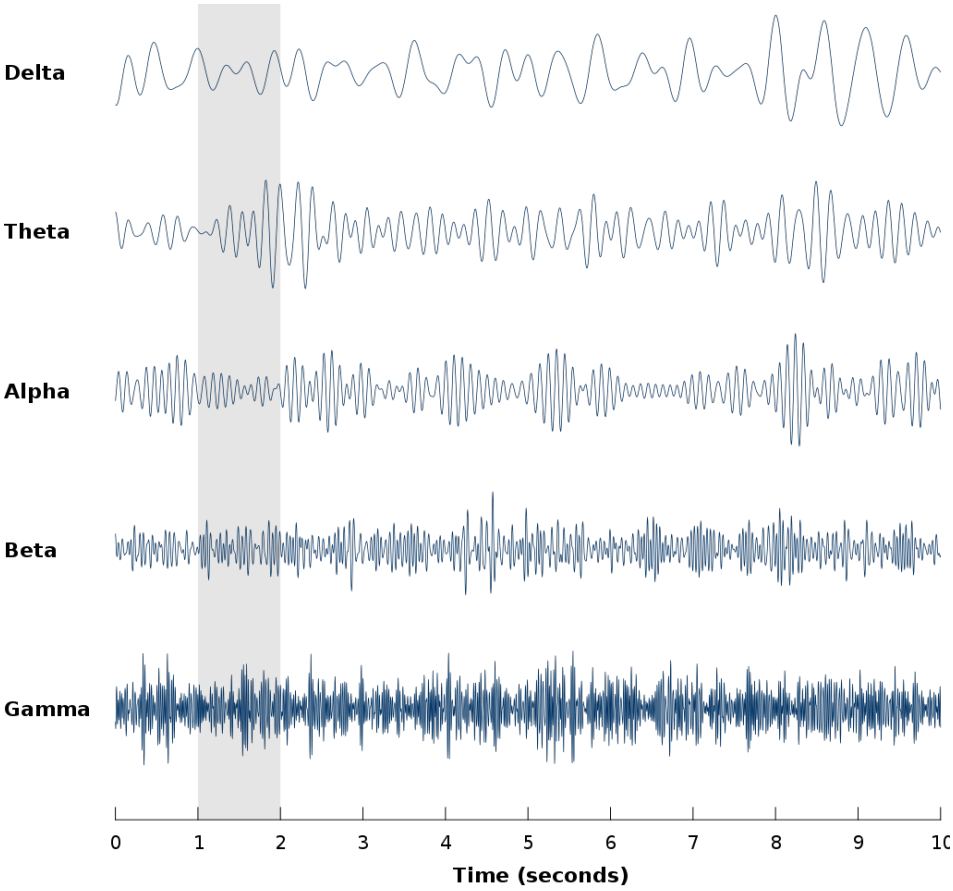


Figure 2.1: Illustration of the most commonly analysed EEG signals' waves [16].

In some researches where the respondent is awake during EEG recording, the delta waves are occasionally filtered out due to the fact that the data might contain unwanted artefacts such as eye blinking that has an average duration span of 250 to 400 milliseconds, or 0.25 and 0.4 seconds [17]. This can be calculated by simply using the frequency formula as defined in Equation 2.1 below.

$$f = 1/T \tag{2.1}$$

Where f is frequency in hertz and T is the period of a given time interval

in seconds [18].

The time period interval of an eye blink will then correspond to a frequency range between 2 and 4 Hz. Additionally, under certain circumstances, it is desirable to filter out the frequency range for delta waves because head movements may contribute to the occurrence of artefacts all the way down to 0.5Hz [19].

2.1.3 Vision and the visual cortexes

All information that is coming through the eyes and being processed by the brain goes through the visual cortex. As light goes through the pupil and then reaches the retina, located in the back of the eye, photo receptors transduce the light photons into a chemo electrical signal to the optical nerve. These photo receptors are called rods and cones, whereas the former is used to interpret shades of grey and the latter a perception of colours [20] [21].

The signal further goes through the optical nerve before it is split off in the thalamus where a low tract reaches the superior colliculi, accounting for primitive visual functions [22]. After passing through the mid-brain, the signal reaches the primary visual cortex located in the occipital lobe, coloured in blue in Figure 2.2.

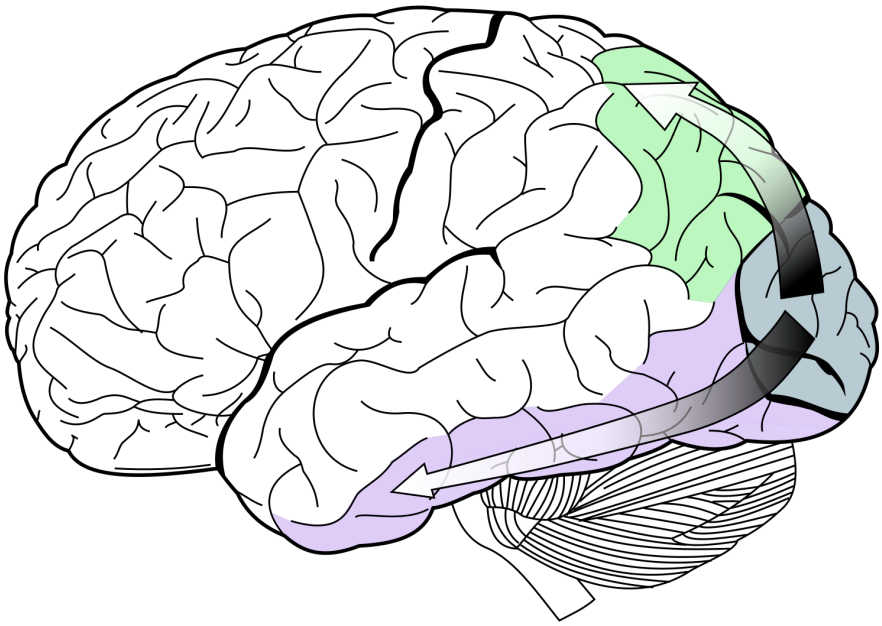


Figure 2.2: Illustration of the brain with the occipital lobe coloured in blue, ventral stream in purple and dorsal stream in green [23].

The visual cortex is divided into 5 main areas. In the primary visual cortex, also often referred to as V1, the information is simply relayed as object outlines, analog to edge detection in image processing. From the primary visual cortex, the information is further transmitted into two pathways called the ventral and dorsal stream. In the ventral stream the information is processed through the visual cortex V4 and leads further into the inferior temporal cortex, which is part of the temporal lobe [24] of the brain.

The area for the ventral stream can be allocated on the cerebral hemisphere as the purple coloured area in Figure 2.2. Information sent through the ventral stream is associated with memory stored in the brain, using stored information to interpret form and objects. This is due to the temporal lobe's communication with the hippocampus, which is important in forming long-term memories [25].The dorsal stream can be allocated as the green coloured area in Figure 2.2. Here the information is processed by the visual cortex V5, in addition the dorsal stream leads the information to the posterior

parietal cortex, part of the parietal lobe on the cerebral hemisphere. The dorsal stream processes information about object locations [26].

2.1.4 Emotions

Emotions are the brain's way to interpret stimuli based on previous sensory experiences correlated with similar stimuli. Emotion is a temporary perception that is processed by the limbic system in the body based on the sensory stimuli that triggers the production and secretion of hormonal substances that change the body's homeostasis, or balance point. The thalamus is located between the cerebral cortex' medial aspect, called the cingulate gyrus [27] and the limbic system in the part of the brain called the diencephalon. The thalamus forms a communication between these two areas of the brain [28]. The hippocampus, which is connected to the thalamus, helps to process this information and retrieve long-term memories from the amygdala. Long-term memories are associated with the information coming from the cerebral cortex and are compared with the information cognitively stored in the amygdala [29] [30] [31].

The information that goes into the amygdala is a two-way communication, where the hippocampus retrieves information from long-term memories to interpret the information from the cerebral cortex and at the same time provide information about current stimuli to the amygdala. Amygdala is a specific neural network in the limbic system [32]. By communication between the temporal lobe and the visual cortex, the information extracted from visual stimuli can be interpreted by the amygdala. The amygdala plays an important role of processing emotions, where new stimuli are associated together with previous sensory impressions with corresponding stimuli in memory. The information sent from the thalamus is also sent to the hypothalamus [33] [29] [25].

The hypothalamus functions as the centre for mediating the peripheral physiological reactions in emotions through signals coming from the sensory cells in the body [34]. These sensory cells can register changes or imbalances in the body and contribute to maintain homeostasis [35]. The pituitary gland, which is located below the hypothalamus, controls the endocrine system and is a gland that controls the production and secretion of hormones in this system [36].

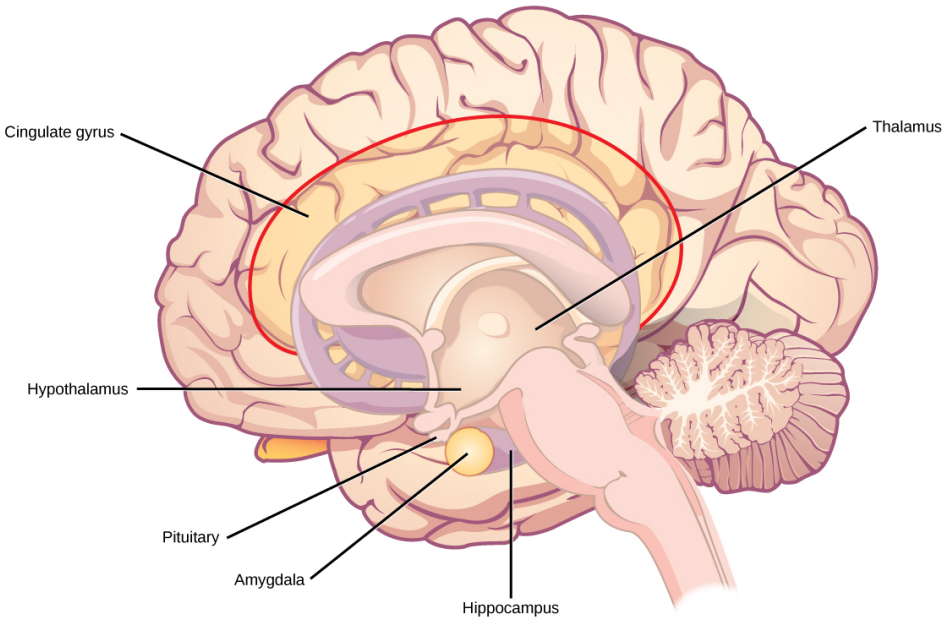


Figure 2.3: Cross section illustration of the brain, depicting the locations of the main parts in the limbic system [37].

When hypothalamus receives input from the sensory cells in the body, the pituitary gland can then be stimulated to secrete hormones. The release of hormones affects the intensity of emotional reactions. Stimuli of the pituitary gland from the hypothalamus occur via hormones, which can be e.g. dopamine. Dopamine is a neurotransmitter related to the body's reward system [38], and secretion of this neurotransmitter in the body will result in a positive sensation. This complex system composes a great part of the human cognitive functions [31] [39].

When the brain is processing information from the body's sensory system the pituitary gland can trigger hormone production in several organs of the body. Secretion of these hormones often leads to physical changes in the body, such as increase of heart rate or pupil dilation. The hormone producing organs are called endocrine organs, which are part of the endocrine system in the human body [40]. Both the hypothalamus and pituitary gland, mentioned above, are part of the endocrine system. The hormones that are released depends on what emotional response is processed in the brain that triggers the organs in the endocrine system [41] [42].

Valence measures if an emotion is positive or negative and how positive and negative it is. The higher the valence for a specific stimulus is for a person, the more positive associations the person has to that specific object, person, taste, smell or scenario. Little is to say about neutral feelings, as in psychology there is still not scientifically defined any neutral emotions, only negative and positive valence [43].

Arousal, or sexual arousal, is when sensory stimuli is processed through the hypothalamus in the limbic system and sends a signal through the spine and into the sacral area of the body. The signal goes further to the reproductive glands, which are part of the endocrine system, and secretes hormones such as androgen and oestrogen hormones. These specific hormones will affect the body and will lead to the feeling of arousal [44] [45].

The amount of different emotional feelings that a human can experience can be infinite. However there are methods to categorise most of these emotions. In 1980, a psychologist with specialised research in emotions and psychotherapeutic process, named Robert Plutchik, defined what is known as Plutchik's Wheel of Emotions. The wheel of emotions is a colour map that illustrates 8 main categories of emotions and what feelings that are linked to the specific emotion and the mixes of feelings [46].

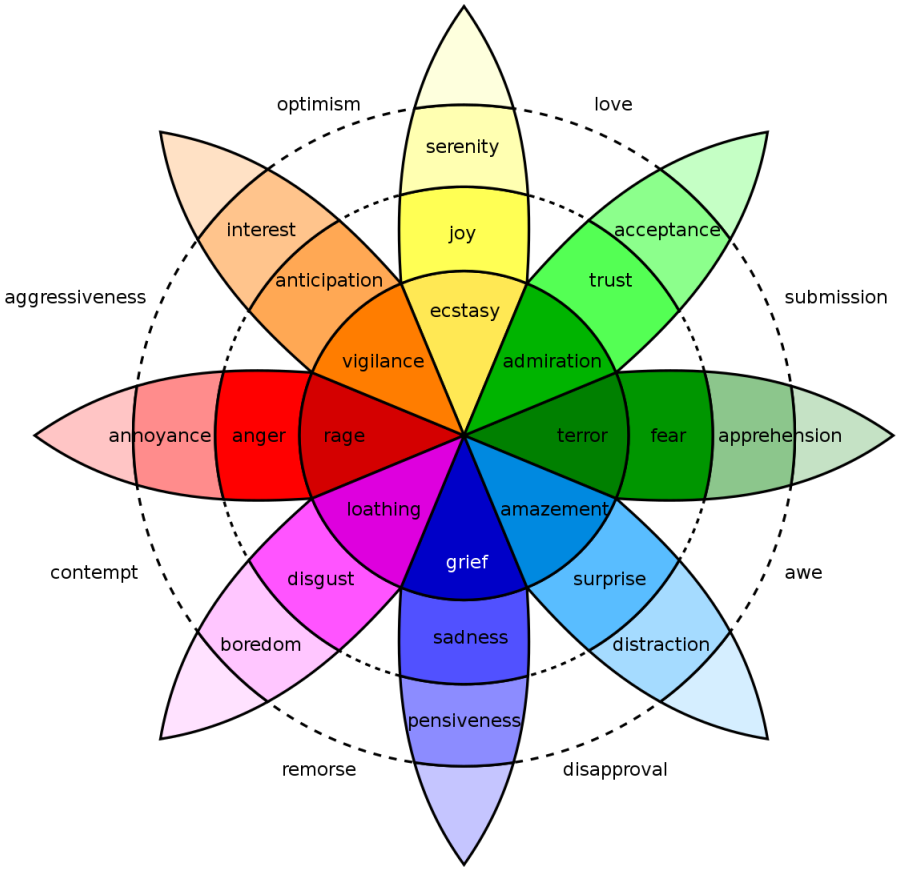


Figure 2.4: Plutchik's Wheel of Emotions [47].

2.2 Electroencephalography

As mentioned in Chapter 1, EEG is a technique to measure cortical activity. The greatest benefit of using EEG, compared to other neural activity measuring methods such as functional MRI(fMRI) and magnetoencephalography(MEG), is that electroencephalography is, in addition to being non-invasive to the patient, also; more versatile, less time consuming and is the cheapest procedure of the three. The data provided by an EEG has a high

temporal resolution, meaning high resolution with respect to time. By performing an EEG, it is possible to record the neural activity of the patient by attaching electrodes directly to the respondent's scalp. The electrical activity measured on the scalp is limited, but can still give a good indication of the state of a person's neural activity in different scenarios [48] [49].

Amplified neural activity measured from an EEG recording, varies normally between amplitudes of 20 and 100 microvolts [50]. Frequencies most commonly measured by an EEG of interest in analysis of brain waves are between 0.5-44Hz [51]. Electrodes used in an EEG must have sufficient contact with the skin to be able to acquire the neural signals that are obtainable on the scalp. To be able to accomplish this, a specific conductive gel is generally used to make contact between scalp and electrode. Other methods include using small needles, but this is less frequently used for research purposes. While an EEG is being recorded, the respondent is exposed to a certain stimuli for a given time period, based on what information that needs to be analysed after the data collection. Each electrode is placed on the scalp in specific positions on the head, so it will be possible to observe how different areas in the head are reacting to the present stimuli. Good EEG protocols are important to keep the recording efficient, reduce wear on equipment and to collect clean data [52].

As described in the previous section, neural activity is triggered by visual stimuli to process information and potentially resulting in an emotional response associated with the specific stimuli. This opens up the possibilities of recording these emotional responses through the use of EEG.

2.3 EEG signal processing theory

This section will elaborate about the theoretical background of some of the most common signal processing techniques of EEG signals.

2.3.1 Removing DC offsets

Early steps in most software for EEG signal processing include removing any DC offset in the data. This is to generate a zero mean value of the data for all channels [53]. Eliminating DC offset can be done by several different methods, where one method is computationally easier than the other. Subtracting the signal mean from each channel removes the DC offset and centres each individual signal at zero volts. This can be a preferred method to use for removing DC offset, because it increases the data accuracy and is preferred if the data is to be statistically analysed in later stages [54] [55]. Signal mean is generally calculated as shown in Equation 2.2 below, where n is number of samples and x is the signal [56].

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.2}$$

Another method is to perform baseline correction on the data set. Baseline correction is commonly used within spectroscopy and image analysis and enhancement [57]. Baseline correction uses statistical methods to calculate an estimator for the new baseline based on asymmetric weighting of deviations in the raw data [58]. Data scientists have developed several algorithms to perform these estimations implemented in open source code [59]. One of the algorithms is called the Zhang Fit algorithm which is an algorithm based on penalised least squares [60]. The statistical calculation of the penalised least squares function, $S_\lambda(f)$, that the algorithm is based on is shown in Equation 2.3 , where f is the functional space for projection, $\boldsymbol{x}_i$ are the smoothing variables, $\boldsymbol{y}_i$ has unknown relations to $\boldsymbol{x}_i$ and linear relation to the regression variables in $\boldsymbol{z}_i$, $\boldsymbol{\beta}$ is a parameter vector, λ is the smoothing parameter and $\boldsymbol{J_2}(f)$ is the roughness penalty often defined by the integral of $\left(\frac{d^2 f(x)}{dx^2}\right)^2$ [61].

$$\boldsymbol{S}_\lambda(f) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i - f(\boldsymbol{x}_i) - \boldsymbol{z}_i\boldsymbol{\beta})^2 + \lambda \boldsymbol{J_2}(f) \tag{2.3}$$

2.3.2 Identifying and removing bad channel data

When analysing EEG data with the DC offset removed it is easier to identify channels that deviate in time-domain when comparing them. These signals are usually defined as bad channel data and are normally detected by manual analysis through plotting the data and visually inspecting the data set [62]. When using the zero mean subtraction method to reject DC offsets, statistical operations can also be done to automatically identify these bad channels [63]. Insufficient contact between the measuring electrode and the scalp during an EEG recording might result in bad channel data. In most cases where bad channel data is observed, the data is simply rejected from the data set by manual analysis [64]. It is also possible to automatically detect bad channels by using statistical approaches. One of the alternatives is to make use of an adjustable threshold parameter [65] that multiplies with the standard deviation of the data set and results in a value used for identifying potential outliers. The signals that are not within the range of the calculated value will be marked as bad channels. The mathematical equation for calculating standard deviation is replicated in Equation 2.4 below, where x is the signal data, $\overline{x}$ is the signal data mean and n is the number of samples in the signal [66].

$$s = \sqrt{\frac{\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{x})^2}{n-1}} \tag{2.4}$$

After the channel outliers are detected the channel data can simply be removed from the data set, which is a common approach to treat bad channels when looking at singular respondents. The reason why bad channels are commonly rejected from the data set is because malfunctioning channels may have zero variance, which may lead to dramatic shrinking of cortical estimates, such as the standard deviation. The calculation of variance is shown in Equation 2.5, which is the same as Equation 2.4 squared [63] [67].

$$s^2 = \frac{\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{x})^2}{n-1} \tag{2.5}$$

When doing an EEG analysis based on ERP, the zero variance will distort

the group averaging of the signals with its low noise estimate. However, if looking at data across multiple respondents, it may be of significance to keep the data dimensions equal across respondents. Obtaining equal data dimensions can be done by reconstructing the signals with interpolation of neighbouring channel data [67].

2.3.3 Interpolation of bad channel data

Interpolation is a method used to achieve the same quantity of data for every data set across respondents in the same study after rejecting the bad channel data. This is to achieve the same dimensions on the data for every subject. Frequently, interpolation algorithms produce an approximated signal as an alternative to the bad channel data using neighbouring channels [67].

Interpolation is commonly used within both one dimensional signal processing as well as for two dimensional signal processing for images. Many interpolation algorithms are accessible through open sources such as Python's interpolate and signal package indexes [68] [69].

The most tangible interpolation method is linear interpolation. This method creates a direct line between approximated signal points of the interpolated signal to generate the new signal. This method is based on the polynomial defined in Equation 2.6 based on calculating a function's slope between two signal points [70].

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \tag{2.6}$$

Another example of an interpolation method is the B-spline algorithm. This is a more advanced technique to perform the signal approximation where the new signal has smooth curves instead of straight lines between the interpolation samples. The new signal using the B-spline algorithm is approximated based on Equation 2.7, where c_j are coefficient values, β is the basis function of order o and Δx is the equal spacing between signal points in x [68].

$$y(x) \approx \sum_{j} c_j \beta^o(\frac{x}{\Delta x} - j) \tag{2.7}$$

2.3.4 Data filtering

Before a complete analysis can be performed on signal data, it is important to perform signal filtering. Filtering is a common digital signal processing method that removes unwanted components that occur under data collection. There are two common types of filtering methods that are categorised based on their impulse response and can be defined through convolution in the time domain [71]. The two types of filters are called Finite Impulse response FIR, and Infinite Impulse Response. The main difference between these filters is that the FIR filter's response results in a finite sum, as its output is defined by present and past filter input values as defined through convolution in Equation 2.8 where x is the signal and h is the filter.

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{2.8}$$

Meanwhile an IIR filter's response results in an infinite sum, as its output is not only based on past and present input values, but also the filter's own output. IIR filters can generally be expressed through convolution as defined in Equation 2.9, where x is the signal and h is the filter.

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{2.9}$$

When filtering a signal a phase shift or delay in signal is introduced. The time delay or phase shift is often higher in a FIR filter than an IIR filter. IIR filters often have a higher resolution for low frequencies than the FIR filter. IIR filters can therefore be a preferred filtering method if the low frequency components in a signal are of interest [72] [73] [74].

Filtering in the time domain through convolution can be computationally

difficult for larger data sets, therefore it is common to perform a transformation of the signal from time domain into frequency domain. For certain transformations, mathematical operations become significantly easier to compute. An example is that the time domain convolution equals simple multiplication in the frequency domain. One of these transformations is called the discrete time Fourier transform and is defined through convolution of a signal and a complex exponential as shown in Equation 2.10, where ω is the frequency [75] [76].

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{2.10}$$

After filtering in the frequency domain, the filtered signal can be transformed back into the time domain through the inverse discrete time Fourier transform as defined in Equation 2.11 [77].

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega \tag{2.11}$$

Band pass filtering is a method used to preserve signals within a certain range of frequencies by amplifying the frequencies of interest and attenuating all the other frequencies outside of this range. This range of frequencies is usually called the filter's bandwidth. A band pass filter can be seen as a combination of a high pass and low pass filter, whereas a high pass filter attenuates the unwanted low frequencies and a low pass filter attenuates the unwanted high frequencies. The outer boundaries of a filter's bandwidth are called the lower and upper cut-off frequencies. Frequencies within the bandwidth will be preserved when using band pass filtering. A cut-off frequency is defined as when the output signal equals $\frac{1}{\sqrt{2}}$, or -3dB, of the input signal [78]. Figure 2.5 below illustrates a band pass filter's characteristics.

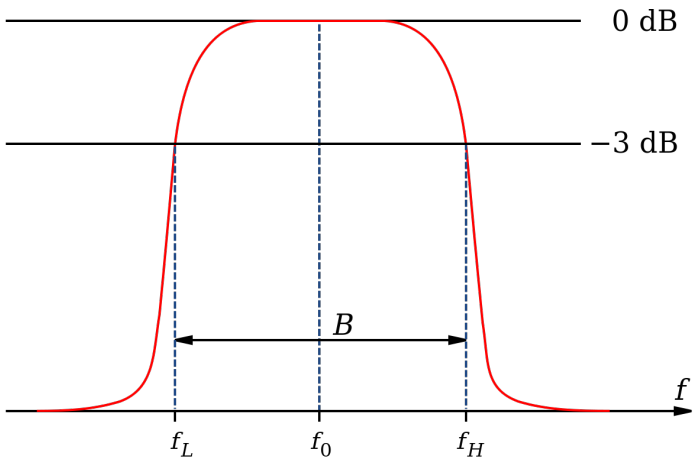


Figure 2.5: Illustration of a band pass filter with its center frequency f_0, lower and upper cut-off frequencies, f_L and f_H, and the filter's bandwidth B [79].

Band pass filters can be categorised by their Q-factor, whereas a high Q factor implies a narrow bandwidth, or small distance between the low pass and high pass filters cut-off frequencies, and a low Q factor implies a broad bandwidth. The Q-factor is calculated from the following Equation 2.12 [80] [81].

$$Q = \frac{f_H - f_L}{f_0} \tag{2.12}$$

The reason why most common filters have roll offs and not straight vertical lines at the cut-off frequencies between the preservation and attenuation is due to physical limitations. If using an ideal filter, which is defined as a filter with vertical roll offs at its' cut-off frequencies from the pass band(filter value = 1), the Gibbs phenomenon occurs. This phenomenon is defined as when oscillation is observed from a resonator in a signal or filter's output that is not present in its input after performing a Fourier transform on a signal with abrupt change in magnitude. Figure 2.6 illustrates this with an ideal low pass filter and its impulse response [82].

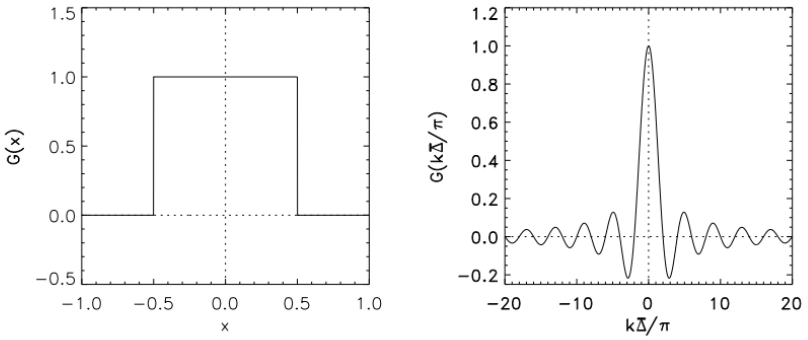


Figure 2.6: An ideal low pass filter and its impulse response [83].

Filters are therefore specifically designed to prevent ringing and at the same time attenuate the unwanted frequencies in the best way possible. One type of filter design is the Butterworth filter, first described by Stephen Butterworth in 1930, where he described how a filter can be normalised for design [84]. The Butterworth filter is known as the best filter when it comes to compromising between preserving and attenuating signals for frequency filtering. Thus this is a common choice of filter design. A Butterworth filter is also called a maximally flat filter due to it not introducing any ripples or ringing artefacts in its stop band nor pass band [85]. In addition, the Butterworth filter is known for having a more linear phase response, meaning less signal delay after filtering than other traditionally used filters. The general frequency response of a Butterworth filter is defined in the frequency domain as in Equation 2.13 below, where ω_c is the cut-off frequency and n is the order of the filter, which adjusts the steepness of the filter's amplitude response [86] [87].

$$H(j\omega) = \frac{1}{\sqrt{1 + \frac{\omega^{2n}}{\omega_c}}} \tag{2.13}$$

In certain cases it is desirable to remove components of a certain range of frequencies from a signal. This is done by band stop filtering. A band stop filter functions like a band pass filter, only that the cut off frequencies for the higher and upper bound are opposites, meaning that the frequencies within

the final filter's bandwidth will be attenuated while all other frequencies will be preserved. A specific type of band stop filter is called a notch filter which is a band pass filter with a high Q-factor. Notch filtering is commonly used in scenarios where the signal is greatly influenced with artefacts originated from the power line, corrupting the analysis of the underlying signal of interest [88] [89].

For analysis of EEG data, notch filtering might be necessary to perform in addition to conventional band pass filtering. This might be the case if the power line noise is prominent in the signal. Before performing band pass filtering on signals in EEG data, the signals should be notch filtered, this is because the high pass filtering used to get rid of DC shifts originating from low frequency components are prone to the high frequency signals and may cause the power line noise to be further amplified instead of attenuated [90].

2.3.5 Data epoching

Data collected from an EEG test usually has different types of stimuli to trigger neural activity. Normally, during an analysis of signals recorded using EEG, the different time intervals of responses to the stimuli the respondent is exposed to is segmented. This is called an event related potential analysis, or ERP for short. When performing an analysis on ERP, it is necessary to isolate the different electrical potentials to align them with the stimuli exposure. This is commonly known as data epoching. This is easily done to the data by simply segmenting the signals in the same given time intervals of the stimuli exposure. Before epoching and starting an ERP analysis, the EEG channels are averaged to get an unambiguous signal representation [91].

2.3.6 Bad epoch elimination

When the mean averaged signal is segmented into epochs, there might be certain epochs that contain specific unwanted artefacts that are not possible to eliminate using conventional signal processing filtering mentioned earlier in this section. These epochs are therefore not wanted to use in an ERP analysis and must be rejected as bad epochs. The simplest method

to identify the bad epochs is to plot and compare them through manual analysis. Typically it is possible to identify bad epochs through signal plots by observing which epochs contain abnormal high amplitudes in microvolts that do not comply with the general amplitude range of the other epochs. Bad epochs are simply epochs that are not representing the valuable underlying EEG data due to artefacts which may have been caused by several factors, such as the respondents making abrupt motions while recording. Some of the most common reasons for these signal artefacts are; electrocardiogram(ECG), electrooculography(EOG) and electromyogram(EMG) [92] [93] [94].

In addition to manual bad epoch identification there exists several methods to perform automatic identification and rejection of these bad epochs' artefacts. Two recognised methods, based on higher order statistics are Independent Components Analysis, ICA for short, or rejection of epochs based on kurtosis. The latter is based on using the statistical purposes of kurtosis, meaning a signal's peakyness to identify epoch outliers. As muscular artefacts normally have higher peaks than underlying true EEG signals, this method can be good to identify unwanted artefacts and mark the epochs for rejection [95].

ICA's algorithm is based on the theory that a signal is composed of multiple signal sources, which is true for EEG signals that have sources coming from the different artefact sources as well as the underlying true EEG from multiple electrodes. Criteria for using ICA is that all signal components need to be statistically independent and not be Gauss-distributed [96]. By separating a signal into individual components, it is possible to compare these and extract the specific unwanted component of the signals that are considered EEG artefacts and not involve these when summing the signals component back together. The biggest benefit of using ICA over pure epoch rejection is that ICA preserves the underlying EEG signal of interest and only rejects the specific artefact source, instead of losing the entire epoch itself, resulting in no loss of epochs in the data set [97].

2.4 Data representation and feature extraction

Typically one or more of the data representations' elements described in this section can be used as features to train a machine learning model to learn the properties of specific signals and distinguish them from one another in classification problems.

2.4.1 Periodograms

To get a better understanding of the signal data collected from an EEG recording it is often beneficial to investigate the signals properties and specific features. A common way to represent data characteristics is through signal periodograms. Periodograms are representations of the presence of different frequencies in the signal. A common method to show this is with use of the Fourier transform, as described in the previous section. Due to the DTFT operation demanding a lot of processing capacity due to its computational complexity, an alternative algorithm that is close to the original DTFT that is easier to compute is commonly used for discrete signal transformation. This is defined as the Fast Fourier Transform(FFT) and is defined as in Equation 2.14 [98] [99].

$$X(m) = \sum_{n=0}^{N-1} x_m(n) e^{-j\frac{2\pi n}{N}k}; k = 0, 1, ..., N-1 \tag{2.14}$$

Just like the DTFT, FFT also has a counterpart called the Inverse Fast Fourier Transform(IFFT) which is defined as in Equation 2.15 [100].

$$x_m(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) e^{j\frac{2\pi m}{N}k}; k = 0, 1, ..., N-1 \tag{2.15}$$

A periodogram categorises the frequency content in a signal by estimating the signal's frequency components [101], using the squared modulus or real component of the FFT and dividing over a window function's non-zero

samples. This is also called the Power Spectral Density Estimation, or PSD estimation of a signal, and is defined as shown in Equation 2.16, where x_m is the signal and M is the size of the window function used which is described in the next paragraph [102] [103].

$$P_{x_m,M}(\omega_k) = \frac{1}{M} |FFT_{N,k}(x_m)|^2 \triangleq \frac{1}{M} \left| \sum_{n=0}^{N-1} x_m(n) e^{-j\frac{2\pi n}{N}k} \right|^2 \tag{2.16}$$

When performing PSD estimation, different results can be expected based on the window functions that are used. A commonly used window function is called the Hann or Hanning window function based on the properties of a cosine wave [104]. The results of the Hann window are defined as in Equation 2.17 where $R_N(n)$ is a standard rectangular window function [105] [106].

$$w(n) = 0.5 \left(1 - \cos \frac{2\pi n}{N+1}\right) R_N(n); n = 1, 2, ..., N \tag{2.17}$$

2.4.2 Welch's method for power spectral density estimation

The periodogram may sometimes be difficult to interpret as it has limitations related to that it is a discrete function. One of the limitations of analysing a periodogram is the picket fence effect, PFE for short, which acts as white noise. The reason why this effect occurs is because the signal contains frequency components that are in between integer multiples of the sample frequency [107]. An alternative PSD estimation method that has immunity to PFE is the Welch estimation method. When using the Welch method, there is less distortion of the PSD function due to the smooth averaging of the frequency components as it results in smooth curves to represent the signal in the spectral domain [108]. The Welch method is based on averaging the traditional PSD over a time frame, resulting in elimination of the signal distortion introduced by the PFE. Welch method's output therefore gives a smoother PSD through mean filtering the signals, making it a preferred method over the periodogram when performing a spectral analysis of a signal. Mathematical definition of the Welch PSD estimation method

is defined as in Equation 2.18, whereas K is the number of frames used for averaging [99] [109].

$$\hat{S}_x^W(\omega_k) \triangleq \frac{1}{K} \sum_{m=0}^{K-1} P_{x_m,M}(\omega_k) \tag{2.18}$$

2.4.3 Scalogram analysis

Another method of representing signal data's characteristics is to use scalograms. A scalogram is a representation that uses heating maps to represent a signal's frequency density at a specific given time of the signal data. Scalograms make use of the wavelet function to extract and represent these properties. There exist two common ways to implement the wavelet function called the Discrete Wavelet Transform(DWT) and the Continuous Wavelet Transform(CWT). The wavelet transform uses a series of time localised small waves(wavelets) and performs convolution between the wavelets and the signal of interest. Due to the localisation of the sine wavelets, this transformation obtains both different frequency information and time information in the transformed signal. The series of wavelets is usually based on what is called a Mother Wavelet, which is the first standardised wavelet that the signal is processed with. Several standardised mother wavelets are defined and a popular wavelet function often used for analysing EEG data is called the Morlet wavelet, defined in Equation 2.19, where ω_0 is the center frequency [110] [111].

$$\psi(t) \triangleq \frac{1}{\sqrt{2\pi}} e^{-j\omega_0 t} e^{-\frac{t^2}{2}} \tag{2.19}$$

A series of alternate versions of the mother wavelet is then convoluted with the signal using scaling and translation factors to increase or decrease the wavelet frequency, and shift the localisation of the wavelet. The general equation of the Continuous Wavelet Transform is shown in Equation 2.20 below, where s is the scaling factor, x is the signal to be transformed and ψ is the mother wavelet [110].

$$CWT(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{t=-\infty}^{\infty} x(t)\psi\left(\frac{t-\tau}{s}\right) dt \tag{2.20}$$

The main difference between the DWT and CWT is that the DWT uses discrete scaling and translation factors, where the scaling is increasing in the power of 2 while the translation factors are simply whole integers. Both transforms are commonly used for denoising of data as well as for feature extraction to use in machine learning models. Time-frequency analysis can also be realised through the use of the Short Time Fourier Transform(STFT), but due to the STFT using linearly scaled sine waves which are not localised in the time domain it results in a uniform spectre. Hence, the wavelet transform is often preferred for time-frequency analysis and overperforms compared to other techniques for signal processing. Wavelet transformation results in high time domain resolution for high frequencies and high frequency resolution for low values in the time domain, making it possible to not only easily observe which frequencies are present in the signal, but also which time these frequencies occur in the signal [112].

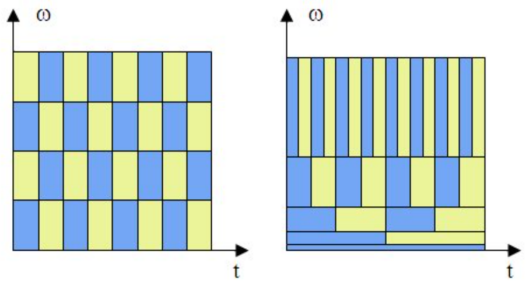
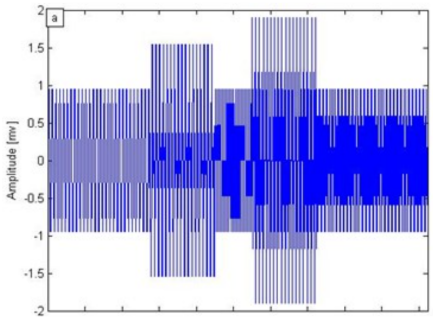


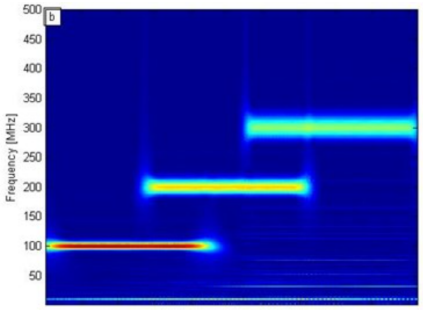
Figure 2.7: Simple illustration of spectrogram using STFT(left) and scalogram DWT(right) [113].

To enhance representation clarity and accuracy of signals in a time-frequency analysis synchrosqueezing is commonly performed on the data. This is a recognised method used in multi-component signals such as biomedical signal analysis. Synchrosqueezing redistributes the energy calcualted from the time-frequency transformation into corresponding frequency bins associated with the signal's computed instantaneous frequencies. This increases the frequency content in the different time instances

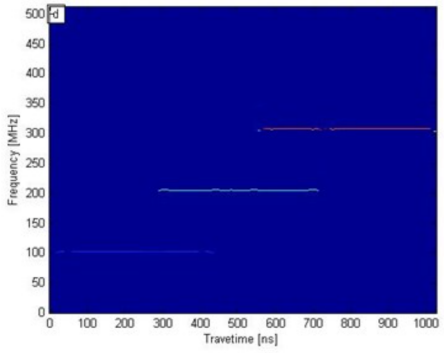
and higher resolution of the transform is achieved. Often, synchrosqueezing improve the characterisation of complex non-stationary signals, making it more feasible to analyse and use for machine learning models [114] [115]. Figure 2.8 illustrates the difference of a scalogram with the use of wavelet transform with and without synchrosqueezing.



(a) Signal in time domain.



(b) Scalogram of signal using wavelet transformation.



(c) Scalogram of signal using synchrosqueezed wavelet transformation.

Figure 2.8: Scalogram representation of a signal with and without synchrosqueezing [116].

2.5 Classification

Classification is a typical data analysis problem for a machine learning algorithm to streamline the grouping of large data that can be a time consuming process and not always easily analysed by humans. This section will discuss some algorithms used in modern multi-class classification problems and ways on how to verify model performance. Multi-class classification is a supervised machine learning method where there are more than two classes' features for a machine learning model to distinguish between based on the classes' labels [117].

2.5.1 Feature selection

Before developing a machine learning model and training it on the data based on the features that are extracted, it might be beneficial to perform operations to the data to optimise which features are to be used. This is called feature selection. Based mostly on statistical methods, common feature selectors operate to find distinct features for the different classes and reduce the amount of model input variables. The amount of features fetched from using feature selection may vary depending on the wanted model's response variables or predicted output [118].

A recognised method commonly used within machine learning and signal processing in general for dimensionality reduction is Principal Components Analysis, PCA for short. PCA reduces high-dimensional data sets, whilst preserving relationships and structure essential to the original data set. Performing PCA to a data set sorts standarised data into a covariance matrix, finds the eigenvectors and corresponding eigenvalues of the covariance matrix and sorts them in descending order based on their importance of data representation. When selecting the number of components to use for dimensionality reduction, a new matrix is generated based on the eigenvectors and reduces the dimensions into the size of the chosen number with the most critical values. The reason why dimensionality reduction is often performed prior to training a machine learning model is to avoid model over fitting, which is a result of the curse of dimensionality. This refers to the formal problem of analyzing data in high-dimensional spaces, where the data set may consist of thousands or more features. However, this analysis will no

longer be valid when applied to lower-dimensional spaces [119] [120]. In other words, if features from a specific data set is used to train a model too extensively, it may fail to predict correctly on other data sets. By including diverse and large data sets or adjust the model's complexity can decrease the level of model over fitting [121].

2.5.2 Machine learning algorithms

There is a multitude of ways to make a machine learn how to solve several problems. Some of the most commonly used in classification problems are shortly elaborated below.

2.5.2.1 Naïve Bayes' algorithm

The Naïve Bayes algorithm is based on Bayes' theorem, often recognised in probability statistics. This theorem bases on sequential events whereas new information influence the initial probability, known as the prior probability, which results in the posterior probability. The prior probability is the likelihood of an event before any conditions are set, meanwhile the posterior probability is the likelihood of an event occurring after observing a particular data point. Naïve Bayes' algorithm is given its name due to the assumptions the method has, which is that its predictors are conditionally independent to the other features and that the outcome is equally influenced by all features [122].

2.5.2.2 k-Nearest Neighbour algorithm

The k-Nearest Neighbour algorithm, or kNN for short, is an algorithm that assumes that data points of a specific class lies in some calculated proximity from other data points within the same class, called neighbours. Using well-known mathematical operations such as calculating euclidean distance between two data points, from the current data point to the potential same-class data point. If the potential data point is within a certain calculated distance away from the initial data point, it will be added as another neigh-

bour. An unclassified label is then predicted through the majority among its k nearest neighbours. The class with the highest frequency among the neighbors will be assigned as the predicted class. [123].

2.5.2.3 Random Forest algorithm

Random forest algorithm is based on a collection of decision trees. Decision trees divide data into finer subsets, often referred to as nodes, based on different criteria or conditions called decision rules. When the majority of a tree's subset recordings indicate a specific class, the decision tree estimator assigns its final prediction to that class. The random forest algorithm is an ensemble method, meaning it averages the results from all decision tree estimators. Tuning model parameters might be crucial to obtain a good model based on the random forest algorithm. These parameters can be the amount of total estimators or decision trees to use for averaging and the depth of each decision tree [124] [125].

Decision rules in decision trees are the conditions or criteria used to make decisions at each node of the tree. These rules determine how the data is split and how the tree progresses from the root node to the leaf nodes. Each internal node of a decision tree represents a decision rule based on one or more features, and each leaf node represents a final decision or outcome.

2.5.3 Evaluation metrics

To evaluate a machine learning model's performance, there exists several methods for numerically and visually represent these and are described below.

2.5.3.1 Confusion matrix

An easy way to visualise the model's performance can be done through a confusion matrix. The confusion matrix shows a grid of the actual classes vs. class predictions made by the model. The amount of predictions made by

the model for the specific class is often visualised using a heat map, where as the brightest colors shows the highest numbers of predicted class. In an ideal example, the brightest colors will appear on the matrix' diagonal, indicating high performance in the model's capability of identifying True Positives and True Negatives for a two-class classifier [126]. Figure 2.9 shows an example of a binary confusion matrix.

		Predicted class: Positive	Predicted class: Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

Figure 2.9: Binary confusion matrix, representing the True Positive(TP), True Negative(TN), False Positive(FP) and False Negative(FP) on a model's predictions vs. actual classes [127].

2.5.3.2 Classification report

Often evaluating if a machine learning model can be ready for use is to look at its classification report. A Classification report shows an overview of the model's performance based on several methods. Commonly, to consider if a model is good or not is by observing the parameters listed below [128] [129] [130].

1. Precision: Gives an indicative percentage on the model's performance in identifying a specific class correctly based on all predictions by the model. Meaning how many of the predictions that the model passed as correct was actually true for that class.
   This is calculated as: $\frac{TP}{TP+FP}$.
2. Recall: Gives an indicative percentage on the model's performance in correctly predicting a class based on all valid labels for that specific class.
   This is calculated as: $\frac{TP}{TP+FN}$.

3. f1-score: Is the harmonic mean between Precision and Recall, resulting in a mean between the two indicators' that promotes similar values for the two.
   This is calculated as: $\frac{TP}{TP+\frac{1}{2}FP+FN}$.

4. Accuracy: Gives an indicative percentage on the model's whole performance in correctly predicting the data.
   This is calculated as: $\frac{TP+TN}{TP+TN+FP+FN}$.

2.5.3.3 Cross-validation

During model training it is common to divide the training data into three different subsets: training set, test set and validation set. The largest partition will be the training subset, which will contain all training data the model will see during training. In order to enhance the model's performance, it compares its predictions on the training set with the corresponding outcomes in the test set. Through a series of training iterations the model learns from its correct and incorrect predictions to improve its accuracy. The validation subset is a subset of the training data that is unseen by the model during training and is used to simply evaluate how the model performs on unseen data. Cross-validation is a common method to find true model performance using different sections of the data as unseen prediction data [131] [132].

Chapter 3

Related works

This chapter briefly discuss articles and research papers on topics that are relevant to this master thesis project.

3.1 Generative Adversarial Network image regeneration

S. Palazzo et al. published a study [133] in 2017 where they were exploring the possibilities in training a machine learning model based on Generative Adversarial Networks, GAN for short, using respondent's EEG signals. The projects objective was to hopefully achieve better understanding of how the brain operates. The respondents were exposed to several images of different animals and certain kinds of food. While the respondents were exposed to these visual stimuli, recordings through an 128-channel EEG were done. The machine learning model was trained with features extracted from the EEG signals and the specific images that were exposed to the different participants based on Linear Short-Term Memory(LSTM) and Rectified Linear Unit(ReLU) non-linearity. The main purpose of this experiment was to see how well a machine learning model would be capable of regenerating an image based solely on the respondents EEG responses and the associative images. The respondents were exposed to 50 images within 40 different

objectclasses. Notch filter integrated in the hardware was used to eliminate power line noise. Furthermore, a second order Butterworth filter was used to extract the alpha, beta and gamma waves in the EEG recordings within the frequency ranges of 14-70Hz. The GAN based model showed an impressive performance on the topic and was capable of reconstructing images of extraordinary resemblance with only the EEG recordings as testing and validation input. Other images were more distorted by certain artefacts and the model was not able to reconstruct images in a well enough manner to be qualified as 'good' results compared to the ones with high original resemblance.

3.2 Error-related potential training for improved machine decision making

In another study [134], Joo Hwan Shin et al. were exploring the possibilities of extracting features from wearable EEG electronics to reinforce a machine's decision making in executing physical actions based on human physio electrical signals. By utilising the concept of Brain-AI Closed-Loop System(BACLoS) with wireless single-channel wireless EEG attached with tattoo-like dry electrodes, they were able to create an IoT-system of the respondent's responses and an autonomous car in real time. This specific study used Error-Related Potential(ErrP) classified through a deep learning algorithm with features extracted from the respondent's alpha waves using a FIR band pass filter(8-12Hz) and the Morlet wavelet transform(5-15Hz) to train the model. If relatively normal EEG responses were detected the model was trained such that the action done was set as a "Yes", meanwhile if ErrP was detected, then it was interpreted as a "No" making the model understand which action was part of good decision-making or not. Several methods for model training was done using deep neural networks(DNN), Long Short-Term Memory(LSTM) of recurrent neural networks, logistic regression(LR), Linear Discriminant Analysis(LDA), Random Forest(RF) and Support Vector Machine(SVM). The model was trained and tested using a remote controlled car driving on a marked out path on paper. Additionally, the model was tested with maze solving games to optimise the time and path taken to solve the puzzle. The results showed that the outcome for the machine's decision-making were improved using the BACLoS concept using human decision-making to reinforce the model.

3.3 Emotion recognition based on smooth spectral features

Tugce Balli et al. in [135], used EEG to extract spectral features to classify three emotional responses: arousal, liking and valence to detect emotional states of the respondents. The respondents were exposed to eight video segments from Turkish TV-shows lasting between 2 and 4 minutes whereas 4 of the video segments had emotional content that were positive and the 4 others had negative content which were shown in random order to every respondent. The EEG data was band pass filtered during recording using a band pass filter within the range of 0.1-250Hz and ocular artefacts were removed using ICA. A Hamming window was used to eliminate overlapping segments of spectral data and FFT was used to transform the data into spectral domain. The frequency ranges of interest for this particular study was between 0.5-30Hz. Sequential Floating Forward Search with Linear Discriminant Analysis(SFFS-LDA) algorithm was used for feature selection for the model training. With the modified SFFS-LDA algorithm the model achieved an accuracy between approximately 67-72% across all three different emotional dimensions.

3.4 Emotion Recognition using multi-level classification model

In article [136], by Yi-Hung Liu et al. published their research using Imbalanced Quasiconformal Kernel Support Vector Machine(IQK-SVM) and Kernel Fisher's Emotion Pattern(KFEP) for emotion recognition using single-trial EEG. The research intention was to be able to develop a multi-level classification model to classify emotional response divided into emotional arousal and valence. Respondents were exposed to images from International Affective Picture System(IAPS) to provoke emotional responses during the EEG recording. The relevant frequency ranges for the proposed methods were chosen to be between 4-45Hz, which includes the wavebands between theta and gamma. Feature extraction was done through the KFEP algorithm which combines Principal Components analysis for dimension reduction and Kernel Fisher Discriminant Analysis to maximise class separability. Results from the KFEP were sent to the next layer of the proposed

model through the IQK-SVM for further classification. Their report concluded with the model performing with classification accuracies of 84.79% for emotional arousal and 82.68% for valence.

Chapter 4

Data collection and equipment specifications

This chapter elaborates about the experimental setup for data collection, which hardware and software that were used and what stimuli was shown to the respondents and how it was represented during this study.

4.1 Stimuli source

Visual stimuli used in this project was fetched from the Open Affective Standardized Image Set, OASIS for short, consisting of images divided in four categories; animals, objects, people and scenery. The images' valence and emotional arousal was validated through a variety of people hired through Amazon's Mechanical Turk. OASIS was chosen as source of stimuli over the International Affective Picture System(IAPS) due to constraints based on the IAPS images' copyrights [137].

4.2 Hardware and software specifications

During this project the Enobio 32-channel wireless electrophysiology sensor system [138] was used to perform the recording of the respondents for collection of data. iMotions' lab streaming layer platform [139] was used to synchronise the visual stimuli and the EEG recording. Electroconductive gel was used to make contact between the EEG electrodes and the respondents' scalps. All the equipment and a computer with necessary recording software were located at the cognitive lab at the University of Stavanger, therefore all experiments and recordings were done at this location.

(a) Gel type 1 **(b)** Gel type 2

Figure 4.1: Electroconductive gel used to improve connection between EEG electrodes and respondents' scalps.

4.3 Stimuli representation

For the two pilot runs of data collection, all 900 images from OASIS were used, showing them in the proposed original order [137]. Every image was displayed on screen while recording EEG data for 5 seconds in a video sequence made using PowerPoint slides. Due to respondents' feedback which could indicate emotional habituation, the stimuli was reduced. Habituation is when physiological responses decrease due to repeated stimuli [140]. The

stimuli was reduced to 120 images, whereas some images that had exceptional high or exceptional low valence appeared randomly among images that had a relatively 'normal' valence. This shortened the total stimuli exposure time from 75 to 10 minutes. An overview of which image that was expected to provoke which specific emotional feeling in the reduced stimuli based on Plutchik's wheel of emotions, as illustrated in Figure 2.4 and described in Subsection 2.1.4, is attached in Appendix A.

4.4 Experimental setup

Figure 4.2 shows the experimental setup of the data recording. The stimuli was displayed on the same computer with the IMotions software that the NIC2 was connected to over WiFi.

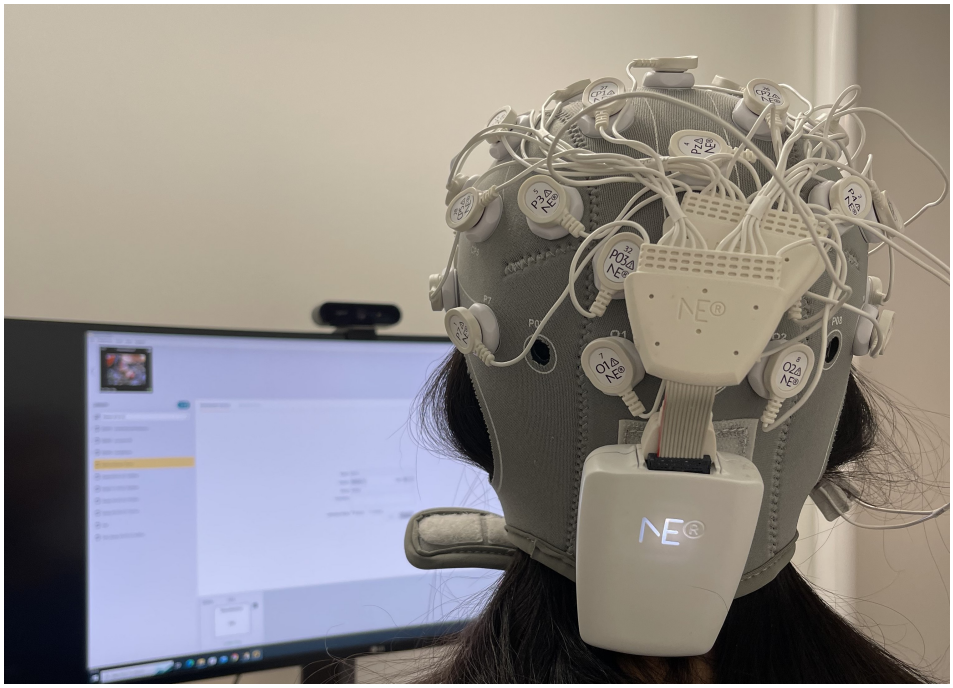


Figure 4.2: Respondent wearing the Enobio 32-channel EEG device, ready for EEG recording.

4.4 Experimental setup

For the experimental setup, the standard 32-channel EEG protocol proposed in the NIC2-software that the Enobio 2 was running on was used. An overview of this protocol's channel placement on specific locations on the scalp is shown in the table in Appendix B.

IMotions' software displayed a pop-up that asked the respondents for their consent to collect their data during the stimuli exposure. This had to be ticked-off by the respondent themselves for the recording to start. The respondents was left alone in the experiment room for the entire duration of the recording and asked to make themselves comfortable for the time being and not to make too many abrupt movements to avoid unwanted artefacts in the EEG data. All data was anonymised and labelled as $anon\_1, anon\_2, ...anon\_N$ in the data set. Table 4.1 shows the number of expected epochs divided into the expected emotional responses during recording, total amount of images and the total running time.

	anon_1,2	anon_3,4,5
Boredom	341	46
Joy	159	26
Sadness	72	12
Disgust	27	9
Arousal	63	12
Admiration	109	6
Fear	107	5
Anger	22	4
Total images	900	120
Total time	1h 15min	10min

Table 4.1: Overview of number of epochs recorded from each respondents divided into expected emotional responses.

Chapter 5

EEG signal processing methods and results

This chapter and Chapter 6 in this thesis report will show the work that has been done during the project with the data gathered as described in the previous Chapter 4. This chapter will focus on illustrating the results of the signal processing of the raw data and explain why the specific methods for processing are used. To not involve an extansive amount of figures, only anon_3's data will be shown in this chapter to give a brief overview of the process done to all respondent's data. All source code can be accessed through the open GitHub Repository.

Proposed pipeline

Based on the theory discussed in Chapter 2, the signal processing steps were done in order according to the following pipeline:

1. **Raw data inspection**
   Here the raw data is inspected to get an overview of how the data looks like before preprocessing.
2. **Baseline correction**

Here the data is baselined to remove any DC offsets to achieve overall zero mean.

3. **Bad channel rejection**
   In this step, the bad channels are automatically identified and removed from the data set.

4. **Bad channel interpolation**
   Here the previous eliminated bad channels are interpolated through neighbouring channels to keep data sets across respondents within the same dimensions.

5. **Notch filtering**
   Here the signal data is notch filtered to attenuate any noise caused by the power line.

6. **Bandpass filtering**
   This step is done to attenuate any signals outside the frequency regions of interest, in addition to eliminate any low frequency drifts.

7. **Group averaging**
   Averaging across all channel data is performed to create an unambiguous interpretation of the signal data to use in an ERP analysis.

8. **Signal epoching**
   Dividing the averaged data into epochs to match time of stimuli exposed to the respondents.

9. **Wavelet transformation**
   Transforming the data using Morlet wavelets to represent the data in a scalogram for a time-frequency analysis and preparing the data for feature extraction.

5.1 Raw data inspection

Before preprocessing the data, the data was loaded into MatLab to get a full view of the exported .CSV-files that were fetched from IMotions. This was to get a simple overview of how the software exported the data. Some of the signals went over the expected length of 300.000 samples with 500 sampling rate, resulting in 600 seconds or 10 minutes. The data points that were taken

in consideration for this project was the samples ranging from the timestamp beginning of the stimuli exposure, named as "StartMedia" up until exact 300.000 samples were collected for all 32 channels. The correct matrices were then loaded into a Python script using Pandas data frames [141]. Figure 5.1 shows the raw data from respondent anon_3 and two representations of the data's power spectral densities based on traditional FFT and another using Welch's method as described in Section 2.3 using the *scipy.signal* python package [142].

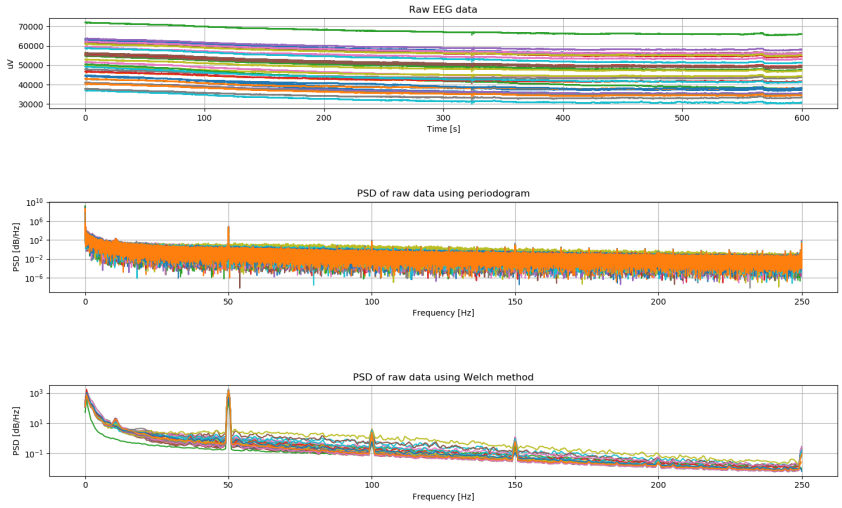
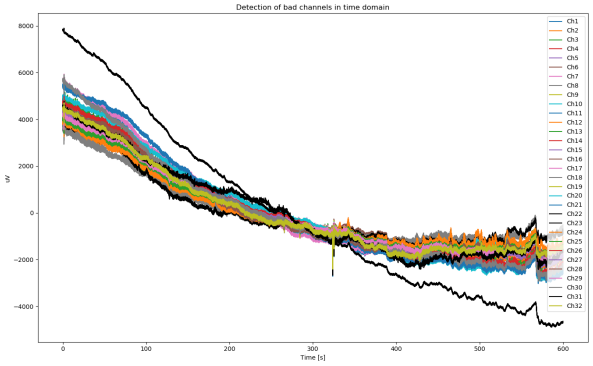


Figure 5.1: Raw data from anon_3 shown in time domain and frequency domain, standard periodogram is plotted in the middle figure and the estimated power spectral density using Welch's method at the bottom. The power line frequency at 50Hz is prominent, which can be observed in both frequency domains.

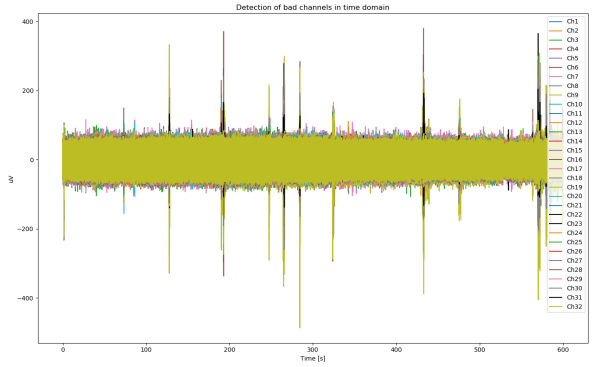
By inspecting Figure 5.1 it is evident that the power line noise at 50Hz is clearly prominent in the raw data. The Welch's method gives a far easier interpretation of the signals than the periodogram as the curves are a lot more smooth and is not corrupted by PFE. Thus the representation of the power spectral density that is used further into the signal processing and analysis is with the use of Welch's method.

5.2 Baseline correction and bad channel rejection

As described in Chapter 2, there are two common ways to perform baseline correction of EEG data. Figure 5.2 shows the result of the two different methods, mean average subtraction and Zhang-fit correction, for baseline correction on respondent anon_3's data. Automatic bad channel detection has been performed for both baseline correction methods, and it can be observed that the baseline correction does not affect the bad channel detection as the same channels are marked as bad for both scenarios. The detected bad channels in this figure are marked in black to easier visualise them.



(a) Mean average subtraction baseline correction

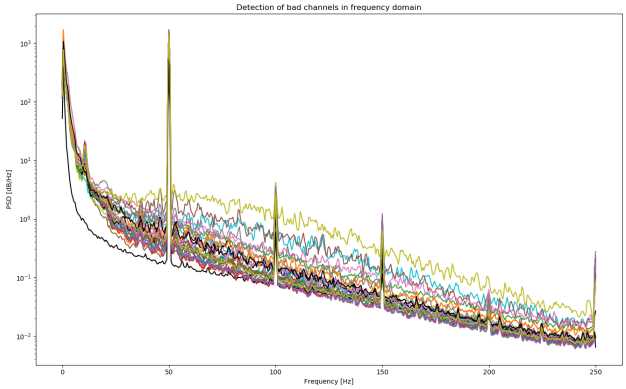


(b) Zhang fit baseline correction

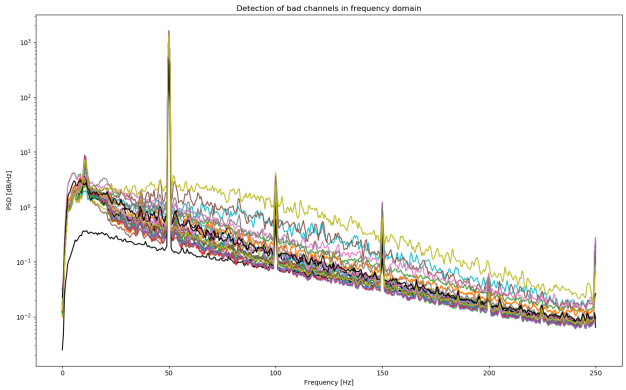
Figure 5.2: Two methods for baseline correction of EEG data from anon_3 in time domain.

Comparison between baseline corrected signal data and the raw data shown at the top in Figure 5.1 can then be done. By visually inspecting the data, the first baseline correction method is the easiest one to use to pick out the bad channels manually, whereas Channel 23 greatly differentiate from the rest of the signals in the data set. In addition, other channels were marked as bad through automatic detection. It is not always so easy to visually inspect channel outliers, so printing of the marked channels was also implemented in the code.

Figure 5.3 shows how the different baseline corrections affect the frequency representation of the signals.



(a) Mean average subtraction baseline correction



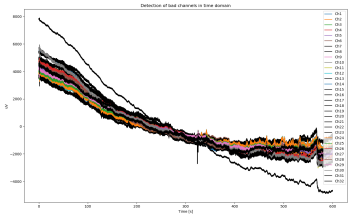
(b) Zhang fit baseline correction

Figure 5.3: Two methods for baseline correction of EEG data from anon_3 in frequency domain with bad channel thresholds = 2.0. The first method preserves all frequency components, meanwhile the Zhang fit method attenuates the low frequencies.

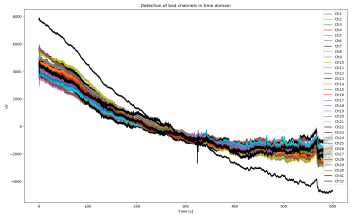
The greatest difference between the two methods through frequency domain representation of the data is the contents of low frequencies. The mean av-

erage subtracted baseline method results in great presence of low frequency contents, meanwhile being attenuated for the Zhang fit method. This is commonly known as low frequency drifts which will be explained in further detail in Subsection 5.4.2. This is also easy to observe by looking at the Figure 5.2 where the data is centered around 0, but drifts between 8000 microvolts to -4000 microvolts. For this project, the simple mean average subtraction method was used due to uncontrolled attenuation of low frequencies occuring with use of the Zhang fit algorithm, which is observable in the signals' amplitudes when band pass filtered as shown in Subsection 5.4.2.

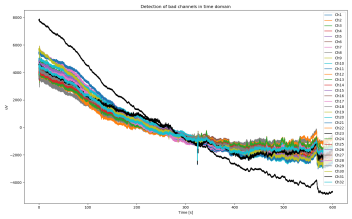
The bad channel detection can be divided into two separate methods, whereas one is detecting bad channels looking at the data in time domain and the other in frequency domain. Both functions are marking the bad channels automatically by calculating the data's standard deviation and comparing each signal to the standard deviation's interquartile range. If the signal is detected as an outlier based on a set threshold, which is multiplied with the standard deviation, the channel is marked as a bad channel. Figure 5.4 shows the results of using the automatic bad channel detection with different thresholds in time domain.



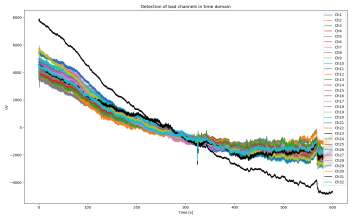
(a) Threshold = 0.5



(b) Threshold = 1.0



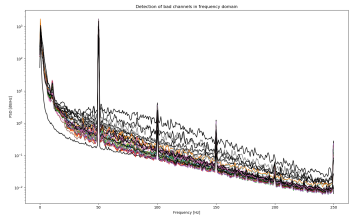
(c) Threshold = 2.5



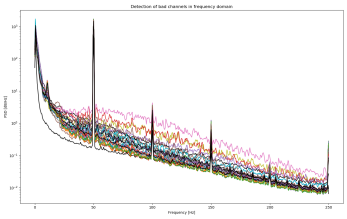
(d) Threshold = 4

Figure 5.4: Bad channel detection in time domain with different threshold values for data from anon_3.

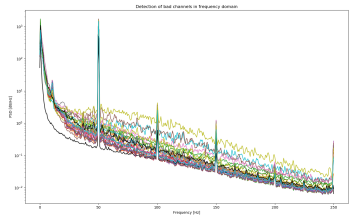
Figure 5.5 shows the results of using the automatic bad channel detection with different thresholds in the frequency domain.



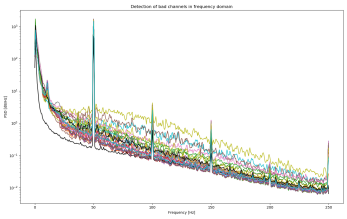
(a) Threshold = 0.5



(b) Threshold = 1.0



(c) Threshold = 2.5



(d) Threshold = 4

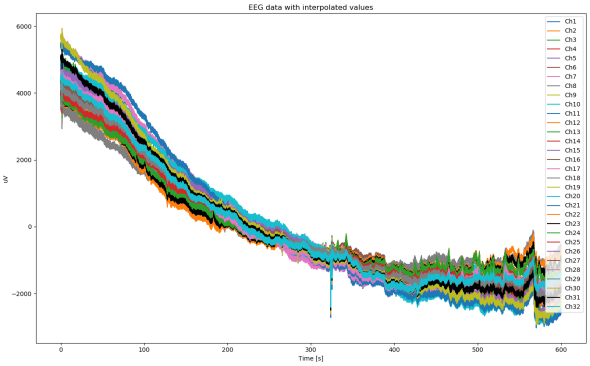
Figure 5.5: Bad channel detection in time domain with different threshold values from anon_3.

Figure 5.4 and Figure 5.5 illustrates that if a lower threshold is chosen, the stricter the bad channel detection functions are. Because preservation of the original signals in the data set is highly preferred for this study, a threshold of 4 was chosen for this specific respondent's response. This specific threshold was chosen because it results in marking only the most outlying channels as bad. After being marked, the channels were set to 0 non-zero arrays, clearing the bad channel data.

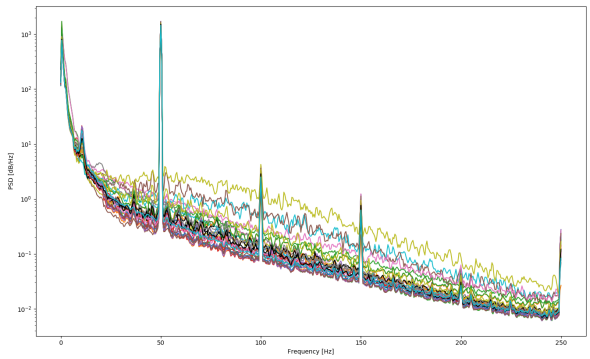
5.3 Channel interpolation

After the channels are marked and cleared, the bad channels could be interpolated to maintain the data quantity. Figure 5.6 shows the linear inter-

polation, as described in 2.3.3, done to the bad channels in anon_3's data set.



(a) Linear interpolation of bad channel data represented in time domain



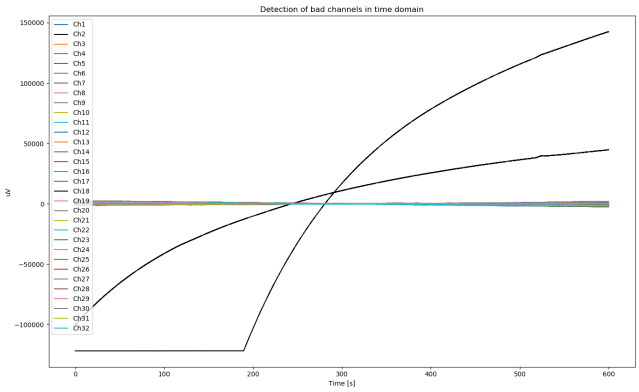
(b) Linear interpolation of bad channel data represented in frequency domain

Figure 5.6: Linear interpolation of bad channel data, channel 23 and 31 for anon_3.

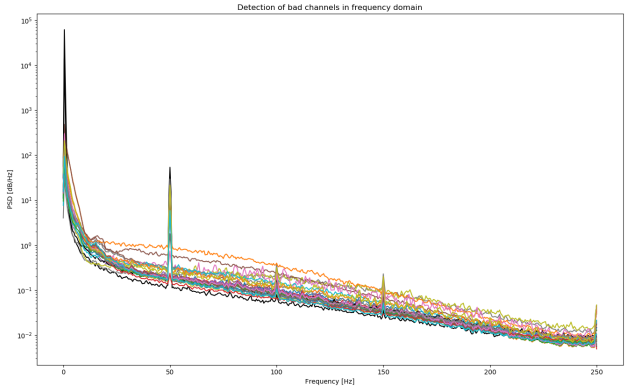
Observations from Figure 5.6 shows that the interpolated channel data are improved by inspecting the marked signals in both time -and frequency domain. The channels previously marked as bad can now be seen as new interpolated signals in black in the figure.

5.3 Channel interpolation

Bad channel rejection and interpolation may be more significantly important in some scenarios than others. Why bad channels should be detected and rejected from the data set is shown in Figure 5.7 that represent the signal data from respondent anon_1.



(a) Linear interpolation of bad channel data represented in time domain



(b) Linear interpolation of bad channel data represented in frequency domain

Figure 5.7: Bad channel detection used in for the data recorded from respondent anon_1.

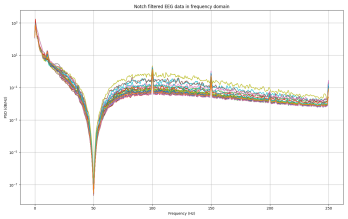
Group averaging over the data in Figure 5.7 for an ERP analysis would result in a signal dramatically distorted by the marked bad channels. Therefore, bad channels should be removed from the data set before continuing to process it.

5.4 Filtering

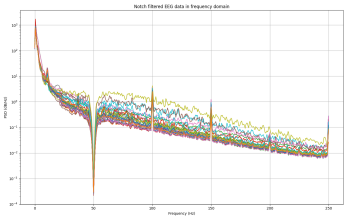
For all filtering methods, the scipy.signal's forward-backward filtering method [143] was used to eliminate any phase shifts or time delays that otherwise would occur when performing digital signal filtering in addition to amplify the wanted underlying EEG signals [144] [145].

5.4.1 Notch filtering

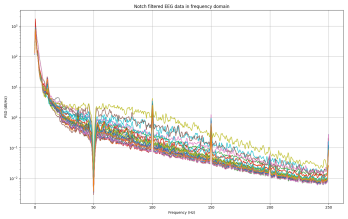
As described in Chapter 2, a notch filter is a band stop filter with a bandwidth defined by the filter's Q-factor. Figure 5.8 shows notch filtering of the data with different Q-factors, specified in sub figures' descriptions, for the data set from anon_3's EEG recording. The Python package *scipy.signal*'s *iirnotch* filter was used to create a second-order IIR notch digital filter [146].



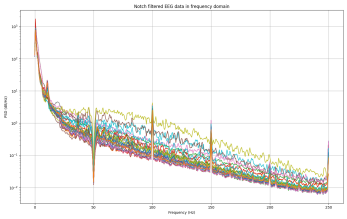
(a) Q-factor = 1



(b) Q-factor = 10



(c) Q-factor = 20



(d) Q-factor = 30

Figure 5.8: Frequency responses after notch filtering with different Q-factors.

For the rest of the signal processing throughout this study, a Q-factor of 30 was used to get a notch filter with a stop-band as narrow as possible to only clearly attenuate the power line frequency. If higher Q-factor is chosen, less of the surrounding frequencies would be attenuated, but the unwanted artefact will not be dampened enough to have the wanted impact in later stages. This is described in further detail later in Subsection 5.4.3.

5.4.2 Band pass filtering

To be able to extract the wanted components of the signal, the data was band pass filtered between the frequencies of 4-45Hz using a 4th order Butterworth IIR filter, inspired by the MNE-package's documentation [147]. This is because the delta waves are not relevant for this project and at the same time this will result in filtering out the eye blinking artefacts and potential body movements that occurred during the recording as described in Chapter 2. Additionally, the delta waves that is in the range of 4Hz and lower is not relevant in this project, because all respondents were awake during the recording and the delta waves corresponds to a deep sleeping state [148]. Figure 5.9 shows band pass filtering of respondent anon_3's data.

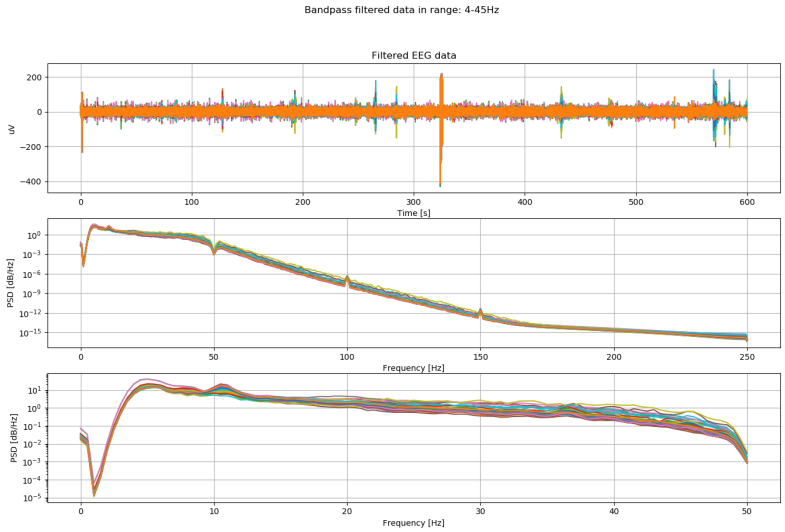


Figure 5.9: Band pass filtered data on the top and their frequency responses with full range 0-250Hz in the middle and zoomed in range 0-50Hz at the bottom.

The reason why the data need to be band pass filtered and not only low pass filtered, to extract the underlying EEG signals, is due to low frequency drifting. High pass filtering is commonly implemented in addition to low pass filtering to get rid of the drifts [149] [150]. Therefore, the combined version in the form of a band pass filter is used. Figure 5.10 shows anon_3's

low pass filtered data. By observation, it is simple to determine that the low frequency drifting persists and that high pass filtering is essential.

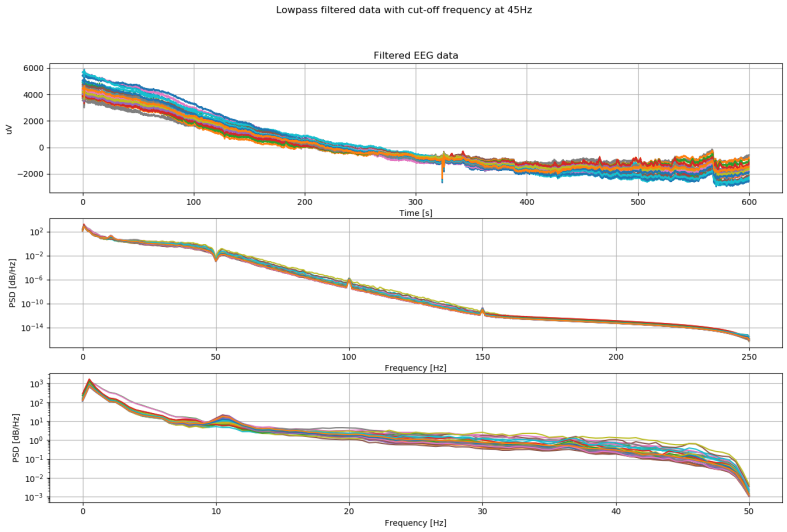
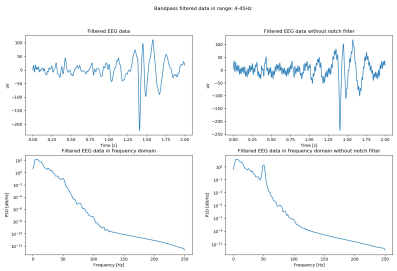


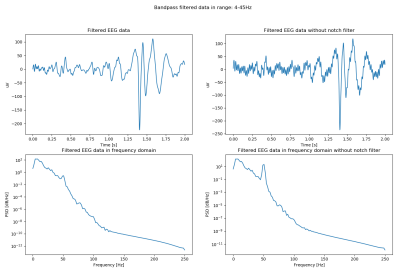
Figure 5.10: Low pass filtering results of data collected from anon_3's EEG responses.

5.4.3 Why notch filtering is necessary

Figure 5.11 shows why conventional band pass filtering is not enough to achieve sufficient filtering results. Figure 5.11a shows a band pass filtered signal from a single channel from anon_3 with notch filtering on the left and without on the right side of the figure. It is plainly evident that the notch filter has a significant impact to the final result, even though the Butterworth band pass filter is defined outside of the power line noise frequency. This can be observed in both the time domain and frequency domain. At 50Hz in the frequency domain of the band pass filtered signal without notch filtering shows a peak of how much the signal hold of that specific frequency and is still too prominent.



(a) Q-factor = 30



(b) Q-factor = 50

Figure 5.11: Figure showing why notch filtering should be used with band pass filtering and why Q-factor 30 gives slightly better performance than 50.

Moreover, the notch filtered signal with a Q-factor of 50 in the lower left corner of Figure 5.11b, demonstrates that the attenuation of the unwanted artefacts originating from the power line noise is not sufficiently dampened

compared to a Q-factor of 30.

5.5 Group averaging and signal epoching

After filtering the data in such manner that only the underlying EEG values of interest is present in the signals, then the group averaging of channels could be done to use the data for an ERP analysis. This was done in the code by simply calculating the average mean of all channels. Meaning that all the channel data were added together and divided by the total number of channels present. Figure 5.12 shows the result of the channel group averaging and epoch boundaries.

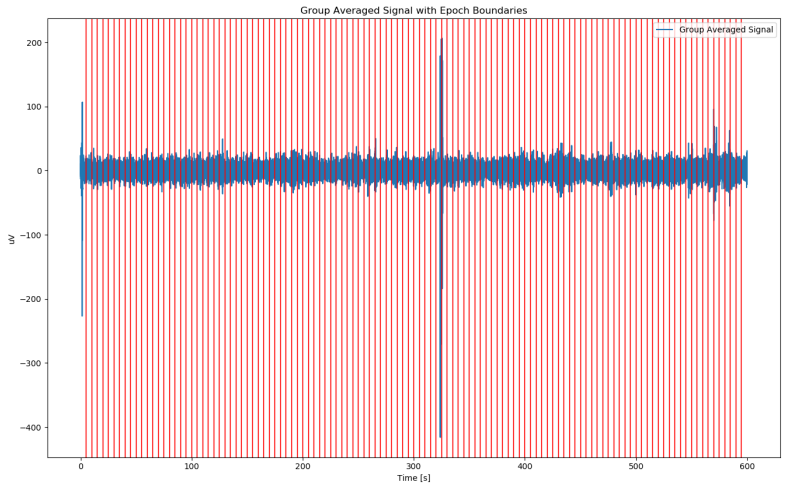


Figure 5.12: Group averaged signal data of respondent anon_3 with epoch boundaries.

Furthermore, the data was segmented into epochs based on the time exposures of the different stimuli. Because there was 120 images, that were used for most of the respondents, it resulted in 120 epochs each data set. The epochs were then plotted to do manual identification and rejection of bad epochs. Attempts of using the FastICA [151] for preserving the epoch data

was done, but without success. Therefore bad epochs were simply removed from the training and test sets. Fortunately, there was only approximately 2-3 bad epochs each data set, resulting in only loss of a small fraction of the total data set. Figure 5.13 shows plots of 5 EEG epochs, whereas epoch 1 clearly can be marked as a bad epoch with its abrupt high oscillations.

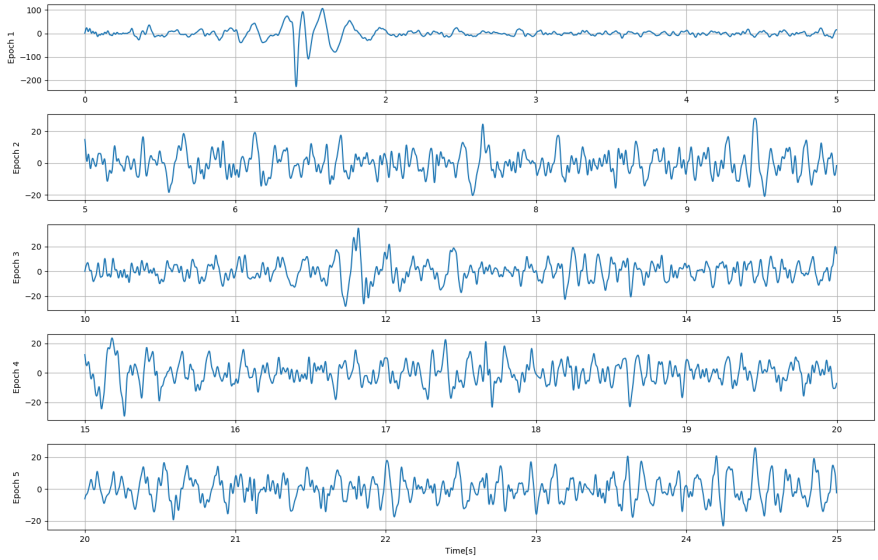
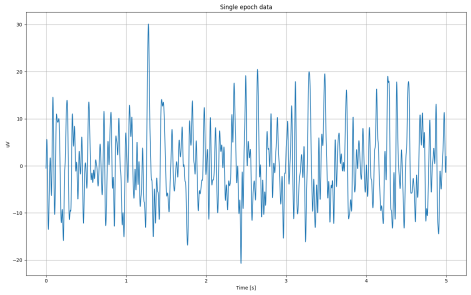


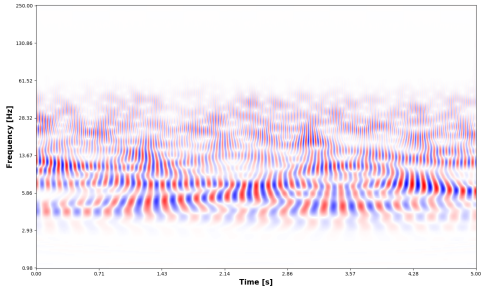
Figure 5.13: Plot of the first 5 epochs in anon_3's group averaged signal data.

5.6 Wavelet transformation

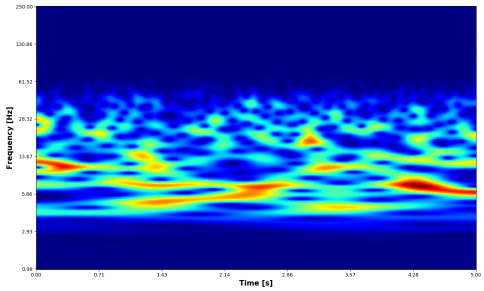
To perform the time-frequency analysis of the data, a python package named *ssqueezypy* [152] is used to easily perform a wavelet transformation and display the data in a fitted scalogram. Figure 5.14 shows a typical scalogram representation of a result from performing wavelet transformation using the complex Morlet wavelet as defined in Subsection 2.4.3.



(a) Epoch in time domain.



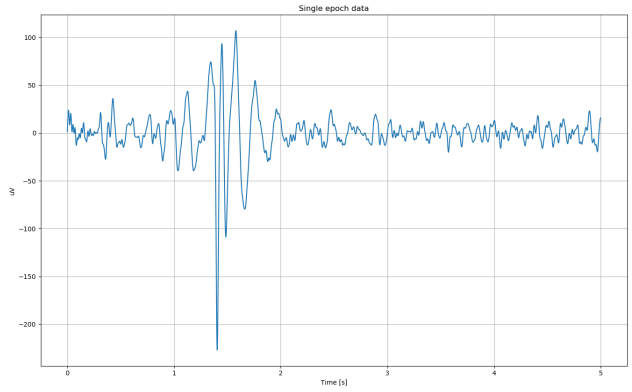
(b) Scalogram of epoch using the transformed signal's complex values.



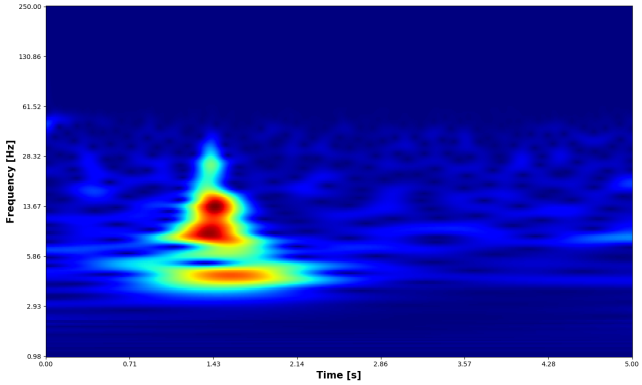
(c) Scalogram of epoch using the transformed signal's absolute values.

Figure 5.14: Scalogram of an ordinary epoch using the complex Morlet wavelet for transformation.

For better understanding the representation of a signal in a scalogram, wavelet transformation was done to the first epoch of respondent anon_3's data, which was marked as a bad epoch. In Figure 5.15 is it fairly simple to observe how the scalogram works.



(a) Epoch data



(b) Scalogram of epoch using the wavelet transformation's absolute values

Figure 5.15: Scalogram for time-frequency analysis of bad epoch.

The scalogram displays when in time the different frequencies occur in the signal and its magnitude. The magnitude is displayed through a range of

colors going from dark blue to dark red, whereas the dark red indicates a high magnitude and dark blue indicates a low magnitude. At approximately around 1.4 seconds for this specific epoch it can easily be observed that the signal has a high content of frequencies at roughly 14, 8 and 3 Hertz. The resulting transformed signal values can then be used for a machine learning model's training or test set in a classification problem.

Chapter 6

Model training and classification results

This chapter will show the results of training and evaluating machine learning models using different methods for model training and feature extraction. Other methods were also tested out, but due to not enhancing model performance they are not included in this chapter. Instead see Appendix C for external results.

Proposed pipeline

1. **Feature extraction**
   Here the processed and transformed data coefficients(features) are set into lists in dictionaries with expected emotional responses as labels and saved as a serialised files.

2. **Feature selection**
   If necessary, feature selection is performed on feature data before used in model training.

3. **Model selection**
   Here the data is used for training and testing with several simple machine learning algorithms among the most commonly used methods

within multi-label classification using the proposed default values to determine which method to go on with for further training based on the highest accuracy.

4. **Model training**
   Here the selected model will go through multiple training iterations to see how the model's performance evolve.

5. **Model evaluation**
   The model is evaluated based on it's performance on predicting unseen data through cross-validation and a complete data set consisting of unseen data using classification reports and confusion matrices.

6. **Parameter adjustments**
   If necessary, model's parameters are adjusted for the purpose of enhancing model performance.

7. **Model mass training**
   The model is trained multiple times with data across different respondents.

8. **Model evaluation**
   Final model is tested on data set consisting of only unseen data and evaluated from its performance through a confusion matrix and classification report. The final evaluation will indicate how the model will perform on predicting other similar signals going further based on the data currently available.

6.1 Feature extraction

The wavelet transformed epoch signals were extracted and loaded into .npy-files which would be easy to extract later for model training and testing purposes. Signal features were stored as arrays in a dictionary for the emotional responses. For testing purposes, three different kinds of dictionaries were tested out with more or less nuanced emotional responses. The first dictionary contained the 8 different emotional responses; joy, admiration, arousal, disgust, sadness, fear, anger, boredom/uninteresting. These responses are based on two of the respondents personal opinions and on the results from the study in [137].

The second dictionary contained three responses which were less nuanced, these were divided in to; positive (containing features from: joy, admiration and arousal), *negative* (containing features from: disgust, sadness, fear and anger), and *neutral* (boredom/uninteresting). Even though there are no real value defined for neutral feelings, as discussed in Subsection 2.1.4, this was still distinguished for this project as there were more boredom or uninteresting images than the positive or negative images combined.

The last dictionary was based solely on valence, as discussed in Subsection 2.1.4, distinguishing between positive and negative emotional responses. Because the models that were used in this project were not supporting complex values, the absolute value of the complex Morlet wavelet coefficients were used.

The *ssqueezypy* package also has a module for a synchrosqueezed version of the complex wavelet, resulting in resolution enhancement [115].

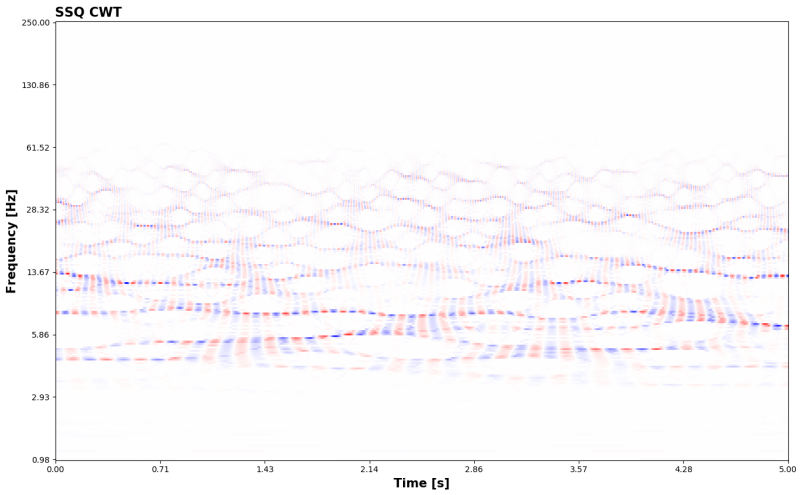


Figure 6.1: Synchrosqueezed complex Morlet wavelet transformed epoch data.

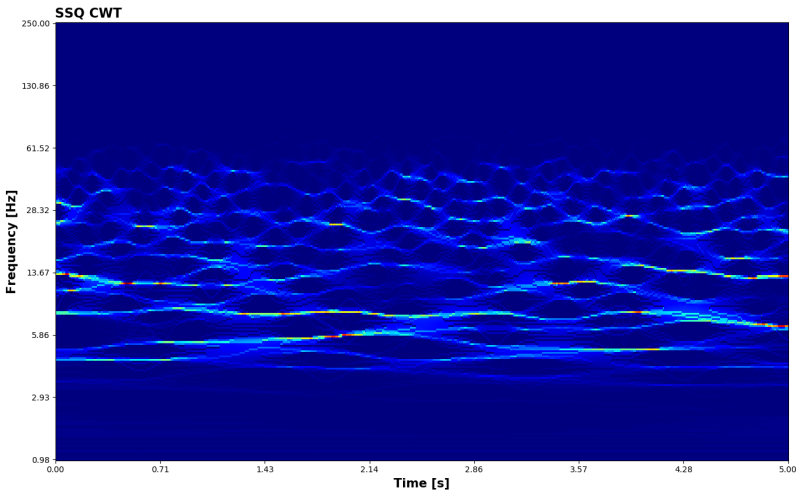


Figure 6.2: Absolute value representation of synchrosqueezed complex Morlet wavelet transformed epoch data.

6.2 Model development

At first, three different machine learning algorithms(kNN, Naïve Bayes and random forest) were tested with two respondent's data for the features that were stored in a feature file using the second dictionary with the three different classes. This was to investigate which algorithm gave the best accuracy results on the first training iteration with default values and decide to go on with further training on the best performing model. *Sklearn*'s [153] modules were used to define, train, test and validate the different models throughout this chapter. Features derived directly from the absolute value of the complex Morlet wavelet transformation's coefficients were used for training during model selection. The classification reports for three different algorithms with default values are listed in the tables below.

	precision	recall	f1-score	support
Negative	0.33	0.15	0.21	13
Neutral	0.48	0.59	0.53	17
Positive	0.30	0.35	0.32	17
Accuracy			0.38	47
macro avg	0.37	0.37	0.35	47
weighted avg	0.37	0.38	0.37	47

Table 6.1: Classification report from model based on k-Nearest Neighbour algorithm with default parameter values.

	precision	recall	f1-score	support
Negative	0.00	0.00	0.00	13
Neutral	0.33	0.12	0.17	17
Positive	0.25	0.53	0.34	17
Accuracy			0.23	47
macro avg	0.19	0.22	0.17	47
weighted avg	0.21	0.23	0.19	47

Table 6.2: Classification report from model based on Naïve Bayes algorithm with default parameter values.

	precision	recall	f1-score	support
Negative	1.00	0.15	0.27	13
Neutral	0.48	0.65	0.55	17
Positive	0.41	0.53	0.46	17
Accuracy			0.47	47
macro avg	0.63	0.44	0.43	47
weighted avg	0.60	0.47	0.44	47

Table 6.3: Classification report from model based on Random Forest algorithm with default parameter values.

By evaluating each classification report, specifically inspecting the accuracy for the different machine learning algorithms, the random forest machine learning method's results in Table 6.3 had the best outcome of 0.47 accuracy. Therefore, this was the model used for training on the features for eight different classes.

Unfortunately, after running multiple iterations of model training, it seemed like the model was oscillating in model performance and not converging linearly to a final accuracy value. This was illustrated by plotting the model accuracy over training iterations and can be observed in Figure 6.3.

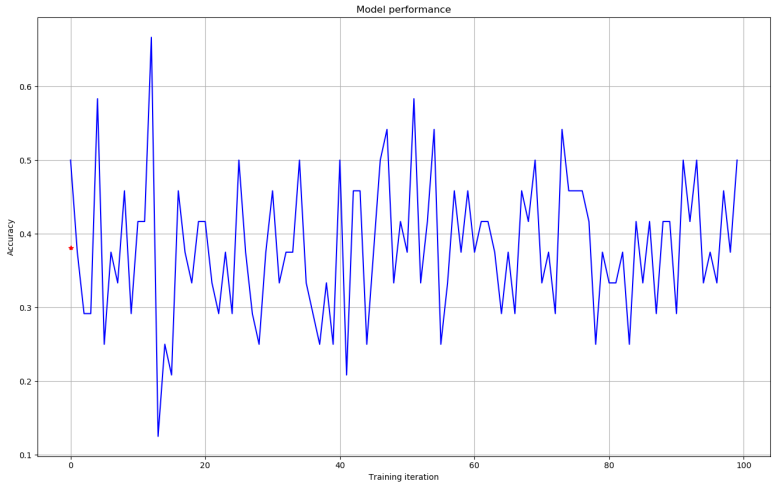


Figure 6.3: Development of model performance through multiple training iterations.

That the model's performance was oscillating could indicate that the amount of properties learned from the features might be too many for the model to be able to distinguish the different features from one another to correctly classify them, which might lead to model overfitting for multiple training iterations.

Therefore, a feature selector based on feature variances of 80% was implemented to perform dimensionality reduction to the data. This is because the EEG features were carrying a lot of information which can be too advanced of a task for a machine learning model to perform. Considering the model might be overfitted , the number of estimators in the random forest model were reduced from the default of 100 estimators to 9. This did not change any of the behaviour of the model's performance and the results were still oscillating. This can be observed in Figure 6.4.

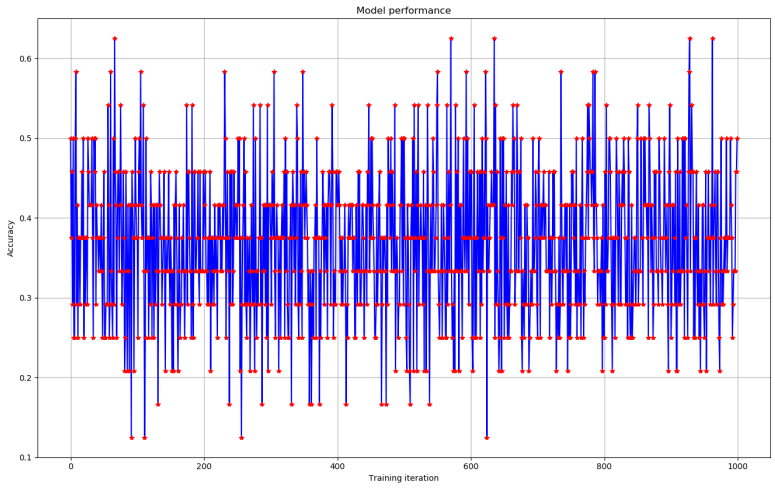


Figure 6.4: Model performance through multiple training iterations with use of a feature selector.

Furthermore, a model based on random forest algorithm was used to train on synschrosqueezed CWT features to achieve higher resolution in the feature data. To minimise the number of features, but at the same time keeping the most critical properties, PCA was performed to shorten down the number of components from each epoch.

6.2 Model development

After training the model for a while on a large data set from one of the respondent results showed the model reached 93% accuracy on predicting on the training data. The confusion matrix can be observed in Figure 6.5 and the classification report is shown in Table 6.4.

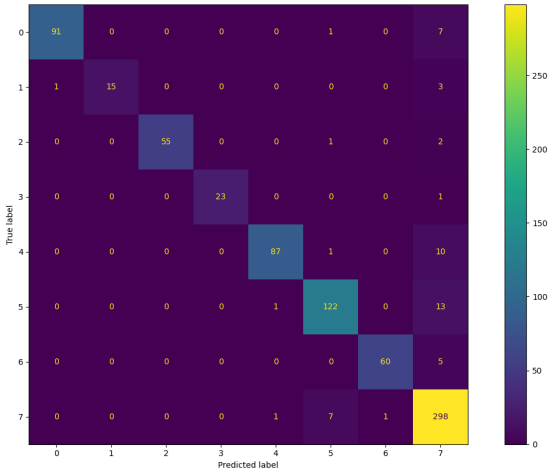


Figure 6.5: Confusion matrix for predicted data on the same trained data set.

	precision	recall	f1-score	support
Admiration	0.99	0.92	0.95	99
Anger	1.00	0.79	0.88	19
Arousal	1.00	0.95	0.97	58
Disgust	1.00	0.96	0.98	24
Fear	0.98	0.89	0.93	98
Joy	0.92	0.90	0.91	136
Sadness	0.98	0.92	0.95	65
Boredom	0.88	0.97	0.92	307
Accuracy			0.93	806
macro avg	0.97	0.91	0.94	806
weighted avg	0.94	0.93	0.93	806

Table 6.4: Classification report from testing on trained model with the training data.

The model was then used to predict on another respondent's unseen data. As the unseen data consisted mostly of features within the 7th class with the label *Boredom*, most of the model's predictions were of that specific class. This would be an indication of a typical overfitted model. The confusion matrix from these results is shown in Figure 6.6 below.

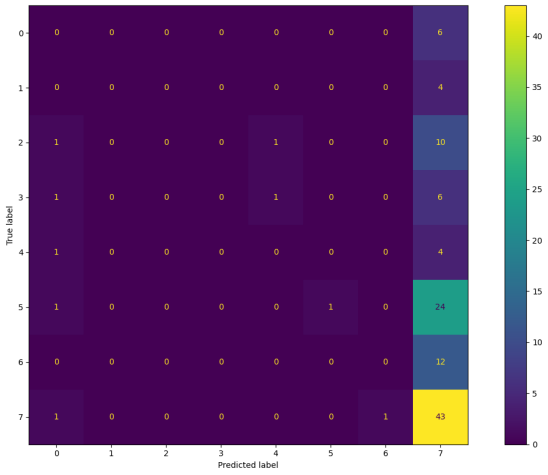


Figure 6.6: Confusion matrix for predicted unseen EEG data from another respondent. The results indicate a favouring of class 7, which is *Boredom*.

As observed from the confusion matrix, the model will predict the 7th class almost all the time. This is also shown in the classification report in Table 6.5.

	precision	recall	f1-score	support
Admiration	0.00	0.00	0.00	6
Anger	0.00	0.00	0.00	4
Arousal	0.00	0.00	0.00	12
Disgust	0.00	0.00	0.00	8
Fear	0.00	0.00	0.00	5
Joy	1.00	0.04	0.07	26
Sadness	0.00	0.00	0.00	12
Boredom	0.39	0.96	0.56	45
Accuracy			0.37	118
macro avg	0.17	0.12	0.08	118
weighted avg	0.37	0.37	0.23	118

Table 6.5: Classification report from testing on trained model with unseen EEG data from another respondent, anon_5.

To increase variation in the training data to avoid model overfitting, the model was trained for a total of three different respondents' data, meanwhile one respondent's data, anon_5, were kept out for solely prediction purposes as before. The model's performance then started to show a trend of more converging accuracy over the training iterations as shown in Figure 6.7, which could confirm less model overfitting.

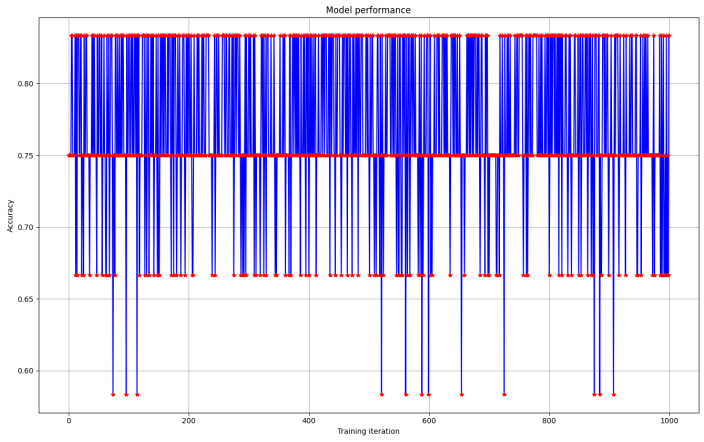


Figure 6.7: Model accuracy over multiple training iterations.

After training the model with data from anon_1, anon_3 and anon_4, then the individual recall and precision increased on the unseen data from anon_5, but the overall accuracy went down. The overall accuracy went from 37% as observed from 6.5 to 32% accuracy on predicting the unseen data of anon_5. The confusion matrix can be observed in Figure 6.8.

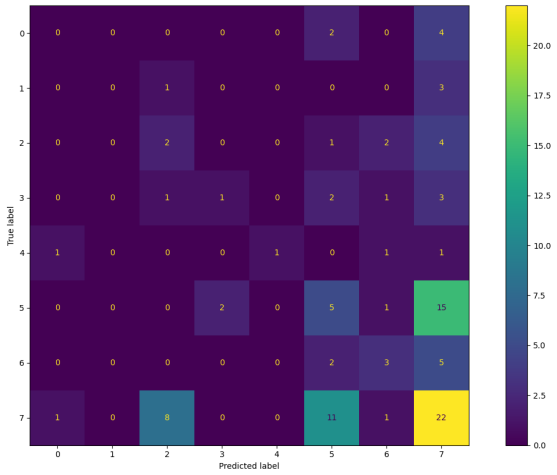


Figure 6.8: Confusion matrix for predicted unseen EEG data from anon_5 with a greater variety of training data from three other respondents.

So, the model became better at predicting a greater variety of emotions in contrast than almost only predicting the emotion *Boredom*, meanwhile the recall value for *Boredom* decreased. A reason why the overall accuracy decreased could be due to the fact that most of the features across all respondents were *Boredom*. Increased recall and precision of the other classes indicated less model overfitting, even though the overall accuracy went down from when the model was overfitted as observed earlier in Figure 6.6.

In attempt of improving model prediction on the unseen data further, the model was trained for 10.000 iterations for each of the training data sets from anon_1, anon_3 and anon_4. Cross-validation was performed on the model using the model's parameter OOB(out of the bag) and gave a score of 0.34%. The model was then used for predicting the unseen data from

anon_5 and it resulted in an accuracy of 0.33% and the results can be observed in Figure 6.9 and Table 6.6.

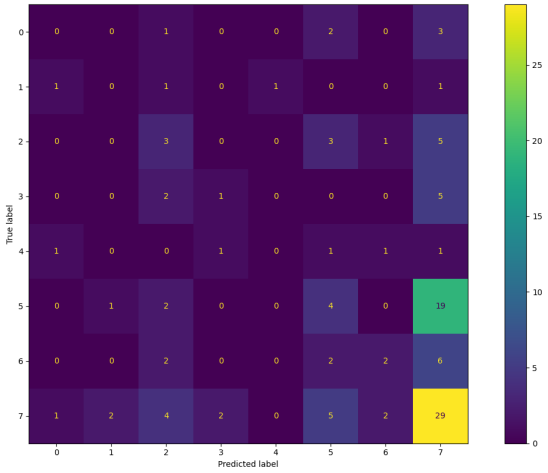


Figure 6.9: Confusion matrix for predicted unseen EEG data after 10.000 training iterations on each training set.

	precision	recall	f1-score	support
Admiration	0.00	0.00	0.00	6
Anger	0.00	0.00	0.00	4
Arousal	0.20	0.25	0.22	12
Disgust	0.25	0.12	0.17	8
Fear	0.00	0.00	0.00	5
Joy	0.24	0.15	0.19	26
Sadness	0.33	0.17	0.22	12
Boredom	0.42	0.64	0.51	45
Accuracy			0.33	118
macro avg	0.18	0.17	0.16	118
weighted avg	0.28	0.33	0.29	118

Table 6.6: Classification report for predicted unseen EEG data after 10.000 training iterations on each training set.

In another attempt of model development, the data features were divided into two classes; positive and negative emotions. To avoid an imbalanced amount of data features, which would result in class favouring, the length of positive and negative emotions were set to be equal, shortening down the class with the most data. This resulted in a confusion matrix and classification report as shown in Figure 6.10 and Table 6.7.

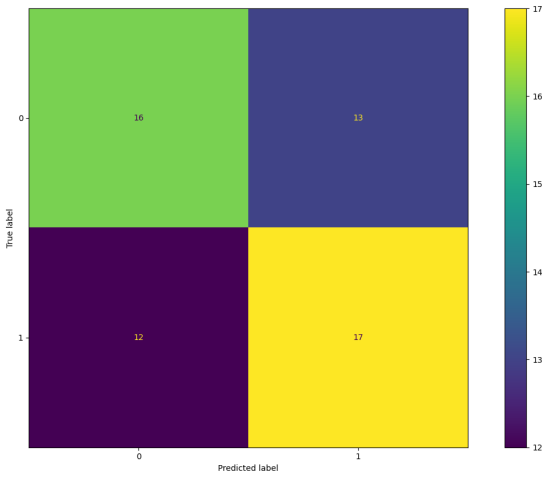


Figure 6.10: Confusion matrix for predicted unseen EEG data after 10.000 training iterations on each training set.

	precision	recall	f1-score	support
Negative	0.57	0.55	0.56	29
Positive	0.57	0.59	0.58	29
Accuracy			0.57	58
macro avg	0.57	0.57	0.57	58
weighted avg	0.57	0.57	0.57	58

Table 6.7: Classification report from trained model on unseen data for 2 classes.

In this thesis project, various additional features, machine learning algorithms, and feature selection methods were explored to enhance model performance. However, the results did not surpass those already discussed in

this chapter, and thus not elaborated in this project. External results can be observed in Appendix C.

Chapter 7

Discussion and conclusion

This chapter will discuss; the project results based on data illustrated in Chapter 5 and Chapter 6, further improvements that can be done to enhance the results, and what other projects can be built upon this project going further.

7.1 Discussion

7.1.1 Signal processing results

Through use of scientific literature and processing tools available through open source code, the signal processing done in this project, shown in 5 resulted in a simple overview of every step of the typical preprocessing procedures involved in EEG analysis. The benefits of performing and visualising the signal processing in such manner is that it gives the user more control of how the raw data is processed and make it possible for the user to fine-tune the parameters for signal enhancement. The implementation of the code developed during this project has resulted in reaching satisfactory standards of general EEG signal preprocessing on the acquired data.

7.1.2 Signal processing improvements

Further improvements of the signal processing methods can be made for easy fetching the wanted EEG recordings by developing a script that allocated the *StartMedia*-mark in the .csv-files and fetches the data timestamps and channel data through data frames. Doing this would avoid the step of importing the large files into a MatLab workspace to visually inspect the data.

Refinement of the bad channel interpolation methods could be necessary for future data filtering, as the current interpolation method would fail to interpolate new data for bad channels if the nearby channels also are marked as bad or if channel 1 or channel 32 is to be interpolated. The challenge with interpolating channel 1 and 32 is that they only have one neighbouring channel. This can be fixed by making the interpolation method look through all channels and selecting the nearest channels that are non-zero to use for interpolation. In the scenario of not selecting a channel that is non-zero will result in an interpolated signal corrupted by another bad channel.

To make the code run more user friendly, methods for selecting parameter values such as bad channel thresholds, notch filter's Q-factor, band pass cut-off frequencies and bad epoch selection could be improved by setting the parameter variables separate from the rest of the script. Moreover, investigation of implementing ICA can be done to decrease data loss for feature extraction and possibly avoid removing bad epochs.

Active band pass filtering could be tested out to increase the data's SNR-value even further, so in addition to attenuate unwanted frequencies in the signal, the frequencies in the pass band would be amplified [154]. Other type of filters than Butterworth filters could also be tested out to see if they could be better suited for EEG data, such as Chebyshev filters [155].

7.1.3 Classification model results

The final results of the classification model did not turn out to be as initially intended for this project due to lack of data. However, it gave a great example on how model overfitting impact the model's capability on estimating

correct predictions. Additionally, the results shown in Chapter 6 show how different feature extraction methods may be applied to improve the data quality and enhance the model's interpretation and learning. Model's predictions of unseen data implies that more varied training data is necessary to make more accurate predictions in the future. With training data from only three respondent and achieving an accuracy just over 30% in distinguishing eight different classes based on emotional responses from unseen data is not yet remarkable. This is also true for predictions of positive and negative valence, where the predictions of the two classes barely surpasses 50% accuracy, which is poor results for a machine learning model.

7.1.4 Classification model improvements

When it comes to the aspect of improving model selection and training, more models can be tested out with setting specific parameters before selecting which machine learning method to work with going into the next steps. This would result in a more extensive workload and the time was limited. However, choosing the correct machine learning method for a specific problems takes time and practice, even if some methods have worked in similar projects, it might not necessarily mean it is the best methods for another.

Other features than the ones exploited in this report could be experimented with to see if other data features are more fit for this classification problem. Extra stratification methods could also be included for possibly improving a model's training and cross-validation data sets. Moreover, involvement of advanced machine learning methods might have improved the model's performance, such as multi class neural networks methods like Softmax [156]. Additional training data for less populated classes would be beneficial to avoid model overfitting in further testing.

7.2 Limitations

During the project there were certain limitations that obstructed project progression and for the sake of formality these limitations are described in

this section.

Time limitations resulted in less exploration of machine learning models and parameters as expected. During signal processing the script would stop visualising the data from anon_2 after filtering, making it impossible to make the right decisions for further preprocessing and feature extraction. Debugging of this was attempted, but no solution to the problem was achieved.

The cognitive lab at UiS was not yet completed and in full operational use at the time for the first data collection, which resulted in lack of guidance in use of EEG equipment and software. Additionally, the data collection itself was a lot more time-consuming than initially expected, even if the stimuli was shortened. If not done right, the data would be completely useless so certain procedures needed to be followed to minimise wear and signal corruption.

7.3 Further work

Moving forward to expand upon the project findings, more EEG data from other respondents should be collected to obtain a greater variety of EEG data utilising the cognitive lab. Images used in this thesis project are uploaded to the GitHub Repository in a PowerPoint presentation in the same order as during data collection and can be reused. The rest of the images can also be fetched from the OASIS files [157]. If enough data is collected, other machine learning methods can be tested to classify the emotional responses in attempt to achieve the optimal outcome, like the more advanced methods described in Chapter 3. Randomising the different test respondents data might be tested to see if it results in an increase of model robustness in predicting unseen data.

For other future work, if the model's performance is substantially increased to enable satisfactory predictions on unseen data, the model can be tested on predicting EEG data that are fetched using other types of stimuli. Also, research on the topic of how the order of different stimuli might impact the model's predictions on different emotional responses can be done. Another interesting research topic would be to investigate if the model will be able

to predict distinction between negative and positive emotional responses to use in lie detection.

7.4 Conclusion

Emotions are challenging to distinguish as people's minds operate differently. Due to the nature of perception, the same stimuli might result in different emotional responses across individuals, with some being more universal than others. Emotions are results of the body's previous sensory experiences interpreted by the brain and humans have different life experiences, thus they will often respond differently [158] [159]. An example of this can be expressed through the image of a polar bear used as stimuli in this thesis project. The different emotional responses might differ and be interpreted as:

- Fear, as the bear is a known predator.
- Joy, because the bear is located in its natural habitat.
- Admiration, as the polar bear is a great majestic animal to look upon.
- Sadness or anger, because to the polar bear is an endangered species due to environmental climate changes.

To pursue an unambiguous interpretation of an emotional response across a limited group of individuals with great variance in life experience and background can be considered highly challenging. This can answer to why the classification model was not able to correctly distinguish the emotional responses from each other with higher accuracy by making correct predictions.

Bibliography

[1] K. I. T. Mario Tudor, Lorainne Tudor, "[hans berger (1873-1941)–the history of electroencephalography]." https://pubmed.ncbi.nlm.nih.gov/16334737/#:~:text=The%20discovery%20of%20electroencephalography%20(EEG,MRI%2C%20DSA%2C%20etc.). PMID = 16334737, Fetched 2023-03-29.

[2] M. C. Staff, "Eeg (electroencephalogram)." https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875. Fetched 2023-04-03.

[3] Z. Cao, "A review of artificial intelligence for eeg-based braincomputer interfaces and applications." https://journals.sagepub.com/doi/10.26599/BSA.2020.9050017, 2021. Fetched 2023-05-03.

[4] Muneera A. Rasheed et al., "Use of artificial intelligence on electroencephalogram (eeg) waveforms to predict failure in early school grades in children from a rural cohort in pakistan." https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0246236. Fetched 2023-05-03.

[5] A. Gregory, "New artificial intelligence tool can accurately identify cancer." https://www.theguardian.com/society/2023/apr/30/artificial-intelligence-tool-identify-cancer-ai, 2023. Fetched 2023-05-23.

[6] I. D. Nienke van Atteveldt, Tieme W.P. Janssen, "Measuring brain waves in the classroom." https://kids.frontiersin.org/articles/10.3389/frym.2020.00096, 2020. Fetched 2023-04-23.

[7] L. R. Krol, "Eeg 10-10 system with additional information." https://commons.wikimedia.org/wiki/File:EEG_10-10_system_with_additional_information.svg, 2020. Fetched 2022-10-22.

[8] V. Kanade, "Top 10 machine learning algorithms in 2022." https://www.spiceworks.com/tech/artificial-intelligence/articles/top-ml-algorithms/, 2022. Fetched 2022-10-20.

[9] T. U. of Queensland, "Lobes of the brain." https://qbi.uq.edu.au/brain/brain-anatomy/lobes-brain. Fetched 2023-05-05.

[10] C. C. medical professional, "Peripheral nervous system (pns)." https://my.clevelandclinic.org/health/body/23123-peripheral-nervous-system-pns, 2022. Fetched 2023-05-05.

[11] M. G. Suzanne Wakim, "11.4: Nerve impulses." https://bio.libretexts.org/Bookshelves/Human_Biology/Human_Biology_(Wakim_and_Grewal)/11%3A_Nervous_System/11.4%3A_Nerve_Impulses/. Fetched 2023-05-05.

[12] N. I. of Neurological Disorders and Stroke, "Brain basics: Know your brain." https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain. Fetched 2023-05-05.

[13] P. Holck, "nerveceller." https://sml.snl.no/nerveceller, 2022. Fetched 2023-05-05.

[14] T. E. of Encyclopaedia Britannica, "neuron." https://www.britannica.com/science/neuron. Hentet 2023-05-05.

[15] Barbara Pavan et al., "In vitro cell models merging circadian rhythms and brain waves for personalized neuromedicine." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9663312/. PMID = 36387022, Fetched 2023-05-08.

[16] L. R. Krol, "Eeg brainwaves." https://commons.wikimedia.org/wiki/File:EEG_Brainwaves.svg. Fetched 2023-05-05.

[17] Kyung-Ah Kwon et al., "High-speed camera characterization of voluntary eye blinking kinematics." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4043155/. PMID = 23760297, Fetched 2023-05-08.

[18] T. F. E. Wikipedia, "Frekvens." https://no.wikipedia.org/wiki/Frekvens. Fetched 2023-05-08.

[19] Hwi-Young Cho et al., "The effect of neurofeedback on a brain wave and visual perception in stroke: a randomized control trial." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4395689/. PMID = 25931705, Fetched 2023-05-05.

[20] G. A. Silva, "From light to sight (part i): Photons to neural signals." https://www.linkedin.com/pulse/from-light-sight-part-i-photons-neural-signals-gabriel-a-silva/, 2017. Fetched 2023-05-08.

[21] E. E. Martin Gjerde, "Kvalitetsanalyse av grønnsaker ved bruk av hyperspektrale bilder." https://uis.brage.unit.no/uis-xmlui/handle/11250/2774434, 2021. Fetched 2023-05-06.

[22] Ryan D. Zubricky, Joe M. Das, "Neuroanatomy, superior colliculus." https://www.ncbi.nlm.nih.gov/books/NBK544224/, 2022. PMID = 31334944, Fetched 2023-06-15.

[23] Selket, "Ventral-dorsal streams." https://commons.wikimedia.org/wiki/File:Ventral-dorsal_streams.svg. Fetched 2023-05-04.

[24] Chai M. Tyng et al., "The influences of emotion on learning and memory." https://www.frontiersin.org/articles/10.3389/fpsyg.2017.01454/full, 2017. Fetched 2023-05-10.

[25] S. Z.-M. Larry R. Squire, "The medial temporal lobe memory system." https://www.science.org/doi/10.1126/science.1896849. Fetched 2023-05-08.

[26] T. F. E. Wikipedia, "Visual cortex." https://en.wikipedia.org/wiki/Visual_cortex. Fetched 2023-05-07.

[27] B. A. Vogt, "Four regions of cingulate cortex and disease vulnerability." https://web.archive.org/web/20170923110052/http://www.thehumanbrain.org/Vogt.htm. Fetched 2023-05-09.

[28] D. H. Fields, "Difference between thalamus and hypothalamus." http://www.differencebetween.net/science/health/difference-between-thalamus-and-hypothalamus/. Fetched 2023-05-09.

[29] S. Hasselbalch, "Det limbiske system." https://videnscenterfordemens.dk/da/det-limbiske-system. Fetched 2023-05-09.

[30] T. U. of Queensland, "The limbic system." https://qbi.uq.edu.au/brain/brain-anatomy/limbic-system. Fetched 2023-05-09.

[31] U. of Oslo, "Menneskehjerne." https://www.mn.uio.no/ibv/tjenester/kunnskap/plantefys/leksikon/m/menneskehjerne.html. Fetched 2023-05-09.

[32] C. D. Salzman, "Amygdala." https://www.britannica.com/science/amygdala. Fetched 2021-04-21.

[33] Robert P. Vertes et al., "Chapter 16 - thalamus." https://www.sciencedirect.com/science/article/abs/pii/B9780123742452000164. Fetched 2023-05-28.

[34] P. H. Jan K.S. Jansen, "hypothalamus i store medisinske leksikon." https://sml.snl.no/hypothalamus, 2021. Fetched 2023-05-09.

[35] B. Editors, "How does the endocrine system maintain homeostasis." https://biologydictionary.net/how-does-the-endocrine-system-maintain-homeostasis/, 2018. Fetched 2023-05-09.

[36] C. C. medical professional, "Pituitary gland." https://my.clevelandclinic.org/health/body/21459-pituitary-gland. Fetched 2023-05-09.

[37] C. OpenStax, "Figure 35 03 06." https://commons.wikimedia.org/wiki/File:Figure_35_03_06.jpg, 2016. Fetched 2023-05-09.

[38] H. Cristol, "What is dopamine?." https://www.webmd.com/mental-health/what-is-dopamine, 2021. Fetched 2021-05-09.

[39] J. G. Jan K. S. Jansen, "hypofysen i store medisinske leksikon." https://sml.snl.no/hypofysen, 2020. Fetched 2023-05-09.

[40] K. Brekkå, "Bio110 - forelesning 7 - endokrinologi," 2022. Fetched 2023-05-05.

[41] MedlinePlus, "Hormones." https://medlineplus.gov/hormones.html. Fetched 2023-05-05.

[42] C. C. medical professional, "Endocrine system." https://my.clevelandclinic.org/health/articles/21201-endocrine-system. Fetched 2023-05-05.

[43] J. Vazard, "Feeling the unknown: Emotions of uncertainty and their valence." https://link.springer.com/article/10.1007/s10670-022-00583-1, 2022. Fetched 2023-05-05.

[44] A. P. Association, "sexual arousal." https://dictionary.apa.org/sexual-arousal. Fetched 2023-05-05.

[45] O. L. Dictionaties, "arousal." https://www.oxfordlearnersdictionaries.com/definition/english/arousal. Fetched 2023-05-05.

[46] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice." http://www.jstor.org/stable/27857503. Fetched 2023-05-05.

[47] M. E. 1735, "Plutchik-wheel." https://commons.wikimedia.org/wiki/File:Plutchik-wheel.svg. Fetched 2023-05-05.

[48] IMotions, "Top 3 devices for monitoring and measuring brain activity." https://imotions.com/blog/learning/best-practice/top-3-devices-measuring-brain-activity/. Fetched 2023-05-03.

[49] Jaclyn L. Farrens et al., "Electroencephalogram (eeg) recording protocol for cognitive and affective human neuroscience research." https://protocolexchange.researchsquare.com/article/pex-779/v4. Fetched 2023-05-09.

[50] Eric R. Kandel et al., *Principles of neural science*. Hill McGraw, 2021. Fetched 2023-05-09.

[51] Neurohealth, "Brain wave frequencies." https://nhahealth.com/brainwaves-the-language/. Fetched 2023-05-10.

[52] J. L. Farrens et al., "Electroencephalogram (eeg) recording protocol for cognitive and affective human neuroscience research. protocol exchange." https://erpinfo.org/blog/2019/12/12/eeg-recording-protocol, 2019. Fetched 2023-03-06.

[53] A. Kerekes et al., "Correction of eye-movement artifacts of dc-eeg signals." Fetched 2023-05-11.

[54] M. Stackexchange, "What is the purpose of subtracting the mean from data when standardizing?." https://math.stackexchange.com/questions/317114/what-is-the-purpose-of-subtracting-the-mean-from-data-when-standardizing. Fetched 2023-05-11.

[55] T. F. E. Wikipedia, "Algorithms for calculating variance." https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance, 2023. Fetched 2023-05-10.

[56] T. F. E. Wikipedia, "Arithmetic mean." https://en.wikipedia.org/wiki/Arithmetic_mean. Fetched 2023-05-11.

[57] Kristian Hovde Liland et al., "Customized baseline correction." https://www.sciencedirect.com/science/article/pii/S0169743911001535. Fetched 2023-05-10.

[58] H. F. M. B. Paul H. C. Eilers, "Baseline correction with asymmetric least squares smoothing." https://dokumen.tips/documents/baseline-correction-with-asymmetric-least-squares-correction-with-asymmetric.html?page=1, 2005. Fetched 2023-05-11.

[59] S. at Python Package Index, "Baselineremoval 0.1.4." https://pypi.org/project/BaselineRemoval/. Fetched 2023-05-11.

[60] Zhi-Min et al., "Baseline correction using adaptive iteratively reweighted penalized least squares." https://pubs.rsc.org/en/content/articlelanding/2010/an/b922045c. Fetched 2023-05-10.

[61] S. I. Inc., "The tpspline procedure." https://documentation.sas.com/doc/en/statcdc/14.2/statug/statug_tpspline_overview01.htm. Fetched 2023-05-11.

[62] NeuroTechEdu, "Preprocessing." http://learn.neurotechedu.com/preprocessing/. Fetched 2023-05-11.

[63] T. F. E. Wikipedia, "Algorithms for calculating variance." https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance. Fetched 2023-05-10.

[64] S. C. for Computational Neuroscience, "Removing bad channels by visual inspection." https://eeglab.org/tutorials/06_RejectArtifacts/Channel_rejection.html. Fetched 2023-05-11.

[65] S. C. for Computational Neuroscience, "Automated artifact rejection." https://eeglab.org/tutorials/06_RejectArtifacts/cleanrawdata.html. Fetched 2023-05-11.

[66] M. Hargrave, "Standard deviation formula and uses vs. variance." https://www.investopedia.com/terms/s/standarddeviation.asp, 2023. Fetched 2023-05-11.

[67] M. Developers, "Handling bad channels." https://mne.tools/dev/auto_tutorials/preprocessing/15_handling_bad_channels.html. Fetched 2023-05-11.

[68] T. S. community, "Signal processing (scipy.signal)." https://docs.scipy.org/doc/scipy/tutorial/signal.html. Fetched 2023-05-11.

[69] T. S. community, "Interpolation (scipy.interpolate)." https://docs.scipy.org/doc/scipy/tutorial/interpolate.html. Fetched 2023-05-11.

[70] T. F. E. Wikipedia, "Linear interpolation." https://en.wikipedia.org/wiki/Linear_interpolation. Fetched 2023-05-11.

[71] J. Smith, "Mathematics of the discrete fourier transform (dft) with audio applications, second edition." http://ccrma.stanford.edu/~jos/mdft/. Fetched 2023-05-04.

[72] T. F. E. Wikipedia, "Infinite impulse response." https://en.wikipedia.org/wiki/Infinite_impulse_response. Fetched 2023-05-04.

[73] R. Johnsen, "digitalt filter." https://snl.no/digitalt_filter, 2023. Fetched 2023-05-12.

[74] C. Globe, "Difference between fir filter and iir filter." https://circuitglobe.com/difference-between-fir-filter-and-iir-filter.html. Fetched 2023-05-12.

[75] T. F. E. Wikipedia, "Discrete-time fourier transform." https://en.wikipedia.org/wiki/Discrete-time_Fourier_transform. Fetched 2023-05-12.

[76] tutorialspoint, "Fourier transforms." https://www.tutorialspoint.com/signals_and_systems/fourier_transforms.htm. Fetched 2023-05-12.

[77] M. K. Saini, "Inverse discrete-time fourier transform." https://www.tutorialspoint.com/inverse-discrete-time-fourier-transform. Fetched 2023-05-25.

[78] ElectronicsTutorials, "Passive band pass filter." https://www.electronics-tutorials.ws/filter/filter_4.html. Fetched 2023-05-12.

[79] Inductiveload, "Bandwidth 2." https://commons.wikimedia.org/wiki/File:Bandwidth_2.svg. Fetched 2023-05-12.

[80] T. F. E. Wikipedia, "Q factor." https://en.wikipedia.org/wiki/Q_factor. Fetched 2023-05-12.

[81] L. about Electronics, "Quality factor." http://www.learningaboutelectronics.com/Articles/Quality-factor-calculator.php. Fetched 2023-05-12.

[82] T. F. E. Wikipedia, "Ringing artifacts." https://en.wikipedia.org/wiki/Ringing_artifacts. Fetched 2023-05-12.

[83] Aakash30jan, "Box filter." https://commons.wikimedia.org/wiki/File:Box_filter.png. Fetched 2023-05-13.

[84] S. Butterworth, "On the theory of filter amplifiers.." https://www.changpuak.ch/electronics/downloads/On_the_Theory_of_Filter_Amplifiers.pdf. Fetched 2023-05-14.

[85] Hank Zumbahlen et al., "Chapter 8 - analog filters." https://www.sciencedirect.com/science/article/abs/pii/B9780750687034000080. Fetched 2023-05-14.

[86] D. N. AlHinai, "Chapter 1 - introduction to biomedical signal processing and artificial intelligence." https://www.sciencedirect.com/

science/article/abs/pii/B9780128189467000019. Fetched 2023-05-14.

[87] E. Tutorials, "Butterworth filter design." https://www.electronics-tutorials.ws/filter/filter_8.html. Fetched 2023-05-14.

[88] E. Tutorials, "Band stop filter." https://www.electronics-tutorials.ws/filter/band-stop-filter.html. Fetched 2023-05-12.

[89] T. F. E. Wikipedia, "Band-stop filter." https://en.wikipedia.org/wiki/Band-stop_filter. Fetched 2023-05-12.

[90] I. N. Alain de Cheveigné, "Filters: When, why, and how (not) to use them." https://www.sciencedirect.com/science/article/pii/S0896627319301746. Fetched 2023-05-15.

[91] A. K. M.M. Bradley, "Event-related potentials (erps)." https://www.sciencedirect.com/science/article/abs/pii/B9780123750006001543. Fetched 2023-05-15.

[92] M. Developers, "Rejecting bad data spans and breaks." https://mne.tools/stable/auto_tutorials/preprocessing/20_rejecting_bad_data.htmlhttps://mne.tools/stable/auto_tutorials/preprocessing/20_rejecting_bad_data.html. Fetched 2023-05-22.

[93] N. Helseinformatikk, "Emg og nevrografi." https://nhi.no/sykdommer/hjernenervesystem/undersokelser/emg-og-nevrografi/. Fetched 2023-05-22.

[94] Nelly Elsayed et al., "Brain computer interface: Eeg signal preprocessing issues and solutions." https://e-tarjome.com/storage/btn_uploaded/2020-03-11/1583927128_10503-etarjome%20English.pdf. Fetched 2023-05-22.

[95] Arnaud Delorme et al., "Automatic artifact rejection for eeg data using high-order statistics and independent component analysis." https://www.researchgate.net/publication/2552520_Automatic_Artifact_Rejection_For_EEG_Data_Using_High-Order_Statistics_And_Independent_Component_Analysis, 2002. Fetched 2023-05-23.

[96] Lisha Sun et al., "Independent component analysis of eeg signals." https://ieeexplore.ieee.org/document/1504590. Fetched 2023-05-23.

[97] M. Developers, "Repairing artifacts with ica." https://mne.tools/stable/auto_tutorials/preprocessing/40_artifact_correction_ica.html. Fetched 2023-05-23.

[98] Diffzy.com, "Difference between fft and dft.." https://www.diffzy.com/article/difference-between-fft-and-dft-862?utm_content=cmp-true, 2023. Fetched 2023-05-15.

[99] J. O. S. III, "Welch's method." https://ccrma.stanford.edu/~jos/sasp/Welch_s_Method.html, 2011. Fetched 2023-05-15.

[100] M. Escabí, "Chapter 11 - biosignal processing." https://www.sciencedirect.com/science/article/abs/pii/B9780123749796000113. Fetched 2023-05-12.

[101] T. F. E. Wikipedia, "Spectral density estimation." https://en.wikipedia.org/wiki/Spectral_density_estimation. Fetched 2023-05-15.

[102] J. O. S. III, "The periodogram." https://ccrma.stanford.edu/~jos/sasp/Periodogram.html, 2011. Fetched 2023-05-15.

[103] J. O. S. III, "Power spectral density estimation." https://ccrma.stanford.edu/~jos/mdft/Power_Spectral_Density_Estimation.html, 2011. Fetched 2023-05-15.

[104] R. W. S. Inc., "Hanning function." https://help.perforce.com/pv-wave/2017.1/PVWAVE_Online_Help/pvwave.html#page/Foundation/hanning.html, 2019. Fetched 2023-05-15.

[105] I.-H. W. Dah-Jing Jwo, Wei-Yeh Chang, "Windowing techniques, the welch method for improvement of power spectrum estimation." https://www.techscience.com/cmc/v67n3/41590/html, 2020. Fetched 2023-05-15.

[106] B. H. S. Alvar M. Kabe, "Chapter 5 - analysis of continuous and discrete time signals." https://www.sciencedirect.com/science/article/abs/pii/B9780128216156000058, 2020. Fetched 2023-05-15.

[107] K. F. C. Yan Feng Li, "Eliminating the picket fence effect of the fast fourier transform." https://www.sciencedirect.com/science/article/pii/S0010465507004663, 2008. Fetched 2023-05-15.

[108] G. Schuster, "Chapter 5 - analysis of continuous and discrete time signals." http://www.laurent-duval.eu/Documents-Common/Schuster_G_2010_lect_spectrum_upbw.pdf. Fetched 2023-05-15.

[109] J. Otis M. Solomon, "Psd computations using welch's method." https://www.osti.gov/servlets/purl/5688766, 1991. Fetched 2023-05-15.

[110] S. Feike, "Multiple time series classification by using continuous wavelet transformation." https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442, 2020. Fetched 2023-05-22.

[111] J. O. S. III, "Continuous wavelet transform." https://ccrma.stanford.edu/~jos/mdft/Power_Spectral_Density_Estimation.html, 2011. Fetched 2023-05-22.

[112] A. Taspinar, "A guide for using the wavelet transform in machine learning." https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/, 2018. Fetched 2023-05-22.

[113] Hellingspaul, "Stft wvl compare." https://commons.wikimedia.org/wiki/File:STFT_WVL_compare.jpg, 2011. Fetched 2023-05-22.

[114] Gaurav Thakur et al., "The synchrosqueezing algorithm for time-varying spectral analysis: robustness properties and new paleoclimate applications." https://arxiv.org/pdf/1105.0010.pdf, 2012. Fetched 2023-06-07.

[115] T. M. Inc., "Wavelet synchrosqueezing." https://se.mathworks.com/help/wavelet/gs/wavelet-synchrosqueezing.html. Fetched 2023-06-07.

[116] Juncai Xu et al., "Ground-penetrating radar time-frequency analysis method based on synchrosqueezing wavelet transformation," *Journal of Vibroengineering*, vol. 18, no. 1, 2016. Fetched 2023-06-14.

[117] J. Brownlee, "Multi-label classification with deep learning." https://machinelearningmastery.com/multi-label-classification-with-deep-learning/, 2020. Fetched 2023-06-05.

[118] J. Brownlee, "How to choose a feature selection method for machine learning." https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/, 2020. Fetched 2023-06-05.

[119] W.-M. Lee, "Using principal component analysis (pca) for machine learning." https://towardsdatascience.com/using-principal-component-analysis-pca-for-machine-learning-b6e803f5bf1e, 2022. Fetched 2023-06-07.

[120] P. Pub, "Principal component analysis — unsupervised learning model." https://medium.com/hackernoon/principal-component-analysis-unsupervised-learning-model-8f18c7683262, 2019. Fetched 2023-06-07.

[121] G. Developers, "Machine learning glossary - overfitting." https://developers.google.com/machine-learning/glossary#overfitting, 2023. Fetched 2023-06-07.

[122] IBM, "What are naïve bayes classifiers?." https://www.ibm.com/topics/naive-bayes. Fetched 2023-06-07.

[123] O. Harrison, "Machine learning basics with the k-nearest neighbors algorithm." https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761, 2018. Fetched 2023-06-08.

[124] A. Shafi, "Random forest classification with scikit-learn." https://www.datacamp.com/tutorial/random-forests-classifier-python, 2023. Fetched 2023-06-08.

[125] IBM, "What is a decision tree?." https://www.ibm.com/topics/decision-trees. Fetched 2023-06-08.

[126] S. Narkhede, "Understanding confusion matrix." https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 2018. Fetched 2023-06-08.

[127] Errachete, "Binary confusion matrix." https://commons.wikimedia.org/wiki/File:Binary_confusion_matrix.jpg, 2019. Fetched 2023-06-09.

[128] G. Developers, "Classification: Precision and recall." https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. Fetched 2023-06-09.

[129] G. Developers, "Classification: Accuracy." https://developers.google.com/machine-learning/crash-course/classification/accuracy. Fetched 2023-06-09.

[130] R. Kundu, "F1 score in machine learning: Intro calculation." https://www.v7labs.com/blog/f1-score-guide, 2022. Fetched 2023-06-09.

[131] R. Data, "Machine learning - cross validation." https://www.w3schools.com/python/python_ml_cross_validation.asp. Fetched 2023-06-08.

[132] G. Developers, "Validation set: Another partition." https://developers.google.com/machine-learning/crash-course/validation/another-partition. Fetched 2023-06-08.

[133] Simone Palazzo et al., "Generative adversarial networks conditioned by brain signals." https://openaccess.thecvf.com/content_iccv_2017/html/Palazzo_Generative_Adversarial_Networks_ICCV_2017_paper.html, 2017. Fetched 2023-05-03.

[134] Joo Hwan Shin et al., "Wearable eeg electronics for a brain–ai closed-loop system to enhance autonomous machine decision-making." https://www.nature.com/articles/s41528-022-00164-w#citeas, 2022. Fetched 2023-05-04.

[135] Tugce Balli et al., "Emotion recognition based on spatially smooth spectral features of the eeg." 10.1109/NER.2013.6695958, 2013. Fetched 2023-05-20.

[136] Yi-Hung Liu et al., "Emotion recognition from single-trial eeg based on kernel fisher's emotion pattern and imbalanced quasiconformal kernel support vector machine." 10.3390/s140813361, 2014. Fetched 2023-05-20.

[137] B. Kurdi et al., "Introducing the open affective standardized image set (oasis)." https://link.springer.com/article/10.3758/s13428-016-0715-3, 2016. Fetched 2023-05-20.

[138] IMotions, "Enobio 32." https://imotions.com/products/hardware/enobio-32/. Fetched 2023-05-20.

[139] IMotions, "Powering human insight." https://imotions.com/products/imotions-lab/, 2019. Fetched 2023-05-20.

[140] Joseph J.H. Liang et al., "Habituation." https://www.sciencedirect.com/science/article/abs/pii/B9780128096338907866, 2019. Fetched 2023-05-26.

[141] N. Inc., "pandas.dataframe." https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html. Fetched 2023-05-29.

[142] T. S. Community, "Signal processing (scipy.signal)." https://docs.scipy.org/doc/scipy/reference/signal.html. Fetched 2023-05-29.

[143] T. S. Community, "scipy.signal.filtfilt." https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html. Fetched 2023-05-29.

[144] J. O. S. III, "Forward-backward filtering." https://ccrma.stanford.edu/~jos/fp/Forward_Backward_Filtering.html, 2007. Fetched 2023-05-29.

[145] D. Kiracofe, "filtfilt. causal versus non-causal filters." http://www.mechanicalvibration.com/filtfilt_Causal_versus_non_.html. Fetched 2023-05-29.

[146] T. S. Community, "scipy.signal.iirnotch." https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.iirnotch.html. Fetched 2023-05-29.

[147] M. Developers, "mne.filter.filter_data." https://mne.tools/stable/generated/mne.filter.filter_data.html. Fetched 2023-05-29.

[148] Hwi-Young Cho et al., "The effect of neurofeedback on a brain wave and visual perception in stroke: a randomized control trial." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4395689/. PMID = 25931705, Fetched 2023-05-29.

[149] W. Kester, "Section 6 - digital filter." https://www.analog.com/media/en/training-seminars/design-handbooks/mixedsignal_sect6.pdf. Fetched 2023-05-29.

[150] O. b. f. a. o. w. Wikibooks, "Digital signal processing/digital filters." https://en.wikibooks.org/wiki/Digital_Signal_Processing/Digital_Filters, 2022. Fetched 2023-05-29.

[151] scikit-learn developers, "sklearn.decomposition.fastica." https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html. Fetched 2023-05-29.

[152] J. Muradeli, "ssqueezepy 0.6.3." https://pypi.org/project/ssqueezepy/. Fetched 2023-05-29.

[153] scikit-learn Developers, "scikit-learn machine learning in python." https://scikit-learn.org/stable/. Fetched 2023-06-03.

[154] E. Administrator, "Active filters design." https://www.electronicshub.org/active-filters-design/, 2019. Fetched 2023-06-13.

[155] E. Administrator, "The electrical engineering handbok: 2 - digital filters." https://www.sciencedirect.com/science/article/abs/pii/B9780121709600500621, 2005. Fetched 2023-06-13.

[156] G. Developers, "Multi-class neural networks: Softmax." https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax, 2022. Fetched 2023-06-08.

[157] Benedek Kurdi et al., "Introducing the open affective standardized image set (oasis)." https://osf.io/6pnd7/, 2016. Fetched 2023-06-08.

[158] J. Loomis, "The nature of perceptual experience." https://people.psych.ucsb.edu/loomis/jack/nature%20of%20perception.pdf, 1975. Fetched 2023-06-08.

[159] L. S. E. JR., "Perception of natural environments." https://journals.sagepub.com/doi/10.1177/001391656900100105, 1969. Fetched 2023-06-08.

Appendix A

List of stimuli epochs and expected emotional responses

Tree	Boredom	O
Tree	Boredom	O
Tree	Boredom	O
Squirrel	Joy	P
Brick wall	Boredom	O
Naked man	Arousal	P
Cardboard	Boredom	O
Dead person	Sadness	N
Cardboard	Boredom	O
Graveyard	Sadness	N
Half-naked man	Arousal	P
Kitten	Joy	P
Galaxy drawing	Admiration	P
Sleeping cat	Joy	P
Fence	Boredom	O
Graveyard	Sadness	N
Grass	Boredom	O
Puppy	Joy	P
Kittens	Joy	P
Cat	Joy	P
Cat	Joy	P
Flower	Admiration	P
Squirrel	Joy	P
Starved boy	Sadness	N
Squirrel	Joy	P
Mini-city	Boredom	O
Grass	Boredom	O
Grass	Boredom	O
Mugs	Boredom	O
Machine gun	Anger	N
Cockroach	Disgust	N
Snow	Boredom	O
Mugs	Boredom	O
Dead people	Fear	N
Puppy	Joy	P
Oral intercourse	Arousal	P
Dog	Joy	P
Stacked glasses	Boredom	O
Dog	Joy	P
Firehydrant	Boredom	O
Cockroach	Disgust	N
Smiling dog	Joy	P
BBQ	Joy	P
Hurt dog	Sadness	N
Naked men	Arousal	P
Dog and guinea pig	Joy	P
Unconcious person	Sadness	N
Smiling dog	Joy	P
Fence	Boredom	O
Plain landscape	Boredom	O

Tree logs	Boredom	O
Feces	Disgust	N
Firehydrant	Boredom	O
Sushi	Joy	P
Sad man	Sadness	N
Graveyard	Sadness	N
Grass	Boredom	O
Galaxy	Admiration	P
Dog	Joy	P
Half-naked men	Arousal	P
Firehydrant	Boredom	O
Machine gun	Fear(Afraid)	N
Galaxy	Admiration	P
Roses	Admiration	P
Man looking out to water	Sadness	N
Lamb	Joy	P
Kids smiling	Joy	P
Grass	Boredom	O
KKK	Anger	N
Gun	Anger	N
Feces	Disgust	N
Fence	Boredom	O
Lions eating carcass	Disgust	N
Monkey	Joy	P
Half-naked couple	Arousal	P
Frustrated woman	Boredom	N
Grass	Boredom	O
Man with gun	Fear	N
Concrete wall	Boredom	O
Dog	Joy	P
Bloody legs on hospital bed	Disgust	N
Paperclips	Boredom	O
Intercourse	Arousal	P
Naked woman	Arousal	P
Stack of paper	Boredom	O
Piece of paper	Boredom	O
KKK	Anger	N
Pinecones	Boredom	O
Half-naked men	Arousal	P
Pebbles	Boredom	O
Half-naked woman	Arousal	P
Auschwitz	Sadness	N
Racoon	Joy	P
Pebbles	Boredom	O
Naked woman	Arousal	P
Snow	Boredom	O
Rocks	Boredom	O
Rooftiles	Boredom	O
Rooftiles	Boredom	O
Rooftiles	Boredom	O

Eye tumor	Disgust	N
Pinecones	Boredom	O
Boy crying	Sadness	N
Confused older woman	Sadness	N
Dog and kitten	Joy	P
Pebbles	Boredom	O
Scary face	Fear	N
Scary face	Fear	N
Naked man	Arousal	P
Concrete wall	Boredom	O
Shark	Joy	P
Needle injection	Disgust	N
Manholes	Boredom	O
Snow	Boredom	O
Severed finger	Disgust	N
Girls smiling	Joy	P
Tree logs	Boredom	O
Concrete wall	Boredom	O
Galaxy	Admiration	P
Thread	Boredom	O

NOP	N = Negative, O = Neutral, P = Positive
Boredom	O
Admiration	P
Joy	P
Disgust	N
Arousal	P
Sadness	N
Fear	N
Anger	N

Appendix B

EEG recording system electrode protocol

Ch1	P7
Ch2	P4
Ch3	Cz
Ch4	Pz
Ch5	P3
Ch6	P8
Ch7	O1
Ch8	O2
Ch9	T8
Ch10	F8
Ch11	C4
Ch12	F4
Ch13	Fp2
Ch14	Fz
Ch15	C3
Ch16	F3
Ch17	Fp1
Ch18	T7
Ch19	F7
Ch20	Oz
Ch21	PO4
Ch22	FC6
Ch23	FC2
Ch24	AF4
Ch25	CP6
Ch26	CP2
Ch27	CP1
Ch28	CP5
Ch29	FC1
Ch30	FC5
Ch31	AF3
Ch32	PO3

Appendix C

Extra results from additional ML experiments

Logistic regression:

```
Accuracy: 0.425531914893617
              precision    recall  f1-score   support

    Negative       0.50      0.38      0.43        13
     Neutral       0.44      0.41      0.42        17
    Positive       0.38      0.47      0.42        17

    accuracy                           0.43        47
   macro avg       0.44      0.42      0.43        47
weighted avg       0.43      0.43      0.43        47
```

Figure C.1

SVM:

```
Accuracy: 0.40425531914893614
C:\Users\gjerd\AppData\Local\Programs\Python\Python37\
e are ill-defined and being set to 0.0 in labels with
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

    Negative       0.00      0.00      0.00        13
     Neutral       0.47      0.53      0.50        17
    Positive       0.36      0.59      0.44        17

    accuracy                           0.40        47
   macro avg       0.28      0.37      0.31        47
weighted avg       0.30      0.40      0.34        47
```

Figure C.2

```
Accuracy: 0.40425531914893614
              precision    recall  f1-score   support

    Negative       0.50      0.15      0.24        13
     Neutral       0.41      0.53      0.46        17
    Positive       0.38      0.47      0.42        17

    accuracy                           0.40        47
   macro avg       0.43      0.38      0.37        47
weighted avg       0.42      0.40      0.38        47
```

Figure C.3

Normalised histograms:

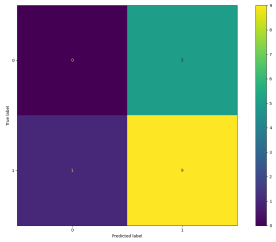


Figure C.4

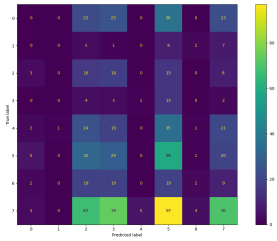


Figure C.5

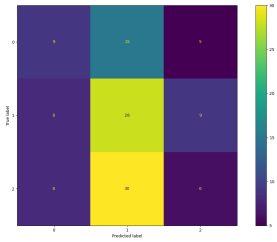


Figure C.6

Gridsearch:

```
Fitting 4 folds for each of 210 candidates, totalling 840 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:  1.0min
[Parallel(n_jobs=-1)]: Done 184 tasks      | elapsed:  5.2min
[Parallel(n_jobs=-1)]: Done 434 tasks      | elapsed: 12.6min
[Parallel(n_jobs=-1)]: Done 784 tasks      | elapsed: 22.7min
[Parallel(n_jobs=-1)]: Done 840 out of 840 | elapsed: 24.7min finished
0.4048780487804878
```

Figure C.7

Hjort descriptors:

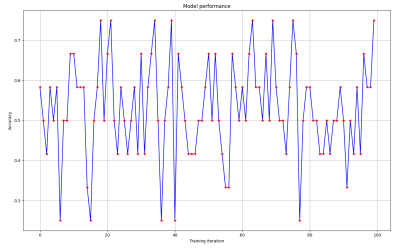


Figure C.8

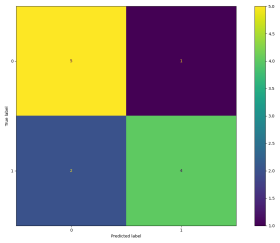


Figure C.9

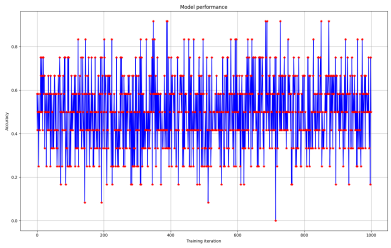


Figure C.10

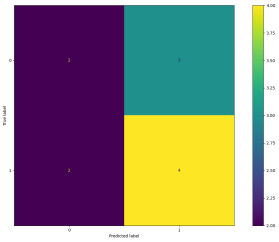


Figure C.11

Appendix D

Code

All source code can be accessed through the open GitHub Repository in the following link: GitHub Repository.

README.md:
This is code developed under a master thesis project for the purpose of analysing EEG data and attempt to train a machine learning model to classify different emotional responses. The thesis is a part of the master's degree in Robottechnology and signalprocessing with specialisation in health technology at University of Stavanger.

Simple user guide: Use final.py-script for preprocessing of smaller sampled data, e.g. 300.000 samples. Make sure to set the correct values for bad channel detection and bad epoch rejection when saving features.

Use final2.py for preprocessing smaller data using Zhang-fit algorithm for baselining.

Use the script in longer_data_preprocessing.py for preprocessing and extracting features for longer data sets, e.g. 2.250.000 samples

The file model.py is for the purpose of doing iterative model training and testing.

ml_LR.py, ml_NB.py, ml_RF.py, ml_SVM.py, ml_kNN.py and ml_nn.py are files for proposed methods for development of different machine learning models.

ssqpcaNN .npy-files are the features extracted using synchrosqueezed wavelet transformation with PCA of 12 components. First N refers to the respondent number and the second N refers to the total number of classes. Respondent l is named l for large, which indicates the features extracted from one of the larger data sets.

Appendix E

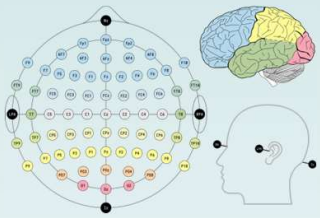
Poster

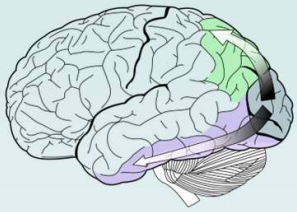
AI assisted classification of emotional responses based on visual stimuli

Martin Gjerde, M.Sc. Robot technology and signal processing with specialisation in health technology
University of Stavanger,, Department of Electrical Engineering and Computer Science

Introduction

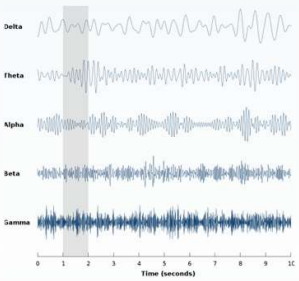
This project will be a part of research to determine if it is possible to make use of machine learning models to classify human's emotional responses based on visual stimuli. With use of electroencephalography(EEG), data will be collected from volunteering respondents that will be exposed to images from a standardised image set. When humans are intrigued or feel a spike of enjoyment, it sends off a spike of neurotic energy in the brain. Electroencephalography can be used to measure the brain's activity by measuring the electric charge between synapses when the neurons are stimulated. The main achievement from the results of this project will be to see if it is possible to classify people's EEG responses and if they are finding what they look at appealing to them or not. One of the challenges in this thesis will be to make sufficient use of known signal processing methods to be able to extract the correct information based on the literature available. Then the epochs will segment the signal, dividing the event related potentials from one another. By later examining the resulting filtered data in more detail, hopefully a pattern of EEG epoch recordings will be recognised and make it possible to correlate to the specific themed image that the respondent is looking at. These patterns will then be extracted as specific features to be set as trigger points that will be detected by a trained machine learning model.

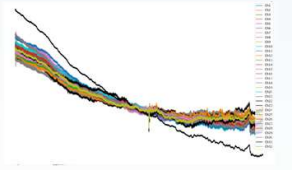


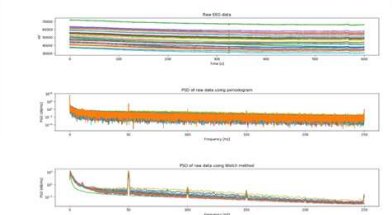


Methodology

For the experimental setup, the standard 32-channel EEG protocol proposed in the NIC2-software that the Enobio 2 was running on was used. IMotions' software displayed a pop-up that asked the respondents for their consent to collect their data during the stimuli exposure. This had to be ticked-off by the respondent themselves for the recording to start. The respondents was left alone in the experiment room for the entire duration of the recording and asked to make themselves comfortable for the time being and not to make too many abrupt movements to avoid unwanted artefacts in the EEG data. All data was anonymised and labelled as anon_1, anon_2,...anon_N in the data set. During this project the Enobio 32-channel electroencephalography device was used to perform the recording of the respondents for collection of data. iMotions' lab streaming layer platform was used to synchronise the visual stimuli and the EEG recording. Electroconductive gel was used to make contact between the EEG electrodes and the respondents' scalps. All the equipment and computer software were located at the cognitive lab at the University of Stavanger; therefore, all experiments and recordings were done at this location.

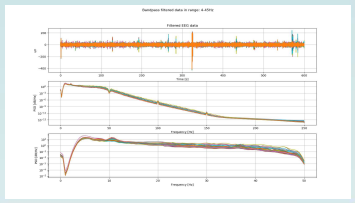


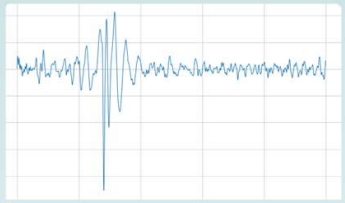


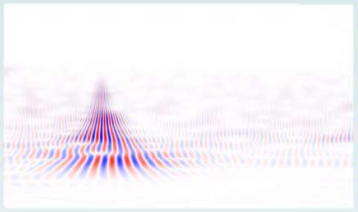


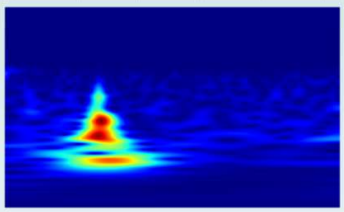
Results

The raw data was processed step by step using traditional signal processing techniques. Based on channels' standard deviations, bad channels were removed from the data automatically and interpolated using linear interpolation methods. Notch filtering was performed to remove any power line noise before performing band pass filtering between 4-45Hz to highlight the underlying EEG data that was of interest for this thesis project and attenuate the rest. All channel data from each respondent were group averaged to perform an event related potential(ERP) with a 5 second interval, segmenting the time of stimuli exposure. Bad epochs(epochs containing irrelevant oscillations) were then removed from the data set, which were maximum 2-3 epochs each respondent, making it a small fraction of only losing approximately at most 1/60 of total epoch data.



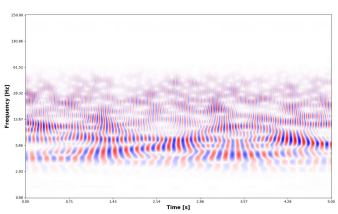


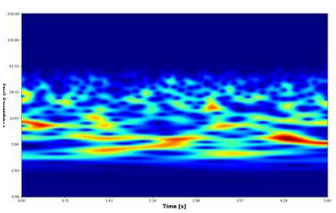




Conclusion

The Morlet wavelet was used to perform a time-frequency analysis of the epochs and the resulting wavelet transformed data from each epoch were then extracted as features for training the machine learning model. As the model is currently under development, no results from the machine learning model is not yet ready for discussion, therefore the answers for the project's main question is still inconclusive and will hopefully be complete in the report in mid-June. Reports and articles discussing related work showed promising results, but with using highly complex machine learning techniques for classification model training and testing. For this project it is desirable to test out easier algorithms for basic model training such as naïve bayes, k-nearest neighbours and random forest for the multi-label classifier.





Acknowledgements

All respondents were volunteering men and women with no prior health conditions that would complicate the data collection and results. All code were developed in python using specified packages like SciPy and Ssqueezypy to preprocess the data, represent it in several ways and perform the epoch signal feature extraction.