



**BRNO UNIVERSITY OF TECHNOLOGY**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF COMPUTER SYSTEMS**

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

**FEATURE EXTRACTION AND SELECTION  
FOR EMOTIONS DETECTION FROM EEG  
SIGNALS USING PYTHON**

FEATURE EXTRACTION AND SELECTION FOR EMOTIONS DETECTION FROM EEG SIGNALS  
USING PYTHON

**BACHELOR'S THESIS**

BAKALÁŘSKÁ PRÁCE

**AUTHOR**

AUTOR PRÁCE

**SIMONA ČEŠKOVÁ**

**SUPERVISOR**

VEDOUČÍ PRÁCE

**Dr. SOYIBA JAWED, MSc**

**BRNO 2023**

# Bachelor's Thesis Assignment



143557

Institut: Department of Computer Systems (UPSY)  
Student: **Češková Simona**  
Programme: Information Technology  
Specialization: Information Technology  
Title: **Feature Extraction and Selection for Emotions Detection from EEG Signals Using Python**  
Category: Signal Processing  
Academic year: 2022/23

## Assignment:

1. Study and learn about the various emotions.
2. Get acquainted with feature extraction and selection methods, and their application to the detection of emotions.
3. Find out the challenges for emotion detection and the limitations of the existing methods.
4. Design a model for feature extraction and feature selection of emotion detection.
5. Implement and evaluate the designed model.
6. Conduct critical analysis and discuss the achieved results and their contribution.

## Literature:

- Based on the supervisor's recommendation.

## Requirements for the semestral defence:

- Fulfillment of Items 1 to 3 of the assignment.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Jawed Soyiba, Dr., MSc**  
Head of Department: Sekanina Lukáš, prof. Ing., Ph.D.  
Beginning of work: 1.11.2022  
Submission deadline: 10.5.2023  
Approval date: 31.10.2022

## Abstract

This work deals with the extraction and selection of features of EEG signals for emotion detection. Processing these signals included steps such as signal pre-processing, extraction of its features and subsequent selection of features. For verification of the correct implementation, the extraction and selection results were evaluated by a machine learning algorithm. This work works with the already measured DREAMER dataset.

## Abstrakt

Tato práce se zabývá extrakcí a selekcí vlastností EEG signálů pro detekci emocí. Proces zpracovávání těchto signálů obsahuje kroky jako předzpracování signálu, extrakci jeho vlastností a následnou jejich selekci. Pro ověření správné implementace byly výsledky extrakce a selekce ohodnoceny algoritmem strojového učení. Tato práce pracuje s již naměřeným datovým souborem DREAMER.

## Keywords

EEG, signals, signal pre-processing, feature extraction, feature selection, emotions classification

## Klíčová slova

EEG, signály, předzpracování signálů, extrakce vlastností, selekce vlastností, klasifikace emocí

## Reference

ČEŠKOVÁ, Simona. *Feature Extraction and Selection for Emotions Detection from EEG Signals Using Python*. Brno, 2023. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Dr. Soyiba Jawed, MSc

## Rozšířený abstrakt

Tato práce je zaměřena na použití metod v oblasti detekce emocí pomocí EEG signálů. Použitý dataset EEG signálů se nazývá DREAMER a byla zde uplatněna neinvazivní metoda měření. Detekce emocí lze využít při lékařských posudcích a léčbě psychických onemocnění.

Hlavní částí této práce byla extrakce a selekce vlastností, které lze získat z EEG signálů, naměřených z 23 subjektů, kteří sledovali 18 krátkých videí. Na konci každého videa každý subjekt ohodnotil, jak se cítil pomocí skóre na stupnici 1–5 pro emoční dimenzi valence a arousalu.

Celý proces probíhal ve čtyřech krocích a každý z těchto kroků navazuje na ten předchozí:

1. Příprava EEG signálů
2. Extrakce vlastností signálu
3. Selekcce nejdůležitějších vlastností
4. Ohodnocení předchozích kroků

Pro dosažení co nejpřesnějších výsledků bylo nutné neupravený EEG signál vyčistit od šumu, mrknutí oka a dalších nechtěných zásahů do signálu, které pronikly při nahrávání. K tomu byl použit pásmový filtr, který propouštěl frekvence mezi 0,5–40,0Hz, nástroj ICA a manuální čištění komplexnějších segmentů. Vzorkovací frekvence byla snížena na 64Hz. Signál byl poté připraven pro hlavní část mé práce, nalezení nejlepšího řešení a kombinace metod extrakce a selekce vlastností signálu.

Před extrakcí bylo nutné signál rozdělit na jednotlivá frekvenční pásma a na 4 skupiny dle umístění elektrod, z kterých signál pochází, z lidského mozku. Frekvenční pásma rozdělují EEG signál na pět druhů vln – delta, theta, alfa, beta a gama. Osm elektrod se při měření nacházelo ve frontální části a po dvou elektrodách se dalších šest nacházelo v okcipitální, temporální a parientální části mozku.

Časová a frekvenční doména signálu poskytovala vlastnosti, které charakterizují signál, proto jsem extrahovala statistické vlastnosti, Hjortovy parametry, entropie a výkonovou spektrální hustotu. Všechny vlastnosti byly zvlášť extrahované pro každé frekvenční pásmo a pro každou ze čtrnácti elektrod. Extrahovaná data jsem normalizovala, protože poté dosahovala lepších výsledků.

Dohromady jsem dosáhla celkového počtu 10220 extrahování vlastností, které tvořily vektor, pro jeden subjekt a jedno video. Celková matice vektorů obsahovala tedy 414 vektorů. Při tomto množství bylo nutné některá nepodstatná data vyřadit, neboli použít metodu selekce. Navrhla jsem 3 možné úpravy skóre a 3 metody selekce. Během testování se ukázalo, že nejlepší výsledky pro selekci byla schopná dodat kombinace binárního skóre s lineární diskriminační analýzou. Binární skóre bylo rozděleno na nízkou a vysokou valenci a arousal. Ostatní metody dosáhly negativních výsledků s výrazně menší přesností měření.

Ohodnocení proběhlo s použitím čtyř klasifikátorů strojového učení a nejlepší výsledky jsem dosáhla pomocí naivní Bayesovské klasifikace. Pro objektivní hodnocení jsem použila k-násobnou křížovou validaci. Průběh selekce a ohodnocení selekce bylo zvlášť rozděleno pro emoční dimenzi valence a arousalu. Proto byla metoda implementována dvakrát s dvěma klasifikátory, natrénovanými pro danou emoční dimenzi.

Rozdělení elektrod do jednotlivých částí dokázalo, že vlastnosti extrahované z frontální části mozku dosáhly největší přesnosti. Tento výsledek jsem ověřila s literaturou. K zlepšení

výsledků jsem dále testovala jakou přesnost byly schopné vytvořit vlastnosti extrahované z časové a frekvenční domény. Frekvenční doména se jasně ukázala jako lepší a největší přesnost byla dosažena s použitím výkonové spektrální hustoty.

Výsledné řešení extrakce emocí se skládá z použití pouze vlastností signálu jeho frekvenční domény a z frontální části rozdělení mozku, které byly následně normalizované. Řešení selekce bylo složeno z binárního označení skóre sebehodnocení subjektů a lineární diskriminační analýzy.

Toto řešení ohodnocené naivní Bayesovskou klasifikací dosáhlo průměrné přesnosti 98,05%. Emoční dimenze valence dosáhla průměrné přesnosti 97,61% a dimenze arousalu 98,49%.

# Feature Extraction and Selection for Emotions Detection from EEG Signals Using Python

## Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mrs. Dr. Soyiba Jawed I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....  
Simona Češková  
May 9, 2023

## Acknowledgements

I want to thank Mrs. Dr. Soyiba Jawed for all the consultations and help in finding the solution to my thesis.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theoretical introduction</b>	<b>4</b>
2.1	Electroencephalography . . . . .	4
2.2	EEG Signals . . . . .	4
2.3	Division of the Human Brain . . . . .	6
2.4	Valence–Arousal Model . . . . .	7
2.5	Features . . . . .	7
2.6	Methods for feature selection and classification . . . . .	9
<b>3</b>	<b>Data</b>	<b>13</b>
3.1	Dataset . . . . .	13
3.2	Signal–preprocessing . . . . .	15
<b>4</b>	<b>Proposed solution</b>	<b>18</b>
4.1	Feature Extraction . . . . .	19
4.2	Feature Selection . . . . .	20
4.3	Classification of Emotions . . . . .	24
<b>5</b>	<b>Validation and Evaluation</b>	<b>28</b>
5.1	Validations . . . . .	28
5.2	Evaluation . . . . .	30
5.3	Proposed Feature Selection Method . . . . .	33
5.4	Discussion . . . . .	35
<b>6</b>	<b>Implementation of My Solution</b>	<b>36</b>
6.1	Dependencies and Structure of Files . . . . .	36
6.2	Operations with Data . . . . .	36
6.3	Pre–processing . . . . .	38
6.4	Feature Extraction . . . . .	38
6.5	Feature Selection . . . . .	40
6.6	Learning Classifiers . . . . .	41
6.7	Process of Selection and its Classification Evaluation . . . . .	42
<b>7</b>	<b>Results</b>	<b>44</b>
7.1	Improvement Proposal . . . . .	44
	<b>Bibliography</b>	<b>45</b>

# List of abbreviations

EEG	Electroencephalography. 1, 3, 4, 6, 7, 9, 13, 16–18, 31, 34, 36, 38, 44
FFT	Fast Fourier transform. 1, 9
KNN	$K$ -nearest neighbors. 1, 28, 35
LDA	Linear discriminant analysis. 1, 10, 30, 31, 33–35, 40, 42, 43
mRMR	Minimum redundancy and maximum relevance. 1, 10, 22, 23, 30, 40
NB	Naive Bayes. 1, 35, 42
PCA	Principal component analysis. 1, 9, 11, 22, 23, 30, 38, 40
PSD	Power spectral density. 1, 8, 9, 34
PTP	Peak to peak. 1, 20
RF	Random forest. 1, 35
RMS	Root mean square. 1, 20
ROC	Receiver operating characteristic. 1, 35
STD	Standart deviation. 1, 20
SVM	Support vector machine. 1, 11, 29–31, 41, 42

# Chapter 1

## Introduction

The aim of this thesis is to get acquainted with the process of emotion prediction by using EEG signals. Humans feel emotions depending on their mental state, substantiated by some event or action. Therefore, after measuring data affected by some events, it can be determined what emotions the human felt at the time.

My main motivation for this thesis is that the results of emotion prediction can be used in medicine to detect depression and subsequent treatment. By using the prediction of emotions for a mentally ill person, it is possible to determine how much their health differs from that of a healthy person to determine their treatment accordingly. The mentally ill person's measured values tend to be closer to negative emotions compared to a healthy person.

Firstly this process involved understanding the principles of EEG and its other parts related to this topic. Learning about EEG led to raising important questions such as: What EEG signal is, and how the human brain is capable of generating one? How differ the individual measured signals? Answers to these subjects are explained in chapter 2. This thesis worked with two emotions, valence and arousal. Their purpose was also explained in chapter 2, including all other algorithms and methods further used.

Secondly, a dataset which enables implementation and testing is named DREAMER and defined in chapter 3.

Further, an essential part constitutes a feature extraction and selection, which comes next after cleaning signal during 3.2, because it could contain some unwanted noise. Feature extraction clarified in 4.1 comprises the practical use of several concepts, such as time-domain, frequency-domain and entropy features. Feature selection follows up at 4.3, and it works with the features and methods to select only the best ones. Finally, emotion classification 4.3 evaluated the entire process and affirmed new findings.

An essential step in the course of the solution was experimentation and testing 5, which led to the compilation of the final solution 5.3 and its implementation 6.

# Chapter 2

## Theoretical introduction

This section explains crucial aspects of EEG studying subsequent to practical usage. Described methods and terms are further used in the following chapters Data 3 Proposed solution 4.

### 2.1 Electroencephalography

Electroencephalography is a technique to observe how the brain functions [9]. This technique is based on non-invasive studying electrical brain activity, which is caused by brain cells named neurons. The observation itself takes place with the help of electrodes located on the scalp [28]. They recognize neurophysiological response [30] and Abnormalities can be found in collected data in contrast with the brain's normal function and then be classified as Parkinson's disease or Alzheimer's disease [8]. EEG signals do not have a usage field only in medicine for diagnosing people suffering from the neurological problems. Their other uses include identity authentication, emotion state recognition and design of the Brain computer interface systems [24].

### 2.2 EEG Signals

The EEG signals are stochastic signals formed by a group of sinusoids which differ in the frequency spectrum. The frequency range of the whole signal is between 0 and 100 Hz [45][19]. Also, the EEG signal includes unwanted parts called artifacts. In this group of unsettling signal sections are eye movements, and muscle or lateral eye movements [45] [15].

The 10 20 system for placement of electrodes was used in this thesis, mentioned in chapter 3.

10–20 system is based on the positions of sensors called electrodes on the head scalp. It means, that each electrode has to be placed after each step in distances accorded to 10 or 20% showed at 2.1. Apart from the necessary distance, their placement is on anatomical landmarks which it is considered to be nasion, inion and preauricular points [18].

#### 2.2.1 Frequency Waves

From EEG signals can be extracted waves with different frequency spectrums. Most important information can be found in the band around 0.5 to 30 Hz [19].

The application of low-pass filters is necessary for removing noise and attenuating the signal. High-pass filters are used to cut out slow fluctuation or direct current com-



## Gamma $\gamma$

The last frequency spectrum is within 30 – 100 Hz [22] for the gamma wave. Gamma waves are recorded only occasionally when a human brain is processing more than one different sense at the same time. This process is called cross-modal processing [22].

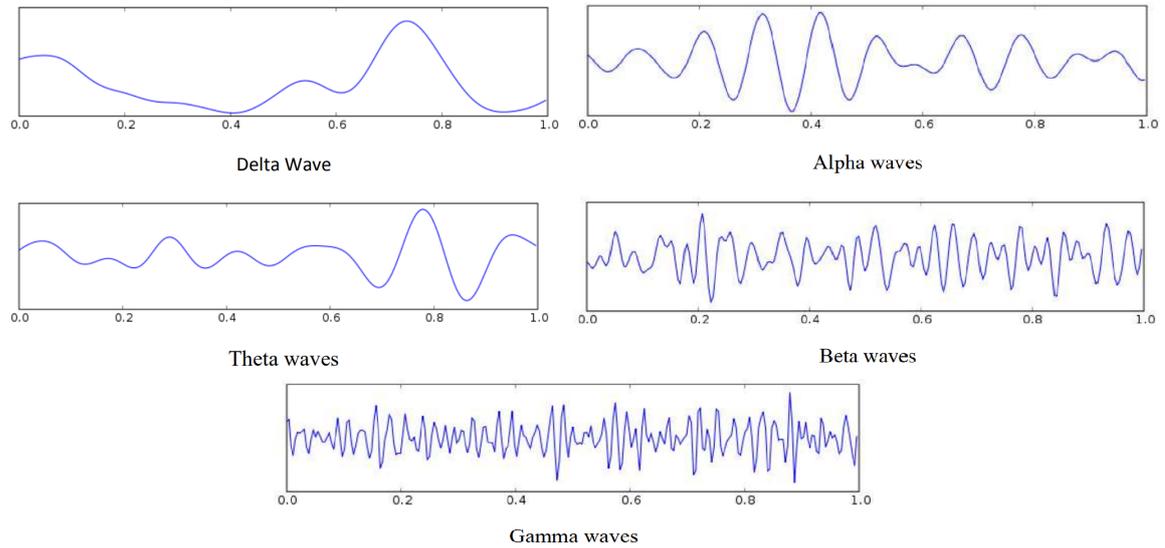


Figure 2.2: These figures show different behaviour of the EEG waves during 1 second. Figures were taken from [22].

## 2.3 Division of the Human Brain

The human brain can be subdivided into three parts – cerebellum, brainstem and cerebrum shown in this figure 2.3.

Cerebrum is further split into left and right cerebral hemispheres [32]. the left hemisphere tends to be responsible for positive emotions and in the opposite the right hemisphere is more associated with negative emotions [1].

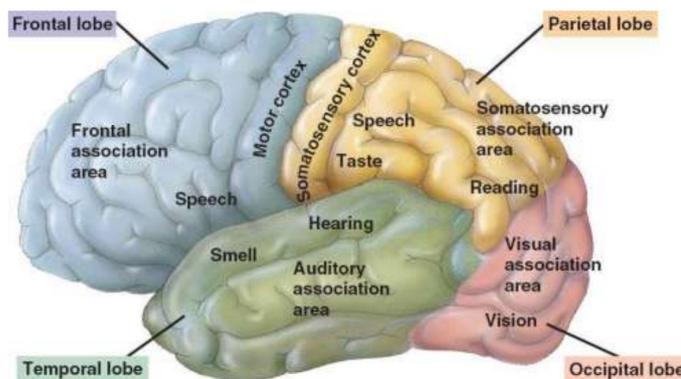


Figure 2.3: Partition of human brain from [22].

Both left and right hemispheres are partitioned into Frontal, Parietal, Temporal, Occipital, Insular and Limbic lobes. Each of them arranges a different tasks for the body [22].

Frontal lobe signicates majority of activity in whole cerebrum and is responsible for emotion control. Temporal lobe is responsible for emotional perception [3]. Another feature of the frontal lobe is evoking a surprise. For temporal and occipital lobes it is happiness and disgust. The parietal lobe may cause sadness [1].

Cerebellum manages complex body movements [22]. as motor coordination, posture and balance. Cerebellum lies under the cerebrum’s occipital and part of the temporal lobe. Cerebellar peduncles are structures that attach cerebellum to brainstem. [32].

## 2.4 Valence–Arousal Model

Valence and arousal are crucial emotions for assembling an emotion classification algorithm, because their ratio describes possible found emotion [13].

Valence and arousal 2D model is in figure 4.7 representing complex emotions.

The numeric values of valence and arousal coordinate to point on a 2D model 2.4. Valence dimension measures if a human has negative or positive feelings and the arousal dimension measures whether a human feels bored or excited. [13].

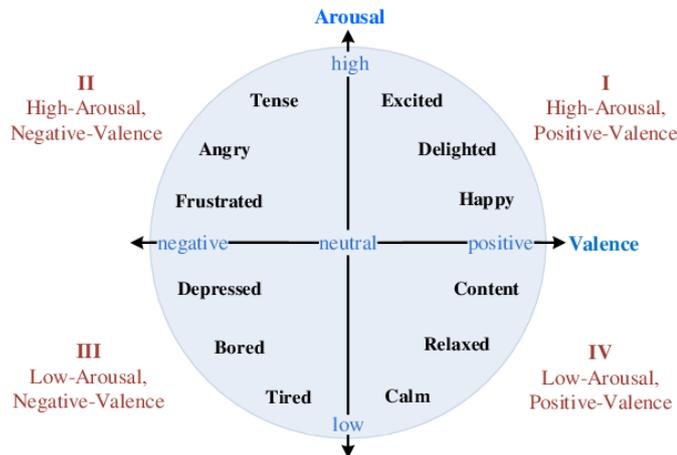


Figure 2.4: Valence and arousal model from [57].

## 2.5 Features

The process of searching for important pieces of information from EEG signals involves analysing a signal through its features. The signal can be represented in the time–domain or frequency–domain [47]. The representation is shown in figure 2.5.

If signal analysis aims to discover a signal composition, then it is fitting to use representation in the frequency domain. Fourier analysis can be used in this situation [36].

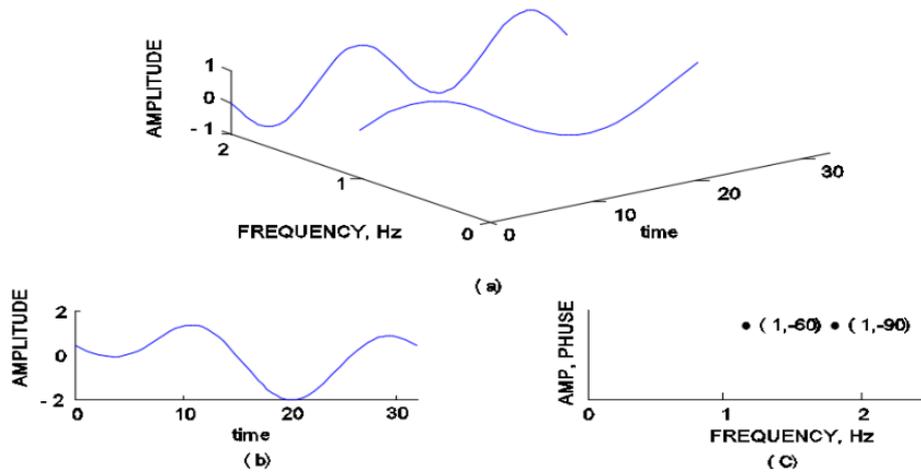


Figure 2.5: Figure shows a representation of the same signal (a) in time domain (b) and frequency domain (c). Figure is from [36].

### 2.5.1 Time-domain Features

Time-Domain includes features:

- Statistical: mean, median, variance, standard deviation, skewness, and kurtosis [47]
- Zero-crossing rate [47]
- Entropies [47]

### Hjorth Parameters

Three Hjorth Parameters are activity, mobility and complexity. Each of them is used to display different statistical properties [39].

Activity corresponds to the variance of the signal and measures the power spectrum in the frequency domain. It means that if the activity values are higher, higher frequencies are more common. This also applies otherwise with low frequencies [39].

Mobility returns a proportion of the standard deviation of the power spectrum [39].

Complexity displays a resemblance between a signal and a pure sine wave. If the signal approaches a shape similar to the sine wave, complexity gets near to number 1 [39].

$$activity = var(y(t)) \quad mobility = \sqrt{\frac{var(y'(t))}{var(y(t))}} \quad complexity = \frac{mobility(y'(t))}{mobility(y(t))} \quad [39] \quad (2.1)$$

Where  $y(t)$  is a function in time-domain and  $y'(t)$  is a derivation of the function.

### 2.5.2 Frequency-Domain Features

The power spectral density is a frequency-domain feature. A signal contains frequency components, and PSD facilitates indicating these components. It means that PSD (figure 2.6) calculates power for each frequency component and depicts their differences in distribution [11].

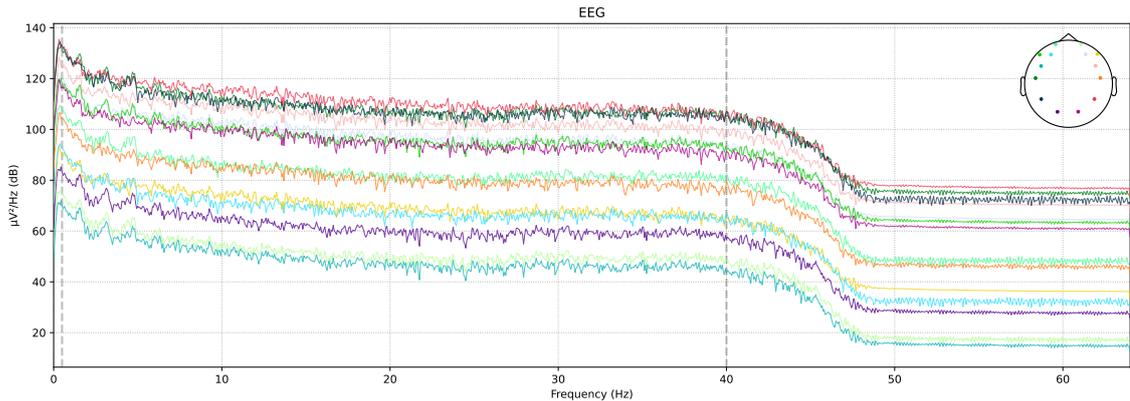


Figure 2.6: Power spectral density of 14 channel EEG signal dataset DREAMER. Subject number 10 and movie clip number 12.

PSD can be assessed by welch method. It differs from the fast Fourier Transform in that the resulting frequency spectrum is smoother because the Welch method divides the data into equal-sized units whose signals are attenuated at the ends of the unit's range. Therefore, there are no value jumps during the frequency domain representation. The disadvantage is that, in contrast to the FFT method, some information is lost, which is a reason for the occurrence of overlapping when parts of the signal partially overlap. The calculation of Welch's method takes place by computing the squared root of the FFT for each divided section of the signal. This data gives the number of an estimation of the spectral density, and the average of all provides the result with PSD estimation [43].

## 2.6 Methods for feature selection and classification

In this section, can be found a description of concepts, algorithms and methods for the classification and selection of extracted EEG features. The methods used for selection are described in 2.6.2, 2.6.3, 2.6.4 and 2.6.5. Algorithms for evaluation of feature selection are shown in 2.6.6.

### 2.6.1 Variance

Variance is a concept that I used in conjunction with PCA 2.6.5. Variance is the standard deviation power two.

$$s^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1} [26] \quad (2.2)$$

Its theorem says that after computing a mean value of the whole data ( $\bar{x}$ ), where  $x$  is a sample, every value is compared with the mean. A sum of distances (distance is always a positive value) between the mean and every value is divided by an amount of data values [26].

If the variance value is small, data are concentrated around the mean and their sizes do not differ wildly. However, the bigger valence is the distance between the mean and the number is more significant. This means that data are quite different sizes [26].

### 2.6.2 Pearson Correlation Coefficient

Pearson Correlation Coefficient is between two values  $x, y$  and represents a linear relationship between these two values [26].

$$r = \frac{cov(x, y)}{\sigma_x \sigma_y} \quad \text{where} \quad cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n - 1} \quad [26] \quad (2.3)$$

It is calculated as covariance, which represents an association between two random variables, for  $x$  and  $y$ , then divided by a multiple of standard deviations from these variables [26].

### 2.6.3 Minimum Redundancy and Maximum Relevance

A group of features selected this way expresses fewer data but more pieces of information for the classification algorithm [5].

AlgorithmmRMR chooses features according to two criteria. The first one is that features with a high mutual affinity are reduced. This step ensures minimum redundancy. The second criterion signifies finding features affecting the target variable [42]. the target variable is an aim for prediction from the data. Therefore is used as a central variable to which features are linked [33]. the greedy search algorithm performs the process of picking features. In every step, one feature is selected [42].

### 2.6.4 Linear Discriminant Analysis

To drop a number of features from data, LDA dimensionally decreases duplicative and conditional features. Decreasing dimensions means that LDA reduces the number of dimensions in the data. LDA uses between-class variance (separability), a number calculated as a mean of feature one minus the mean of feature two and a within class variance (absolute value of the difference between the mean and instance of each feature). An output is a lower dimensional space where within class variance is lower, and between class variance is bigger [44].

### 2.6.5 Principal Component Analysis

PCA is a linear method for eliminating redundant dimensions of the data to achieve a data (features) reduction but still keeping the majority of the information. [7].

Data reduction is needed to prevent overfitting a model while classifying features or because specific data carries less information than others [56].

PCA transforms the data so that a new first dimension contains shown at figure 2.7 information with the highest variance. Then, a new second dimension includes data with the second highest variance and continues this way with other dimensions. Because of how great the variance number is, the data hold more information [7].

PCA also excludes features with significant correlation between them because they are not usable by themselves. An output is dimensionally reduced data that represents an ideal collection of features [56].

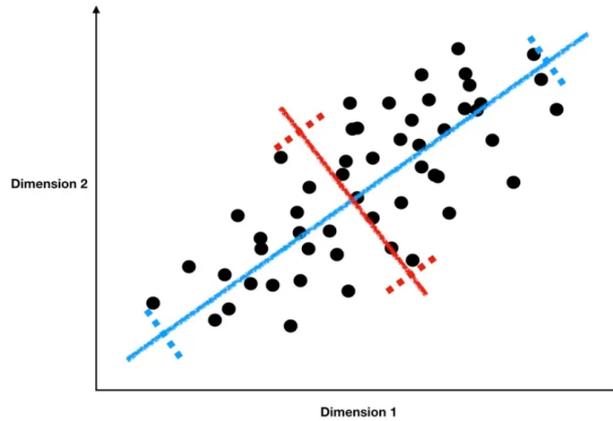


Figure 2.7: PCA over two dimensional data from [56].

### 2.6.6 Classifiers

There are several types of learning classifiers and those used in section 5.2 are described:

- **Support vector machine** is a learning classifier proposed for data classification. Before using SVM for classification, the data has to be labelled because SVM assorts them into groups. For the separation, SVM uses a hyperplane [53]. the hyperplane creates a border between the two groups of data, sorted based on a different classes they tend to [14]. Choosing the right hyperplane aims to maximize the length between separated data [53] [14]. A dashed line in figure 2.8 indicates the hyperplane with support vectors on the sides. The data is formed of data points (little stars and triangles), and if some are located near the hyperplane, they are called support vector [27] [14].

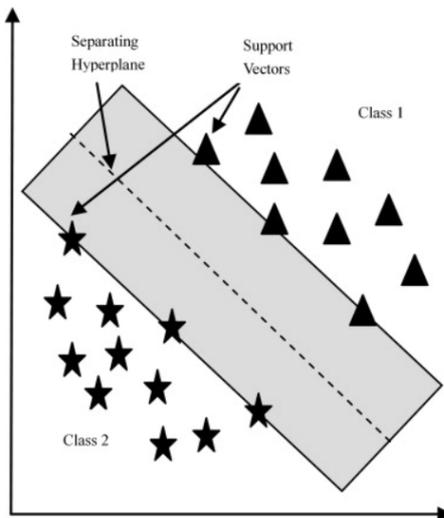


Figure 2.8: Support Vector Machine from [27].

- The ***k*-nearest neighbor** is another classification tool that works by making predictions from its stored data. The number *k* symbolizes the value for which it searches the number of records and selects the most identical to the predicted one [55].
- The **naive Bayes** classifier is a probabilistic learning classifier. One of the methods that can be used to solve Naive Bayes is the Gaussian method. This method is based on the Bayes theorem. The data meet the Gaussian distribution and are continuous [21].

### 2.6.7 Testing methods

The test methods include *k*-fold cross-validation and hold-out validation.

- The ***k*-fold cross-validation** works on the principle of taking a dataset and splitting it into *k* subsets – folds. Then, one fold is selected as a test subset from these randomly created folds, others as train subsets. After training the model and evaluating a test subset, the test subset becomes training one, and another different for testing is picked from rest folds. Exchanging of folds repeats until every combination is attempted [4]. Figure 2.9 describes how were individual subsets of shuffling for testing.

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 1
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 2
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 3
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 4
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Metric 5

Figure 2.9: Example of 5-fold cross-validation process from [4].

- The **hold-out validation** was used by splitting the dataset into two parts: training and testing. Next, the model needs to be trained on the training data for evaluating the testing data. Usually letting one-fifth of the data for testing because the model is unfamiliar with them yet, so evaluation shows its performance. [4]

# Chapter 3

## Data

This chapter describes the data on which this work was built. In the first section 3.1 it is shown what kind of data it is and which dataset it comes from. The second section 3.2 shows the work with the data that is necessary before connecting to other parts of the solution. Cleaning and preparing the data so that the results were not affected by noise.

### 3.1 Dataset

The data used for emotion prediction are from a dataset [23] called DREAMER. This dataset consists of 23 subjects. Each subject has two different types of records, baseline and stimuli. Stimuli records are the last 60 seconds of 18 different movie clips that were indicating different emotions. For clearing current emotional state subjects also watched one neutral video before each movie clip [23].

#### 3.1.1 Channels

For recording the dataset was used the Emotiv EPOC wireless EEG headset [23]. This headset includes 16 contact-sensors to record EEG signals at 128 Hz sampling rate. Electrodes are placed in locations following the International 10–20 system standard 3.1.

14 out of 16 contact-sensors represent 14 channels [15]:

- AF3, AF4, F7, F8, F3, F4, FC5, FC6, T7, T8, P7, P8, O1, O2

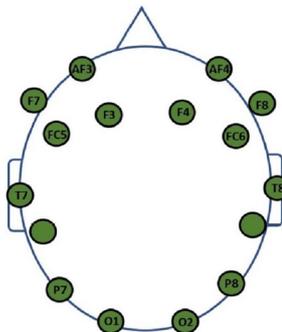


Figure 3.1: Placement of channels from [15].

### 3.1.2 Movie Clips

The selection of movie clips was focused to induce in subjects these nine emotions: [23]

- amusement, excitement, happiness, calmness, anger, disgust, fear, sadness and surprise [23]

Each of these emotions has two movie clips that represent it. The show time of every clip was between 65 and 393 seconds, but recorded was only the last 60 s of the video to prevent mixing different kinds of emotions during measuring. Table 3.1 shows list of movie clips with their target emotion and average valence and arousal score [23].

	Movie clip	Target emotion	Valence	Arousal
1	Searching for Bobby Fischer	calmness	$3,17 \pm 0,72$	$2,26 \pm 0,75$
2	D.O.A	surprise	$3,04 \pm 0,88$	$3,00 \pm 1,00$
3	the Hangover	amusement	$4,57 \pm 0,73$	$3,83 \pm 0,83$
4	the Ring	fear	$2,04 \pm 1,02$	$4,26 \pm 0,69$
5	300	excitement	$3,22 \pm 1,17$	$3,70 \pm 0,70$
6	National Lampoon's VanWilder	disgust	$2,70 \pm 1,55$	$3,83 \pm 0,83$
7	Wall-E	happiness	$4,52 \pm 0,59$	$3,17 \pm 0,98$
8	Crash	anger	$1,35 \pm 0,65$	$3,96 \pm 0,77$
9	My Girl	sadness	$1,39 \pm 0,66$	$3,00 \pm 1,09$
10	the Fly	disgust	$2,17 \pm 1,15$	$3,30 \pm 1,02$
11	Pride and Prejudice	calmness	$3,96 \pm 0,64$	$1,96 \pm 0,82$
12	Modern Times	amusement	$3,96 \pm 0,56$	$2,61 \pm 0,89$
13	Remember the Titans	happiness	$4,39 \pm 0,66$	$3,70 \pm 0,97$
14	Gentlemans Agreement	anger	$2,35 \pm 0,65$	$2,22 \pm 0,85$
15	Psycho	fear	$2,48 \pm 0,85$	$3,09 \pm 1,00$
16	the Bourne Identity	excitement	$3,65 \pm 0,65$	$3,35 \pm 1,07$
17	the Shawshank Redemption	sadness	$1,52 \pm 0,59$	$3,00 \pm 0,74$
18	the Departed	surprise	$2,65 \pm 0,78$	$3,91 \pm 0,85$

Table 3.1: List of movie clips from [23].

### 3.1.3 Evaluation

At the begging of the measurement, the concepts of the VAD model were explained to all participants. The VAD model consists of emotions: valence, arousal and dominance and these three emotions are rated for each movie clip. Per the video, all participants were asked to self-evaluate their feelings. They used a number between 1 and 5 (1 was the least, and 5 was the highest score) and rated according to the VAD model [23].

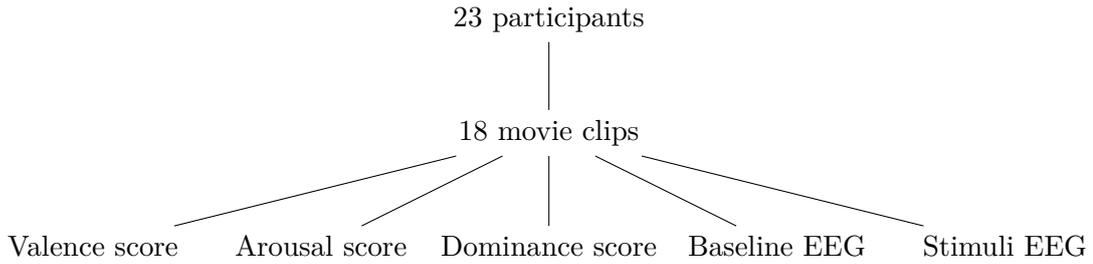


Figure 3.2: Structure of subjects and movie clips.

I created a tree structure 3.2 according to the dataset DREAMER to simplify the data and node relations. From these three emotions scores, my task was to choose two. I selected valence and arousal because they create a clearer combination to represent emotions on a valence-arousal map 4.7 than combined with dominance.

## 3.2 Signal-preprocessing

From the original signal, which was full of noise and unwanted segments, it was necessary to create a new one, more efficient for further data analysis. The first step was defining high-pass and low-pass filters according to the smallest and largest possible frequency that carries pieces of information. The high-pass filter was set at 0.5 Hz because of the delta's wave lower limit. The low-pass filter is set at 40 Hz.

### 3.2.1 Cleaning Artifacts

I used ICA for cleaning artifacts that the band-pass filter could not remove. If other diverse signals in the original signal were presented, then the ICA tool could find them and separate these noise signals from the rest of the usable data [15].

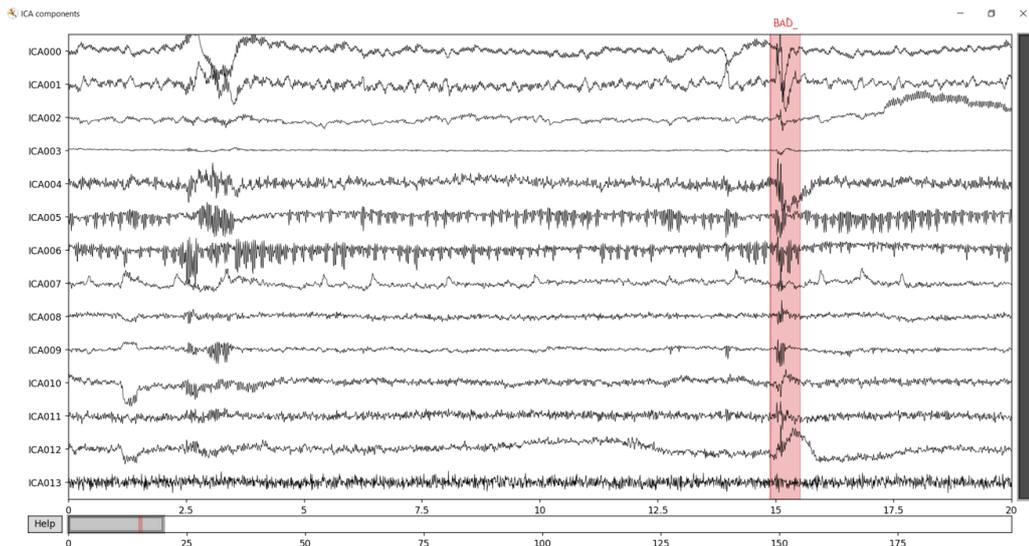


Figure 3.3: Uncleaned data in the python tool ICA.

The data from the dataset DREAMER contained much noise and disturbing parts—for example figure 3.3. some data could not be properly cleaned by ICA, so I manually cleaned all other presented artifacts. A red highlighted part marked as BAD\_ could be caused by an eye blink because the dislocation was presented simultaneously in all channels. These highlighted parts were then removed.

According to [23], in the beginning, the number of participants was 25 (the final number was 23). However, due to technical problems during the recording, they had to eliminate two records from the final dataset. Their comment about the technical problem was a reason why these problems could affect individual channels in other participants' measurements. According to this, I assumed that visible noise in channels ICA005, ICA006 and ICA013 was caused by incorrectly placing the cap with electrodes on the head of participants or by loose wire connections.

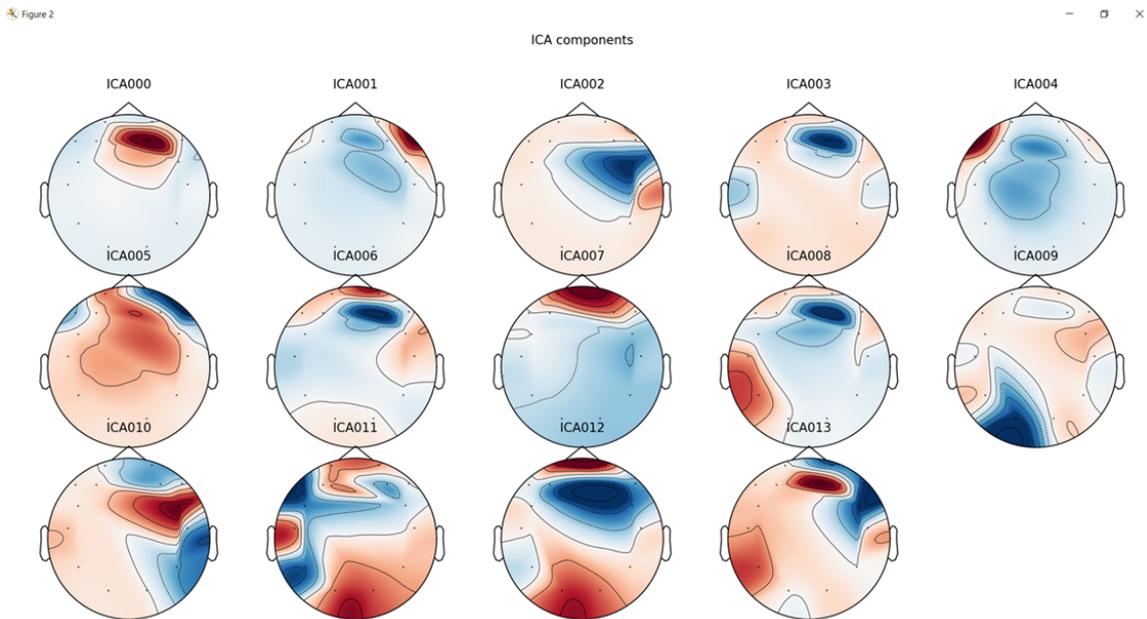


Figure 3.4: Marked activity on the human brain for examination of EEG signal.

Figure 3.4 shows a centre of brain activity for each electrode. For example, a channel on the head marked as ICA007 contained significant activity around the eye area, and in figure 3.4, the same channel has periodical spikes. These spikes could be lateral eye movements following the location on the head. This EEG montage was used for manual cleaning control.

### 3.2.2 Comparison of Clean and Not Clean Data

Pre-processed data was compared with raw data, whose only modification was applying a band-pass filter and setting the standard 1020 montage. No sections were marked as BAD\_ and removed for further use. Here is a comparison between these two different data with applied skewness and peak-to-peak functions.

The peak-to-peak function indicates a signal disparity between its lowest and highest points [54]. Artifacts on the signal create big spikes caused by blinking, for example. Furthermore, these spikes are so significant that they produce abnormalities in extracted

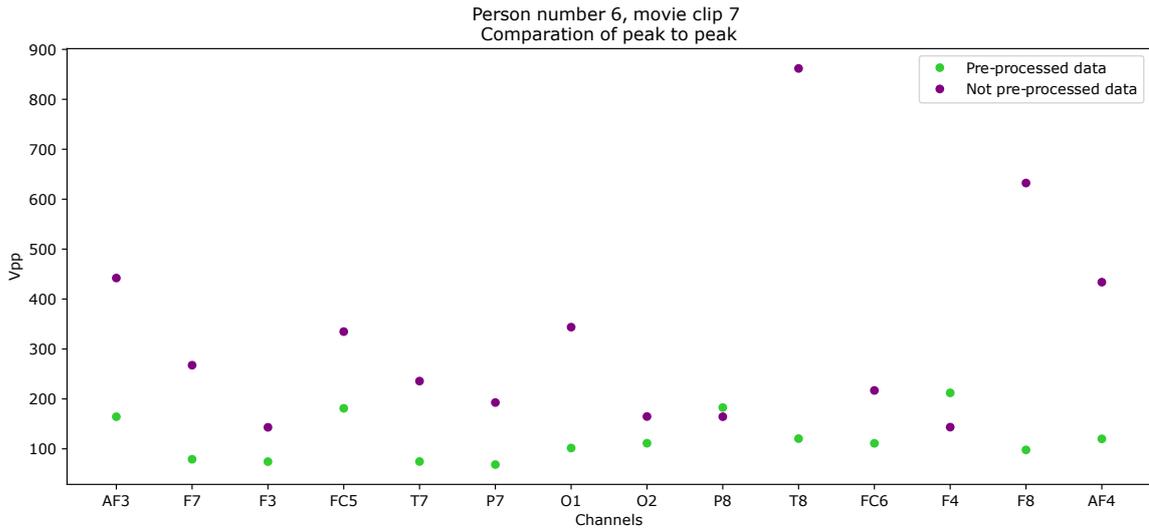


Figure 3.5: Comparison between cleaned and uncleaned signal with calculated peak-to-peak values of each channel.

features. This phenomenon can be seen in the figure 3.5, showing the big difference between the data with removed artifacts (pre-processed) and data that includes artifacts (not pre-processed). It is clear from figure 3.5 that channel T8 is most affected by oscillations.

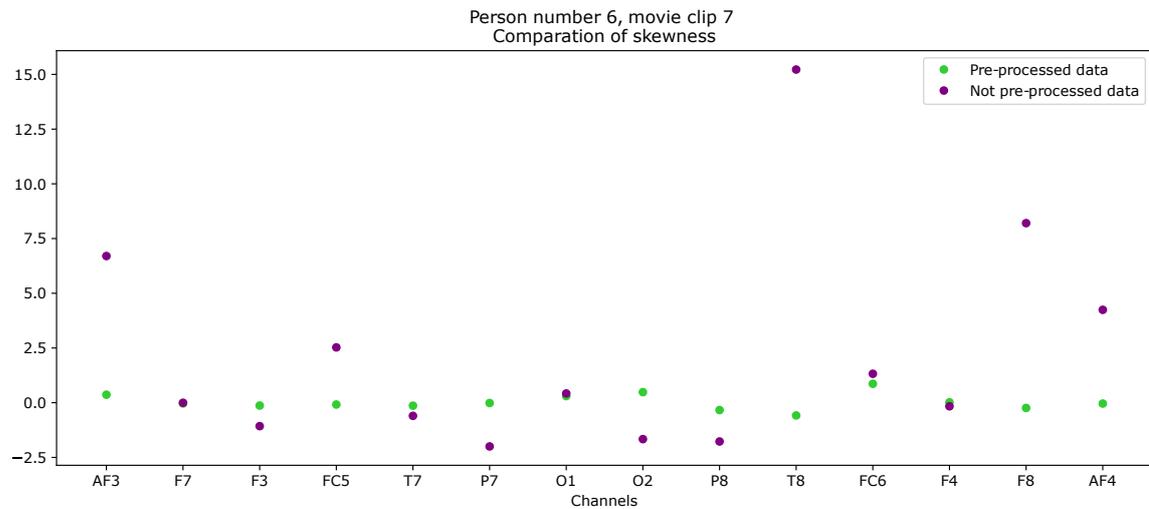


Figure 3.6: Comparison between cleaned and uncleaned signal with calculated skewness values of each channel.

From figure 3.6 picturing skewness of the EEG signal is clear that T8 channel was indeed the most affected one by the noise. However, after noise extraction, signals of every channel were in a norm. This proved that preprocessing was successful because the signal showed no noticeable signs of noise.

At this point, all 18 cleaned movie clips were ready to move on to the next step in chapter 4.

# Chapter 4

## Proposed solution

This chapter describes my solution for feature extraction and selection for emotion detection. Due to the fact that this work focuses on experiments, part of which is an uncleaned dataset DREAMER, where the number of artifacts was still presented, its solution will be not only extraction and selection of features but also other elements related to the emotion detection process. The process of the entire work is based on figure 4.1.

A felicitous representation of the whole process is figure 4.1 of a pipeline because the process, including multiple algorithms, is complex and includes several steps. For my method, I followed [38] to split the solution into four subsolutions. These four subsolutions, with a square that symbolizes the result, are marked in uppercase in the pipeline 4.1.

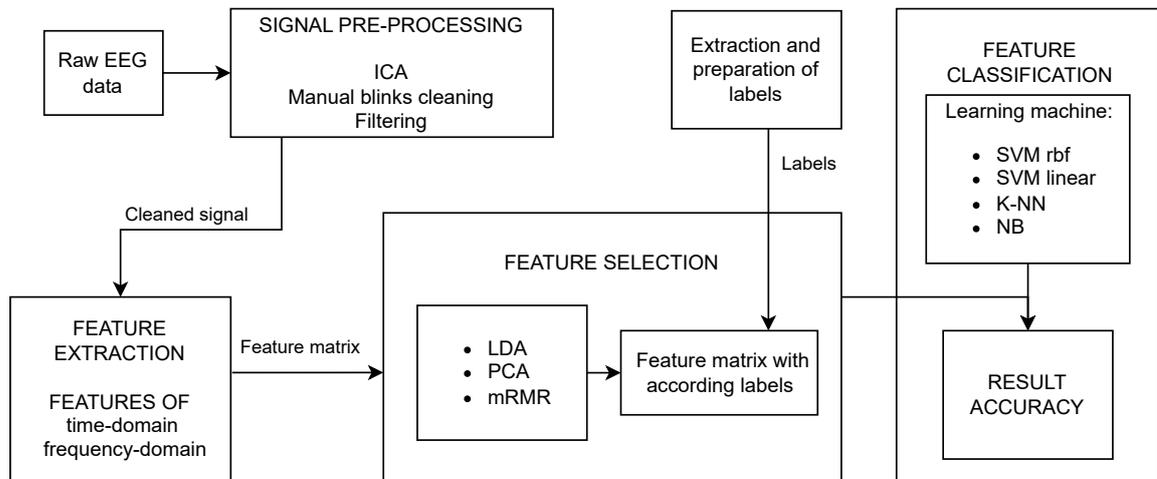


Figure 4.1: The basic pipeline of the emotion recognition system through EEG.

The processing section 3.2 was focused on editing and cleaning the original signals from the DREAMER set for the prospect of using them in the next steps to the solution. The result was a cleaned signal that was ready for feature extraction. During this extraction 4.1, features for both its domains and entropies were calculated. After calculating these properties, they moved to the selection phase 4.2. The selection section consists of the first step of choosing the number of classes for labels and the second step of choosing the learning classifier model that was performing the selection. The result was a matrix that contained only relevant information. In last section 4.3.2, this matrix was imported together

with the classes into the model, which verified the accuracy and correctness of all three previous steps.

## 4.1 Feature Extraction

This section describes work on signal utilization for transformation to a feature matrix. The feature matrix consisted of vectors that contained all possible combinations of calculated properties from frequency bands and channels. The tree structure 4.2 represents feature extraction from the data. Each record with a duration of 60s contained the data for 14 channels. A signal from each channel is subdivided into four brain parts and then into five waves depending on their frequencies.

Every wave has its own time-domain and frequency-domain features. Before extraction, the signal was downsampled from sampling rate of 128 Hz to sampling rate of 64 Hz according to the Nyquist-Shannon sampling theorem [41].

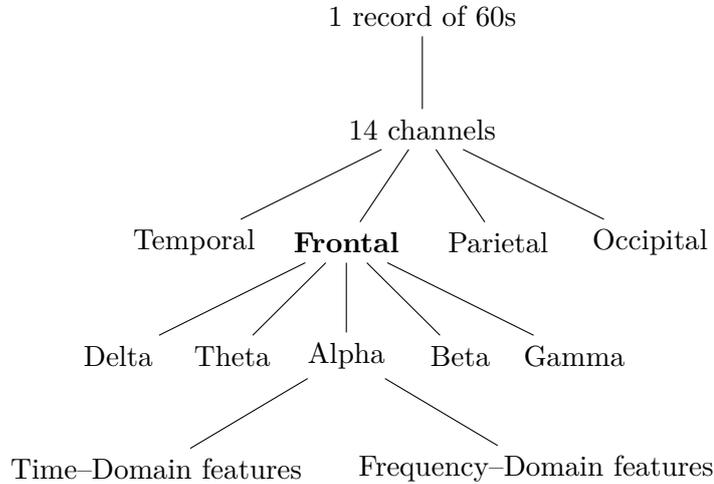


Figure 4.2: Structure of extracted features. For simplicity, the feature nodes come out only from the alpha frequency band, but it applies to all other frequency bands.

### 4.1.1 Preparation for Feature Extraction

This subsection describes the preparation of the data. Previously was mentioned in chapter 2 that the human brain is built from brain lobes. Each brain lobe has a different predisposition for accomplishing diverse tasks. Therefore, I use a channel layout according to the placement of their relevant electrodes.

- Frontal: F3, F4, AF3, AF4, F8, F7, FC5, FC6
- Temporal: T7, T8
- Parietal: P7, P8
- Occipital: O1, O2

The number of features, 1 460, was extracted from the parietal, occipital and temporal lobes, because each contains 2 channels. From the frontal lobe it was 5 840, because it includes 8 channels. A total number of features equals  $14 \times 5 \times (17 + 129) = 10\ 220$ .

### 4.1.2 Features

In [24] and [47] were mentioned time-domain, frequency domain and entropy features that I used during feature extraction. List of features:

#### Time-domain:

- Statistical features: mean, median, variance, Standard deviation (STD), kurtosis, minimum, maximum, Root Mean Square (RMS), skewness, energy, Peak to peak (PTP)
- Hjorth parameters – activity, mobility, complexity
- Entropies: Permutation entropy, Sample entropy and Approximate entropy

#### Frequency-domain:

- Power spectral density

Extracted features in feature matrix were ready for feature selection.

## 4.2 Feature Selection

Feature selection removes attributes when it is established on specific requirements. Therefore, this method helps decrease the number of features needed for the following manipulation of the data [2].

Feature selection section shows in what form it is possible to select only the most essential data from the overall feature matrix. Labels and classes 4.2.1 had to be selected before choosing a selection method. After that, three methods were chosen to select data on which the experiments were carried out. One of the parameters necessary for the use of any method is the number 4.2.2 of different data to keep in order to achieve the highest possible accuracy and relevance.

### 4.2.1 Classification of Target Label

Participants in the dataset DREAMER self-evaluated how they felt during the experiment. The participant’s self-evaluation and their features should both constitute their predicted emotion. Therefore I used the self-evaluation as a target variable and proposed three new ranking systems.

The ranking system used in dataset DREAMER is between 1 and 5 (1 is the lowest, and 5 is the highest number [23]).

### Decomposition of Label Systems

The first ranking system was divided into quadrants – figure 4.3 (c) and named as  $Q$  for simplification. The second was partitioned into halves – figure 4.3 (a,b) and named  $VA$ , where for both valence and arousal, ranking was replaced with a binary score, 1 for high and 0 for low, following these rules:

$$\left\| \begin{array}{l} V \leq 3 = 0 \quad \dots \quad LV - \text{low valence} \\ A \leq 3 = 0 \quad \dots \quad LA - \text{low arousal} \\ V > 3 = 1 \quad \dots \quad HV - \text{high valence} \\ A > 3 = 1 \quad \dots \quad HA - \text{high arousal} \end{array} \right\|$$

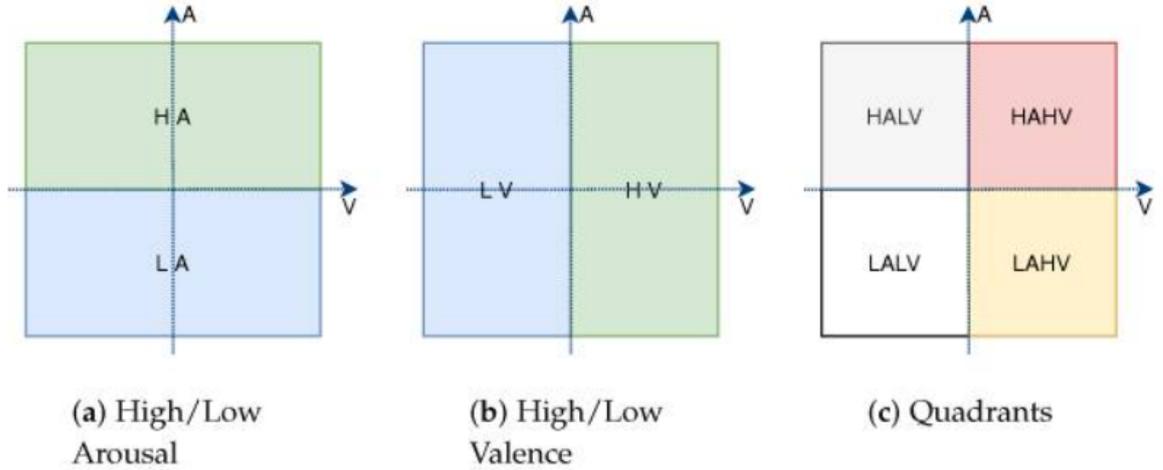


Figure 4.3: Figures of different classification systems from [13].

For example, scores from movie clip number 7 were at first converted into binary raking for valence and arousal separately ( $VA$ ). Then suitably into the quadratic ( $Q$ ) raking system. Here are the results from clip number 7 (each of the 23 subjects has assigned one value):

$$\begin{aligned}
 V: & 1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 \\
 A: & 1,0,0,0,0,0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,0,0,0 \\
 Q: & 3,1,1,1,1,1,3,0,3,3,1,3,1,3,1,3,1,1,3,1,1,1,1
 \end{aligned}$$

The strong correlation is between  $A$  and  $Q$  because, with one exception, it can be seen that when  $A = 0$ , then  $Q = 1$  and when  $A = 1$ , then  $Q = 3$ . This indicates a situation in which only one decisive value (in this case, an emotion) determines the result. On the other hand, if the decisive value, here valence, is almost the same for every subject, then it is not genuinely effective because there is no variation in the data.

I solved this problem by mixing test data from different quadrants to balance the high-/low valence and high/low arousal ratios. Table 4.1 simplifies how labels were edited for the feature selection ranking method  $VA$  and  $Q$ .

Shortcut	Description	Target Emotions	A	V	Q
HAHV	high arousal high valence	surprise, happiness [13]	1	1	3
HALV	high arousal low valence	anger, disgust, fear [13]	1	0	2
LAHV	low arousal high valence	neutral [13]	0	1	1
LALV	low arousal low valence	sadness [13]	0	0	0

Table 4.1: Two proposed ranking methods  $VA$  and  $Q$  for feature selection.

Due to the participants' self-evaluation cannot be divided straight into two groups, because the number of labels is odd, there is also the third option 4.4 to add one neutral label. Rules for creating this third ranking system named  $N$ :

$$\left\| \begin{array}{l}
 V \leq 3 \text{ or } A \leq 2 = -1 \dots L - \text{low valence/arousal} \\
 V = 3 \text{ or } A = 3 = 0 \dots N - \text{neutral valence/arousal} \\
 V \geq 4 \text{ or } A \geq 4 = 1 \dots H - \text{high valence/arousal}
 \end{array} \right\|$$

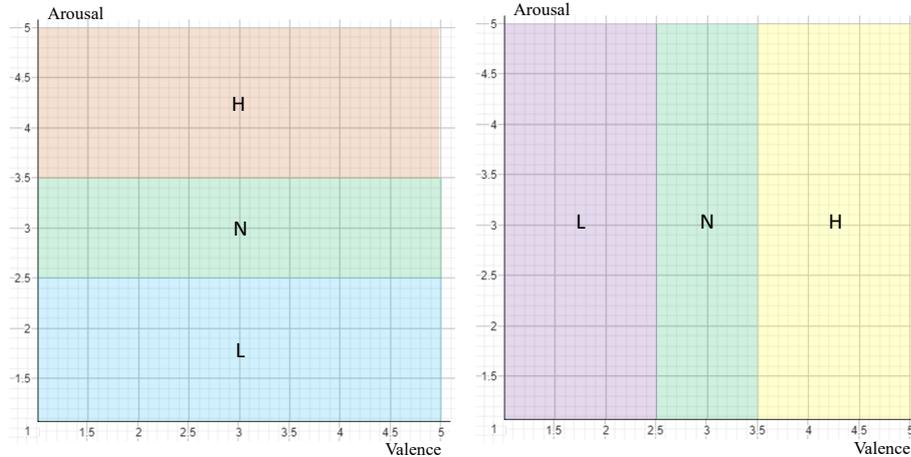


Figure 4.4: Ranking system  $N$  layout of the third three classes option.

#### 4.2.2 Feature Selection Methods

The selection was affected by three different methods to determine the most effective one which will form a part of the resulting model design. I used these methods:

- **Maximum Relevance–Minimum Redundancy with Pearson correlation coefficient**
- **Principal Component Analysis**
- **Linear Discriminant Analysis**

For feature selection is essential to choose a number value of features to prevent overfitting or underfitting in the classification step. I followed this guide [37].

- LDA, as the only method of the three, it accepts as a number to keep for which: number of features  $<$  number of classes  $-1$ . For this reason, I chose one feature for labels containing two and three classes. The number of features for labels containing four classes was set at two.
- PCA and mRMR had no such limitation, so to determine the number of features for mRMR and PCA, I used PCA with 99% variance shown in figure 4.5. Each channel had its own number of features to keep, because for every channel the calculated number from PCA with 99% variance differs. For the figure 4.5 representing selected number of features from one random channel I used this guide [37]

#### Principal Component Analysis and Linear Discriminant Analysis

The first step of using these methods was to fit the feature matrix. Then, after an acceptance of input as the feature matrix, both methods transformed the matrix into a lesser dimension matrix. This new matrix was further used as an input for the learning classifier to evaluate.

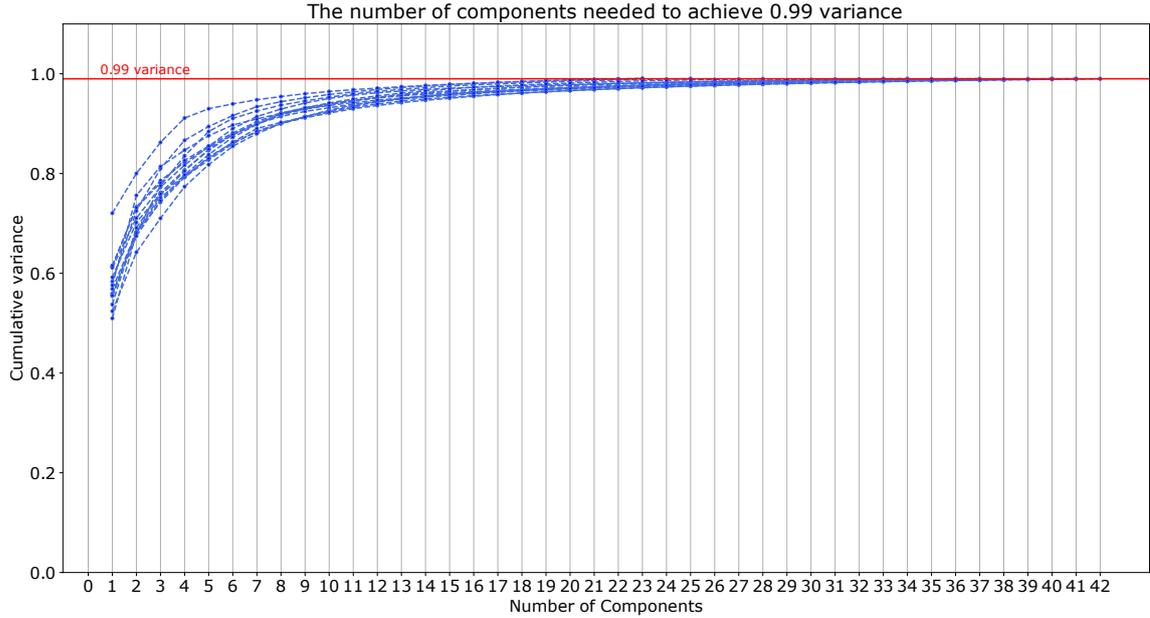


Figure 4.5: Figure shows numbers of features needed to achieve 99% variance. Each number is calculated from one channel with PCA.

### MRMR Method with Person's Correlation Coefficient

According to a high number of features extracted from preprocessed data, I decided to apply the mRMR selection method, which focuses on minimum redundancy in the first place. One of its parameters is the number of features to keep. This number is calculated from the previous step using PCA.

In case, some features were still correlated with each other even after, and the number of features was large. For this reason, I tried to add Pearson's correlation to prevent inaccuracies during classification that could occur. However, the results 5.1 after the addition did not affect the results significantly. Therefore, I decided not to add Person's correlation to other selection methods as well.

In this article [49] is mentioned that when Pearson's correlation coefficient marked as  $r$  applies is  $0.7 \leq |r| < 1.0$ , then it is considered as a strong correlation between features. To increase accuracy, I removed features with 0.7 or higher Pearson correlation coefficient.

The heatmap 4.6 displaying the heatmap indicates the value of Pearson's correlation coefficient as a colour, where the red tone symbolizes a positive correlation and the blue tone a negative correlation [49]. A diagonal is coloured red because, at that point, there is a conjunction of similar features.

Features were marked with numbers that agree with their order in the features matrix, because only their correlations with each other are influential. A large part of the features from the figure have values in the extreme limits above 0.7, which constitute redundancy. Therefore final feature matrix was reduced and only features within the limits were included.

I used this guide [49] to apply Person's correlation coefficient and heatmap.

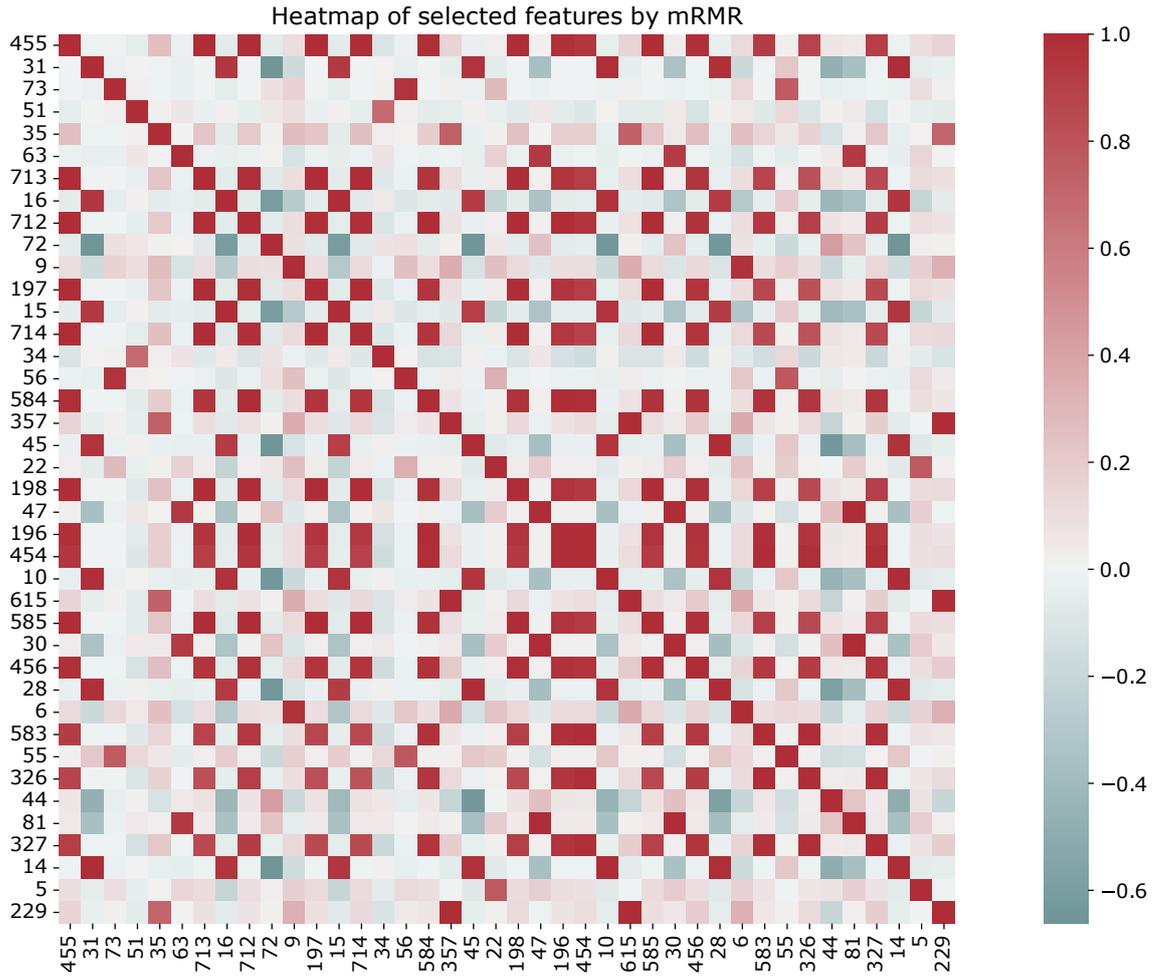


Figure 4.6: Display of Pearson's correlation coefficient in the heatmap.

### 4.3 Classification of Emotions

This section describes the usage of the previous feature selection for emotion recognition and its evaluation. The classifier learning classifier is needed to evaluate whether feature extraction and selection succeeded. Emotion recognition needed a learning classifier algorithm to achieve the final stage at the pipeline shown 4.1.

Accuracy of feature extraction and selection as an essential output of a classifier learning classifier. Before an evaluation I normalized the data. According to this [25], support vector machine expects that the range of the input data is between 0 to 1 or -1 to 1. Therefore, this SVM's condition leads to altering the data to meet the conditions. Two types of normalizations modify the data into the required type.

- *Hard* normalization, during which calculates a min and max value.
- *Soft* normalization works with a mean and standard deviation of one data dimension.

Even if *soft* normalization is recommended, I tried both methods for selection and classification process.

### 4.3.1 Valence and Arousal Categorization

The dataset was recorded while all participants watched a video focused on eliciting one emotion. Therefore, the feature classification process needed to work with particular attributes to predict an emotion, and these attributes required a specific pair of valence and arousal values for each emotion. This attribute was a coordinate  $[x; y]$ , where  $x = \text{valence}$ ,  $y = \text{arousal}$ .

To express every emotion in the coordinate system, I used mean values calculated from all participants' responses [23]. In addition, every movie-targeted emotion was represented twice, so I calculated a mean from two accorded values. A result was  $[x; y]$  coordinates shown in figures 4.7.

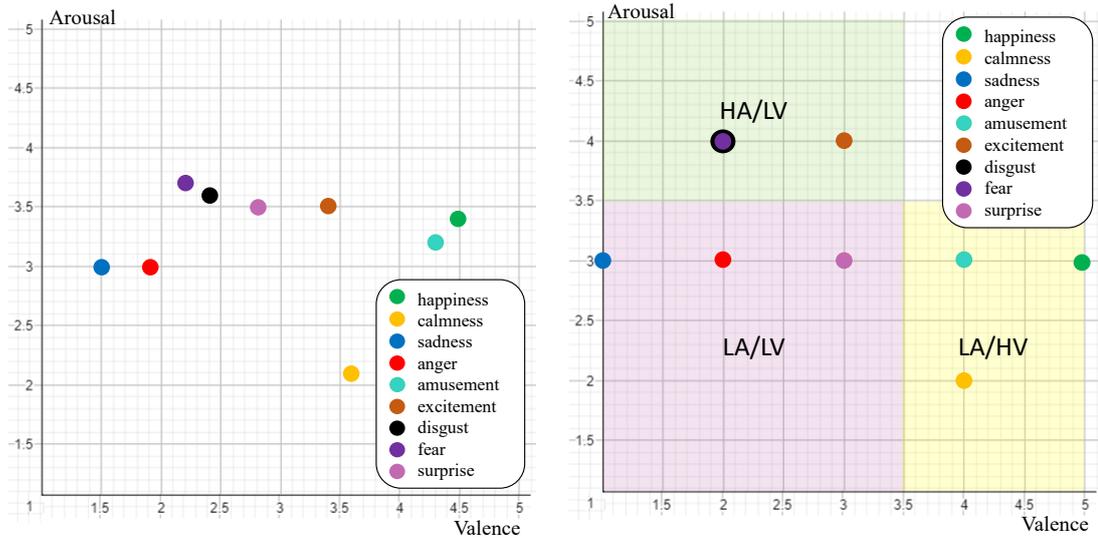


Figure 4.7: The left figure illustrates the mean values of emotions rounded to a decimal place, and the right figure represents the values rounded to whole units.

Disgust and fear are especially close located, unlike the other emotions. Their disparity would be distinguished by adding one more dimension to achieve a 3D view. The  $Z$  axis would represent dominance as the third emotion.

A way of verification of feature extraction and selection method was a distinction of these emotions in an area divided into parts with high/low valence and arousal in the figure 4.7.

The feature classification in 4.3.2 was designed so that it was able to recognize four different emotions according to valence and arousal level. Right figure 4.7 shows the distribution of emotions into these four parts for  $VA$  and  $Q$  labelling methods.

### 4.3.2 Data structure of Classifier Input

Both emotions, valence and arousal, classify in a range of 1 to 5, which means  $5 \times 2$  label options. Therefore, I decided to use the same transformation to a new labelling system from 4.2 to enable application valence and arousal into the classifier learning classifier. The same labelling system used during feature selection was also used for feature classification.

- The first labelling method required two support vector models. The high/low arousal model with  $1A$  output was trained for arousal detection, and the high/low valence

model was trained for valence detection. They both together can recognize a respective quadrant, where a sample belongs. Both  $1A$  and  $1V$  output is binary, 0 for low and 1 for high values.

- The second labelling method resembled the first one but contained an extra neutral class. The differences in the process are displayed on the pipeline 4.8 and 4.9.

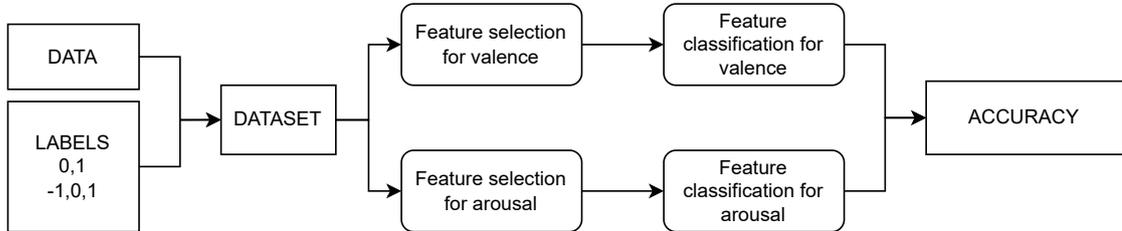


Figure 4.8: Pipeline of the steps for using two learning classifiers.

- The third labelling method was significantly different from its predecessors, as it only included one learning classifier to predict which quadrant an emotion belonged to. Figure 4.9 shows this case.

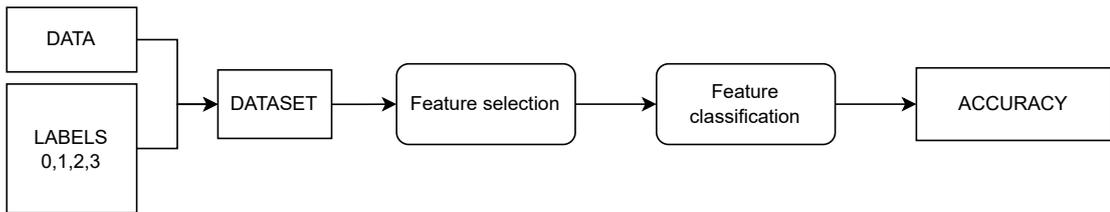


Figure 4.9: Pipeline of the steps for using one learning classifier.

### 4.3.3 Classification model

In the case that one of the methods would be more suitable for only one kind of learning classifier, I used three different ones.

- **SVM with *RBF* kernel**
- **SVM with *linear* kernel**
- ***K*-Nearest Neighbors**
- **Gaussian Naive Bayes**

The classifier learning models were trained with the dataset divided into two parts with an 80:20 (train:test) ratio with their accorded labels. If the model was made for predicting the valence level, it was trained with train valence labels. This training process was tested with the  $k$ -fold cross-validation method with 20% of the dataset for a final check if the model worked adequately.

Splitting channels into four groups provides information about which brain lobe is the most informative during emotion detection [6]. The test was carried out by using only features from one certain channel. Then, feature selection was applied to this channel, and features were fitted into the learning classifier.

# Chapter 5

## Validation and Evaluation

This chapter explains the testing progress and how it influenced the results. During testing, I reached a situation where creating many combinations was possible. I included all these combinations in the solution because the results reached low accuracy values in my initial steps of the solution. Therefore, more options were created to find a proposal solution for feature extraction and selection methods.

labels  $\times$  feature selection methods  $\times$  parameters  $\times$  learning classifiers

Subsections 5.1.1, 5.1.2 and 5.1.3 were incorporated into the entire process, and therefore their results were also the final results. Due to the large number of different combinations that could be included in testing, I first limited the selection of parameters for learning classifiers 5.1.4. Then I narrowed down the choice of ranking method and its number of classes 5.2.1. Finally, selection methods that did not achieve the same results as others were discarded. This step was created by previewing all channels separately in 5.2.2.

### 5.1 Validations

I followed this guide [31] to select and implement validation methods.

Before using one specific classification model, I tested some possible useable models. The main focus of the results was the accuracy of predicted samples compared with target labels. If accuracy exceeded 80%, it was considered a successful result. The second objective was to check if the model's training functioned correctly because the most significant issue was that the final classification came out only one-dimensional. One-dimensional prediction output included only one label class, so instead of proper prediction, the output was composed, for example, only from zeros or number one. This behaviour was mainly caused by poorly selected models' parameters. For this reason, I also added tests for various combinations of parameters, in particular, SVM's *gamma* and *C* parameters and for KNN number of neighbors parameter.

#### 5.1.1 Hold-out Cross-Validation

The hold-out cross-validation method was used in this test process. First, the data was divided into two parts, one train set with 80% of the data and one test set with the rest 20%. Second, every classification model was trained with the same training data and tested with the same test data. Third, the results were evaluated, and the accuracy of these results

was calculated. Lastly, all models were compared based on accuracy, and the best model was selected.

### 5.1.2 *K*-Fold Cross Validation

After selecting one classification model for implementing emotion prediction, I used the *k*-Fold Cross Validation method for testing every learning classifier to achieve the best results. This method verified that the model's training ended successfully. Overfitting the model is possible in a negative scenario, so the method offers verification to prevent it.

### 5.1.3 Proper Functionality of Learning Classifiers

Within involving more methods that should enable better results to be achieved, learning classifiers were often changing their behaviour. As an unwanted behaviour could be considered a state when the learning classifier evaluated all data with only one class from offered classes. If the learning classifier evaluates in this form, proper results could not be achieved. This case shows an example:

Learning classifier result: 1,1  
Labels corresponding to input data: 1,1,1,1,1,1,0,0,1,0,1,0,1,1,1,1,0,1,1,1,1,0

From my observation of this undesirable condition, I could say this happened in the circumstances, when:

- Normalization of the data was erroneously used. The most common was when *hard* normalization was presented and implemented according to [25]. This is the reason I further used only *soft* normalization as the only one in implementation recommended by [25].
- Labels were in an uneven state. The ratio of different classes in the label array did not get close to an ideal state of 1:1 for the two classes, but for example, it contained from 80% only class marked as 1. When this happened, the learning classifier used a value of the class, which was the most presented, to evaluate the entire test data. I solved this problem by using all stimuli signal data from the dataset to create more proportionally evaluated data.
- The data was unevenly mixed into train and test data. Even if the ratio of classes was balanced after accidental integrating data could happen, the labels, mainly in the test subset (because the test subset was smaller than the train subset), were not in symmetry during hold-out cross-validation. I solved this problem by adding 5-Fold cross-validation. Even if some test subsets were not labelled evenly, the rest provided proper results.

### 5.1.4 Parameters of Learning Classifiers

Learning classifier SVM with *RBF* kernel took two parameters, *gamma* and *C*, which could affect a result. During accuracy testing, I added tests for using the best *gamma* and *C* combination with the highest accuracy. SVM with *linear* kernel had a parameter *C* and *k*-nearest neighbors had a parameter, that depicts number of neighbors.

These three selection methods were included in this test, and the results are general for all usages of these methods. At the same time, all the data from the matrix of features

were included because it was necessary to choose some input value before the other tests, thus avoiding a large number of different possible combinations.

	SVM, <i>RBF</i> - $C$	SVM, <i>RBF</i> - $\gamma$	SVM, <i>linear</i> - $C$	$K$ -NN - $n$ neighbor
$VA$	10.0	0.001	0.001	2
$N$	10.0	0.001	0.001	1
$Q$	100.0	0.001	0.001	3

Table 5.1: The best possible accuracy achieved with parameters for selection methods.

It can be seen from table 5.1 that the most common calculated value for SVM *RBF*  $\gamma$  is 0.001 and for  $C$  is 10.0. SVM with *linear* kernel ended up with 0.001 value for its parameter  $C$ .  $K$ -nearest neighbors's parameter number of neighbor had best accuracy for values 1, 2, 3, so I selected a mean of these and for all tests was used the number 2 as the  $n$  neighbor parameter.

## 5.2 Evaluation

After combining all the previous parts, evaluating the assembled model with one of the four selected learning classifiers was necessary. This section shows several experiments and tests performed on the data to reduce the parts of the solution, the results of which did not reach such accuracies as the others. The results of these experiments were presented in section Proposed Solution 5.3.

### 5.2.1 Testing Different Labelling Methods

In this test, only the parameters reached in 5.1.4 were used for learning classifiers. Before performing further tests, I selected the label method to achieve the best possible test result. In table 5.2.1 are the accuracies of feature selection with PCA, LDA and mRMR with Pearson correlation coefficient. The three possible label methods are:

- $VA$  – low/high valence and low/high arousal – two classes
- $N$  – low/neutral/high valence and low/neutral/high arousal – three classes
- $Q$  – low/high valence plus low/high arousal – four classes

A table was created for values achieved during one experiment run. These values were used to plot figure 5.1. Table 5.2.1 shows how the values of individual rows were always descending for every feature selection method in combination with a learning classifier.

Two learning classifiers were implemented for  $VA$  and  $N$  because the first was needed for valence application and the second for arousal classification. The input of labels for the feature selection algorithm was the same as for the learning classifier. It is clear from figure 5.1 that the best percentage is always in the pink bar which depicts  $VA$ . The second best results are shown in the second green bar  $N$  and the last quadratic resolution ended as the worst technique.

It is clear that the best percentage is in the first line, which corresponds to  $VA$ . The second best results are in the second line  $N$  and the quadratic resolution ended as the worst technique. The main result which can be deducted from 5.2 is that the more classes labels

	mRMR				PCA				LDA			
	A	B	C	D	A	B	C	D	A	B	C	D
VA	0.60	0.58	0.57	0.53	0.58	0.58	0.57	0.48	0.92	0.92	0.91	<b>0.93</b>
N	0.41	0.40	0.36	0.33	0.40	0.41	0.36	0.33	0.79	0.66	0.81	<b>0.82</b>
Q	0.36	0.35	0.32	0.25	0.34	0.35	0.31	0.23	0.84	0.68	0.83	<b>0.86</b>

Table 5.2: Accuracy results of testing different labelling techniques, where A = SVM with *RBF* kernel, B = SVM with *linear* kernel, C =  $k$ -nearest neighbors, D = naive Bayes.

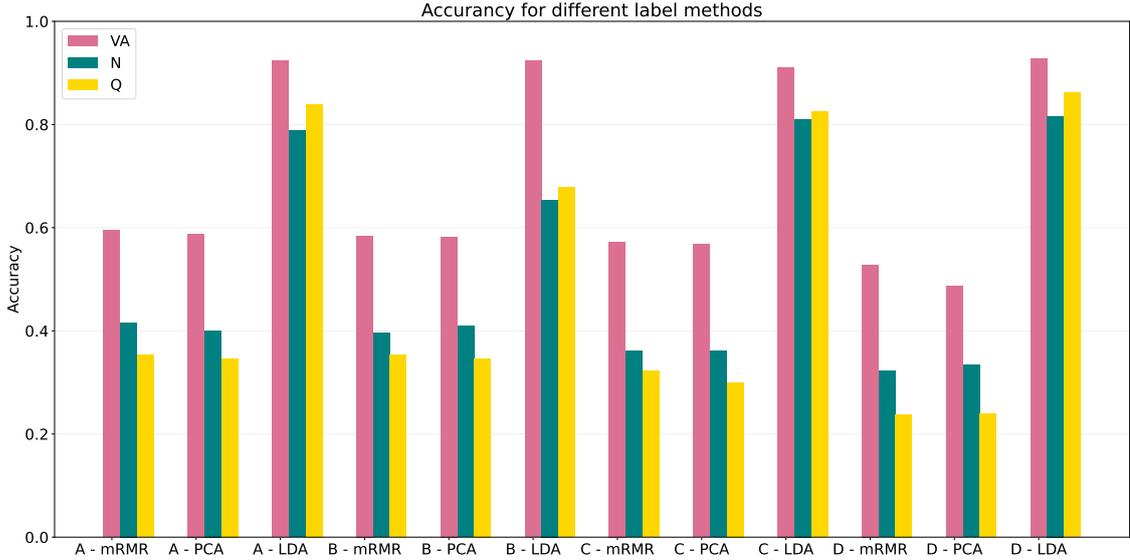


Figure 5.1: Label classes and their accuracy, where A = SVM with *RBF* kernel, B = SVM with *linear* kernel, C =  $k$ -nearest neighbors, D = Naive Bayes.

contained, the lower the score went down. The inverse proportion is between the number of classes and the accuracy value.

At this point, I limited the number of classes for representing valence and arousal to just 2, low and high, which are shown here as VA. Another symptom that can be deduced here is that, in general, the best results went to the LDA method with a combination of naive Bayes. In subsequent tests and experiments in 5.2.2, it will be verified if the LDA method appears to be the best.

## 5.2.2 Accuracy of Different Channels

In this study [6], the authors calculated the accuracy for every channel separately. This finding helped determine the most informative EEG channel and then also the brain region. I did the same thing to create my proposed feature extraction and selection solution. As a test, I used three feature selection methods to select the final one with the best results. As with 5.1, the LDA method came out with the best values, which is why I chose it in the final solution of my work.

From the accuracy results of each channel 5.2, I split channels into four groups according to their position on a head described in 4.1.1. For each of these groups average accuracy

value was calculated. A comparison of the importance of brain regions for emotion detection is shown in figure 5.3.

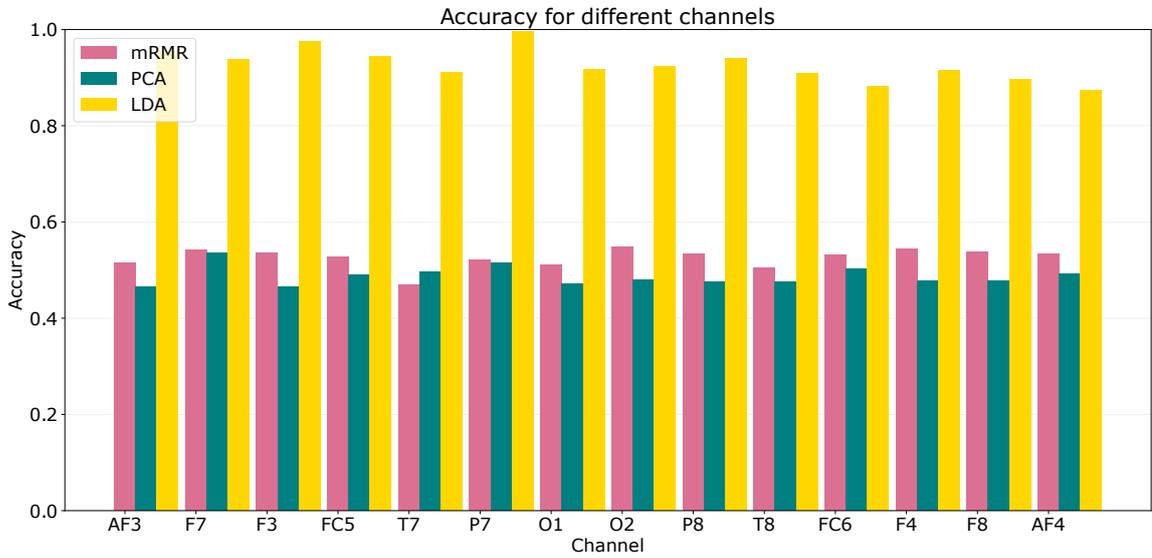


Figure 5.2: Channels and their accuracy.

Figure 5.3 showing accuracy for different lobes shows that in general, the frontal lobe had the greatest accuracy in emotion prediction for all methods. This result proved that the frontal lobe is for my experiments the most important aspect in predicting emotions. For this reason, further experiments were performed on the frontal lobe.

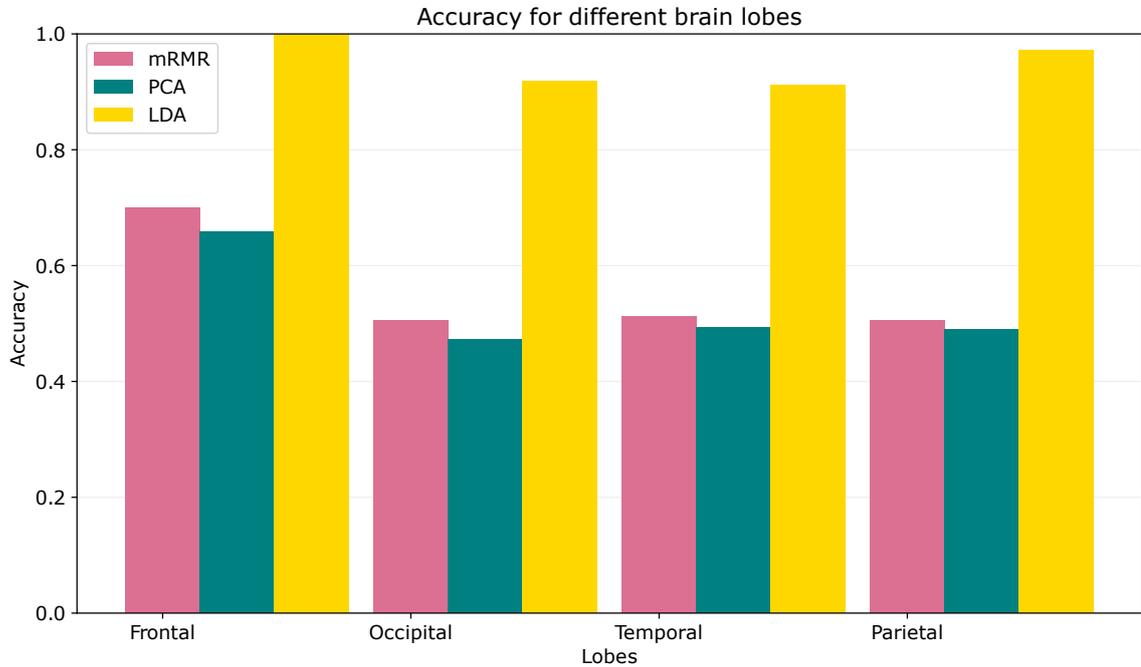


Figure 5.3: Brain regions and their accuracy.

### 5.2.3 Measuring the Accuracy of Frequency Bands Separately

This subsection describes how to detect a features group with the most significant impact on emotion detection. It is necessary to avoid taking all features calculated from all five waves. Instead, a comparison process took place, calculating time-domain features of all waves separately, then repeated with frequency features. Testing was made using naive Bayes and LDA feature selection method.

The first figure 5.4 shows that frontal lobe frequency features were the most dominant.

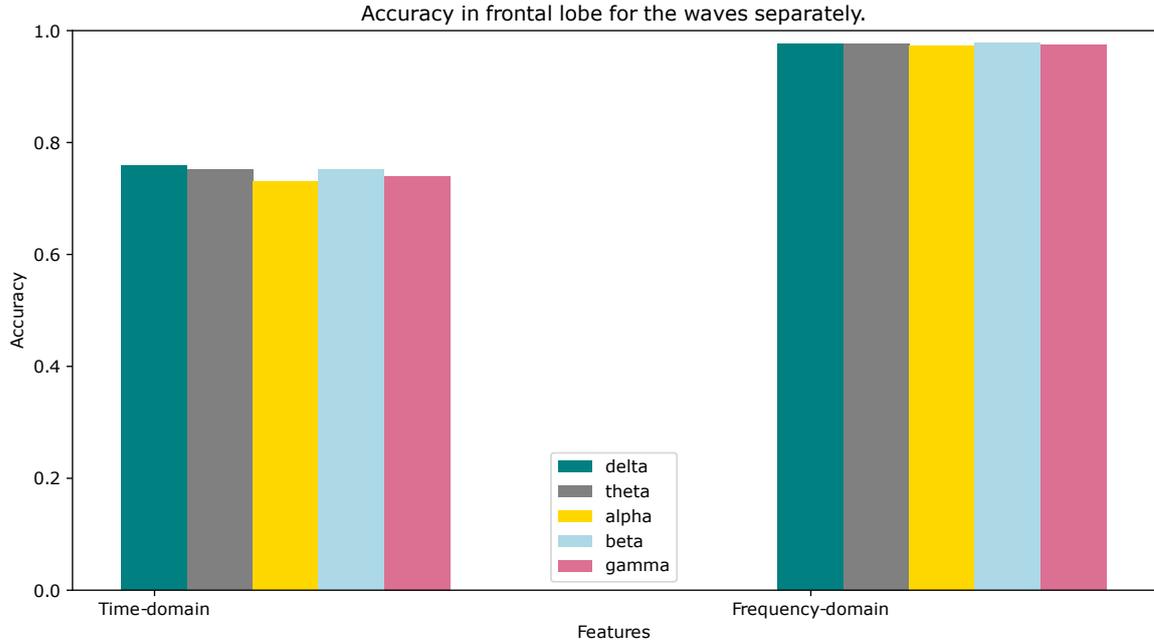


Figure 5.4: Frontal lobe and its accuracy for different waves and features.

The weakest part turned out to be time-domain features. Its results are around 50-60% accuracy, which was due to the fact that the classification was not successful (the same problem was mentioned here 5.1.3). From my experiments, I discovered that if the classifier does not receive sufficiently high-quality data, it evaluates the entire test set with only one class, usually the one that is represented the most times in the test set. This problem also happened with time-domain because there is usually more than a 50% chance that the classifier hit and guessed the correct label since the labels are approximately evenly spaced.

## 5.3 Proposed Seature Selection Method

All the experiments in the previous sections aimed to derive a useful method for selection. Here in this section is a summary of what the resulting solution looked like, as it consisted of several parts.

- The first test found that the form of labels VA with the choice of only two classes was the most effective. The reason was that it was the simplest compared to the other methods and, therefore the easiest for both the selector and the classifier.

- At the same time, this result was tested on several classifiers because the reference classifier should be the one that formed the best combination with the selector, and the resulting accuracy is therefore unaffected only by a bad match. Gaussian naive Bayes had a best match with LDA classifier.
- According to the assignment of the channels to their respective lobes, the frontal lobe stood out the most. After verification in the literature [3] that frontal lobe is truly significant as a source of information for emotion detection, I decided to use the frontal lobe itself as the primary source of features.

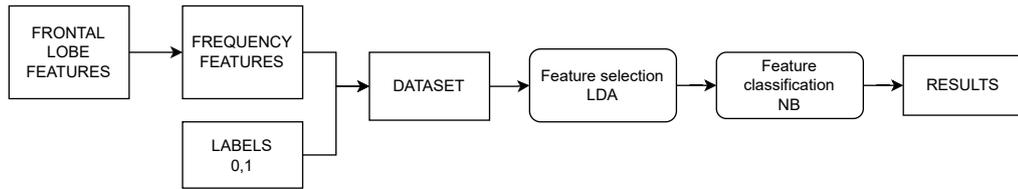


Figure 5.5: Pipeline of the final solution.

- Overall, the LDA selector showed the best result, so it became part of the final solution shown here 5.5.
- In the combination of LDA and features from the frontal lobe, the features of PSD extracted from the frequency-domain EEG signal proved the most useful.

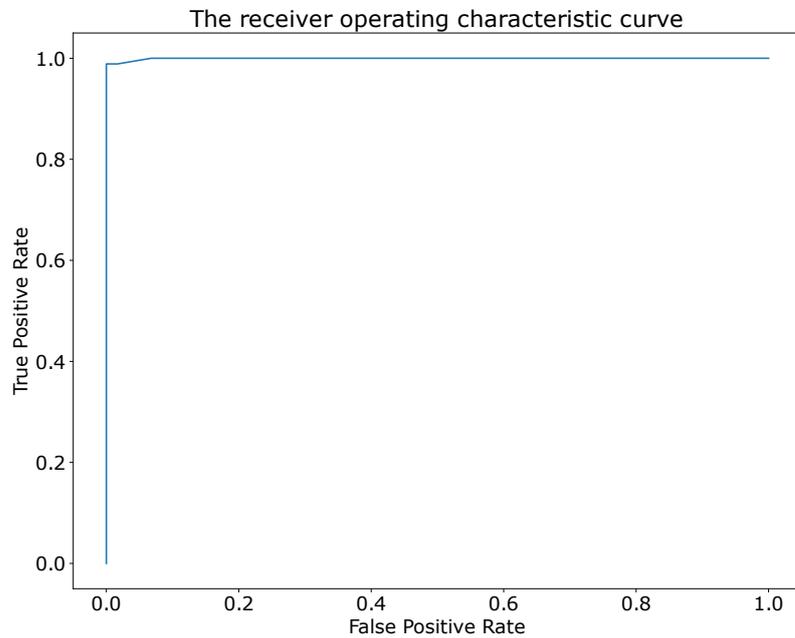


Figure 5.6: Receiver operating characteristic curve of accuracy score.

Model validation took place in the form of 2-fold cross-validation combined with hold-out cross-validation. Results may vary slightly depending on the randomization of the training and test subset data. For this reason, the validation would be repeated twenty times.

Its output was the average accuracy from these ten trials and its variance. Values were rounded to four decimal places. The results achieved an average value of 0.9805 accuracy. Figure 5.6 shows evaluation of LDA feature selection by Gaussian naive Bayes in ROC curve.

## 5.4 Discussion

From the achieved results, it was also possible to read other important values, which are contained in the table 5.3. The results were then compared with other studies dealing with the same topic and using same dataset DREAMER [23]. The study [13] used  $k$ -nearest neighbors and random forest models in which binary classification of arousal and valence occurred, so the result was achieved under similar conditions. However, it may differ in the indication of the labelling method, when they could use a different range for low and high arousal and valence.

In the study [13], they compared their results with other studies and methods. Their results<sup>1</sup> turned out to be the best of all the previous ones. My results were achieved with an LDA selector whose input were binary labels (0 for low and 1 for high) and the matrix composed of feature vectors. LDA output was evaluated by naive Bayes classifier with 2-Fold Cross-Validation. This process was repeated several times, and the results were averaged to achieve the most accurate results.

	My results NB		KNN [13]		RF [13]	
	Valence	Arousal	Valence	Arousal	Valence	Arousal
Recall	0.9543	0.9669	-	-	-	-
Precision	0.9864	0.9986	-	-	-	-
F1 score	0.9693	0.9824	-	-	-	-
Sensitivity	0.9543	0.9669	-	-	-	-
Specificity	0.9902	0.9990	-	-	-	-
Accuracy	<b>0.9761</b>	<b>0.9849</b>	0.9216	0.9372	0.9365	0.9379

Table 5.3: Table of achieved values.

My final solution had a positive result because my final results came out better than the study [13] that is compared in the table 5.3. My result also contained additional information such as recall, precision, F1-score, sensitivity and specificity. Therefore, my method was more realistic than the method proposed by [13] because I considered all evaluations of my solution.

<sup>1</sup><https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8155937/#B4-sensors-21-03414:-:text=5.3.1.%20Arousal%20and,the%20best%20one>.

# Chapter 6

## Implementation of My Solution

This chapter shows my implementation of the proposed solution from previous chapter. For reading simplification of my implementation, all variables were written in snake case and all functions in camel case.

### 6.1 Dependencies and Structure of Files

The scripts with my implemented solution use various libraries that must be installed for the program to run correctly. Therefore, I created a file to simplify the installation process of all dependencies named `requirements.txt`. File `requirements.txt` contains all required libraries. Command: `pip install -r requirements.txt` serves for running it.

Some of the scripts have different usage and parameters and run differently. Therefore, I created `README.md`, where is a description to all scripts.

List of used libraries:

- Scikit Learn [40], Librosa [35], Matplotlib [20], mrmr\_selection [34], antropy [50], scipy [51], seaborn [52], numpy [17], pandas [48], mne [16], PyQt6 [29]

### 6.2 Operations with Data

A significant part of the implementation was managing multiple different types of data. This section describes how measured EEG data and then their extracted features were stored and further converted into 2D arrays.

#### 6.2.1 Storing Data

The original dataset provided all the information about measured data and the subject's self-evaluation in `.mat` file structure. Due to using Python, I converted all EEG data into `.csv` files in MATLAB. These files were loaded during preprocessing.

Preprocessed signals with removed artifacts were saved into `.fif` files. I chose the `.fif` extension because I wanted to clearly distinguish the clean data from the original. Every preprocessed file was stored in a format: `[number of subject]-ica.fif`. Saving and loading `.fif` file was with the help of `mne` library and its function `mne.io.read_raw_fif()`.

Feature extraction was made to extract time-domain and frequency-domain features separately for further particular testing. Each of these two extraction outputs were stored in `.csv` files. The format of saving was `[time/freq]_features[movie clip number].csv`.

## Labels

Labels from the participant’s self-evaluation were converted into a format corresponding to a feature matrix described here [6.2.2](#).

Subject 1, movie clip 1	Subject 2, movie clip 1	...	Subject 23, movie clip 1
-------------------------	-------------------------	-----	--------------------------

Table 6.1: Structure of storing labels into the 1D array.

Every label array contained only valence or only arousal labels. Except one case, when valence and arousal labels were combined together to create the quadratic distribution.

Labels were extracted from `.mat` file into `.csv` file in a form where each row was one movie clip and every odd column was a valence label and an even column arousal label. The number of columns was 46 because two columns belonged to each subject. Script `extract_labels.py` converts the original subject’s self-evaluation into [4.2](#) labelling form. It takes `labels.csv` as an input parameter and returns `VA[arousal/valence].csv`, `N[arousal/valence].csv` and `Qboth.csv`.

## 6.2.2 Matrix of Features

In this case, it was not suitable to implement structures for storing data because every function from used libraries works with 2D array data types. The whole process of keeping data was continuously renewed when it led to applying some modification to the data. This modification was pre-processing, downsampling, normalization, and transformation into the final feature matrix for classification. After each, a new matrix was generated and was used as a whole, not as an individual.

	Feature 1	Feature 2	...	Feature 10220
Subject 1, movie clip 1				
Subject 2, movie clip 1				
...				
Subject 23, movie clip 1				

Table 6.2: Structure of storing data into the 2D array.

During feature selection and extraction, functions do not operate over rows, including individual information for one subject, but instead over columns. This behaviour is because knowledge about the quality of one feature needs a spectrum of all its occurrences to determine if the feature was important enough to be kept. Otherwise, the feature could be redundant or less critical for classification than others. According to which concrete movie clip data were needed, the following data in its own feature matrix were concatenated together in zero axis.

Matrixes of calculated features were stored in `.csv` files for quick and easy extraction into a numpy array or a data frame. For manipulation with `.csv` files was used `csv` library and `pandas` library.

### 6.2.3 Displaying the Results

Figures and charts were plotted with `matplotlib.pyplot` and `seaborn` libraries. All figure were implemented in `plot_graphs.py` and `artifacts_test.py()` files. Functions that rendered individual figure were called from functions implemented in other files. Figures were used to display the results and simplify connections between them. To unify the colours used in the figures, all colors were represented by constants at the beginning of the file. Function `plotPCA99()`, that plotted number of features needed to achieve 0.99 variance with PCA was used from [37].

In addition to the graphic variant, the results were displayed using logs. Functions that contained unimportant side functions were implemented in `help_functions.py`.

## 6.3 Pre-processing

Preparation of a signal was done in `preprocessing.ipynb` Jupiter notebook, because for interactive work with the signal, it was more suitable.

I created a raw array with `mne.io.RawArray()` with `info` parameter including all channel names with their type `EEG` and sampling rate equal to 128 Hz. This conversion into the raw EEG array led to the possibility of applying further signal transformation. As a filter, I decided to use a band-pass filter through a `fir` method.

I used a command `%matplotlib qt6` from `PyQt6` library to plot an interactive figure, where I was looking for artifacts that led to inaccurate results. These segments marked as `BAD_` had their starting index saved in `raw.annotations.onset` and duration of the `BAD_` segment in `raw.annotations.duration`. Based on this knowledge, the marked indexes were separated and deleted with `np.delete()` function. After these adjustments, the signal was ready for observation and extraction of features.

To verify that preprocessing did work and that artifacts were successfully removed, I compared calculated features from preprocessed data with not preprocessed. In a file `artifacts_test.py`, functions were implemented the same way as in 6.4, but with the difference that only skewness and kurtosis were extracted from the signal's time domain. Two bar figure were plotted for comparison.

## 6.4 Feature Extraction

This section shows how feature extraction was implemented into a file named `feature_extraction.py`. The first part describes the main algorithm for loading, calculating and storing data.

Firstly all files of subjects with EEG signals were loaded, converted into an array and stored in a array called `ALL_eeg_signals`, where every row contained one subject's data. Then, after calling a function `mainExtractionLoop()` I applied band-pass filters using `.filter()` function from `mne` library to pull out only certain wave's signal data. Ranges of single waves were established according to [22] and then regulated for coverage of a full frequency range 6.3.

Delta 0.5–4.0 Hz	Theta 4–8 Hz	Alpha 8–13 Hz	Beta 13–26 Hz	Gamma 26–40 Hz
------------------	--------------	---------------	---------------	----------------

Table 6.3: Consecutive waves.

A temporary array `list_f` was appended every time new features were calculated. Calculation of features took place in this order:

- By smaller value of frequency, the waves were picked, and all features from channel AF3 were calculated. Then from channel F7, until channel AF4.

AF3	F7	F3	FC5	T7	P7	O1	O2	P8	T8	FC6	F4	F8	AF4
-----	----	----	-----	----	----	----	----	----	----	-----	----	----	-----

Table 6.4: Sequence of channels.

- First wave was replaced with the following one and, again features from all channels were calculated.

A pseudo algorithm 1 visualizes the process.

---

**Algorithm 1:** The main process of extracting features.

---

**Input:** Preprocessed signal in `-ica.fif` files

**Output:** Matrix of features in `.csv` file

```

1: initialize a matrix of features
2: matrix_of_features = []
3: for every subject do
4:     apply band-pass filters to separate five waves
5:     for each wave do
6:         for each channel do
7:             data = wave.get_data()[channel]
8:             timeFeatures(data)
9:             frequencyFeatures(data)
10:        end for
11:    end for
12:    add features into matrix of features
13: end for
14: return matrix of features

```

---

### 6.4.1 Features

Classic time-domain features were calculated with `numpy`, `statistics`, `librosa` libraries. Hjorth parameters were calculated from `antropy` library the function `hjorth_params(x)` that returned mobility and complexity. Except for activity and its theorem, which is similar to a theorem of variability, for this reason I used `np.var()` function. To determine the number of peaks in the signal, the `find_peaks(x)` function was used to return two parameters. The first one is an array of indexes where peaks were captured. The number of peaks was the length of the index array.

The others were calculated using functions:

- `np.mean(x)`, `statistics.median(x)`, `np.var(x)`, `np.std(x)`, `skew(x)`, `kurtosis(x)`, `np.ptp(x)`, `np.sqrt(np.mean((x)**2))`, `np.min(x)`, `np.max(x)`, `sum(librosa.zero_crossings(x, pad=False))`

Power spectrum density was extracted from data with `scipy.signal.welch()` method. Python library `antropy` provides functions to calculate time-domain entropies:

- approximate `ant.app_entropy(x)`, permutation `ant.perm_entropy(x)` and sample `ant.sample_entropy(x)`.

### 6.4.2 Parameters

Computations of entropies were significantly time-consuming compared to other computations included in feature extraction. To save calculation time, I decided to implement into file `feature_extraction.py` conditions as parameters. These parameters enabled a choice of which group of features should be calculated. The parameter `all` indicated that all features had to be calculated, `time` called calculation of only time-domain features and `freq` only frequency-domain.

## 6.5 Feature Selection

The implementation of methods for feature selection was divided into the implementation of the methods themselves and other related parts.

The beginning was the implementation of PCA because its steps were further used in other parts of the solution. I used this page [7] as a guide for correctly using the library called `sklearn.decomposition`. First, the implemented function `PCANumber()` in `selection_help_functions.py` was used inside with the function `PCA(n_components = 0.99)`, where the number 0.99 shows that it is necessary to keep only the number of components that make up 99% of the variation. This is because there were many extracted features, and it was necessary to define from the beginning number (`PCA(n_components)`) to limit them. The function `pca.fit_transform()` was used for PCA selection, which reduced the number of dimensions in the data and left only relevant features.

To implement PCA selection of numbers to achieve 0.99 variance 4.5 in part 4.2.2, I used guide [37]. A code to plot 0.99 variance in this function `plotPCA99()` in `plot_graphs.py` was used from [37] and edited.

To implement the mRMR method, I used the `mrmr` library by Samuele Mazzanti, who described it in the article [33]. Used function was `mrmr_classif()`, whereas a parameter I inserted the result from the PCA calculation of the function `PCANumber()` from the file `selection_help_functions.py`. The other two parameters were data and labels related to them. For the application of the Pearson correlation coefficient, I used the implementation from [49], namely the function named `identify_correlated()`, which can be found in the file `selection_help_functions.py`. At the same time, I used parts of his code and modified them as needed in the function I called `plotHeatmap()` in the file `plot_graphs.py`.

For the implementation of LDA, I used the `sklearn.discriminant_analysis` library, of which the function `LinearDiscriminantAnalysis()` was most needed, which created an LDA environment for transforming the feature matrix into the lower data dimension with `.fit()` and `.transform()` methods.

## 6.6 Learning Classifiers

Implementation of the last step in the pipeline 4.1 of learning classifiers was in the file `feature_classification.py`. It consists of applying 2.9, which wraps up all the learning classifier's functions.  $K$ -fold was implemented with library `sklearn.model_selection` and its implementation was based on this [31] source.

After splitting the matrix of features into five parts, these parts rotate in a  $k$ -fold loop and make up `X_train`, `X_test`, `y_train`, `y_test` parameters for classifier learning functions.

---

**Algorithm 2:** 5-fold cross-validation

---

**Input:** Matrix of features as an array

**Output:** Accuracy

- 1: divide matrix of features into 5 parts
  - 2: **for** *every combination of parts* **do**
  - 3:     train learning classifier models with `X_train`, `y_train`
  - 4:     test learning classifier models with `X_test`, `y_test`
  - 5: **end for**
  - 6: **return** Average accuracy from predicted data `y_pred`
- 

- The library `sklearn` was used to implement the SVM learning classifier, SVM requested two essential parameters. One of them is *gamma*, and the second is *C*. After creating SVM with `svm.SVC(kernel, C, gamma)` function, for training the `.fit(X_train, y_train)` function was used.

Prediction `y_pred` is a result of `.predict(X_test)`.

- `KNeighborsClassifier()` function from `sklearn.neighbors` was used for the implementation of  $k$ -nearest neighbors method. The implementation process was the same as for SVM classification, but I used parameter `n_neighbors` instead of parameter `C` and *gamma*. The *n\_neighbors* influenced the duration of the search of data for predicting labels.
- The implementation of Gaussian Naive Bayes was not fundamentally different either. Function `GaussianNB` from `sklearn.naive_bayes` setted a model for further fitting and testing. No additional parameters were used for `GaussianNB()`.

The critical result came from this prediction used by `sklearn` library for `metrics` and checked by `metrics.accuracy_score(y_test, y_pred)`. The results were achieved with the help of functions: `precision_score()`, `recall_score()`, `f1_score()` and `accuracy_score()`.

### 6.6.1 Testing Learning Classifiers

*Gamma*, *C* and *n neighbor* were able to affect the accuracy results. Therefore, for significant uninfluenced changes in testing during the implementation of SVM and  $k$ -nearest neighbors, I decided to make a test which determined the most called parameter for *gamma* and *C*. A file `best_parameters.py` returned four values. The first two were calculated from SVM

with *RBF* and the third from SVM with *linear* kernel and the last one from *k*-nearest neighbors with *K* parameter that represented number of neighbors.

The overall implementation is on the same base as 6.6, but with a difference that, in this case, was the resulting value of accuracy insignificant because it returned only the value of *C* and *gamma* parameters. There was no difference for SVM with *linear* kernel, except no *gamma* parameters were presented. A process of searching for the best *C* and *gamma* match with the dataset is a **for loop**, where different combinations of values are successively applied to the classifier. The differences for *gamma* and *C* range from 0.001 to 100.0 for SVM and for *k*-Nearest Neighbors, *K* went in range 1 to 20.

A new learning classifier is trained every time for new parameters. The test took the matrix of features and, according labels, called a selection into individual channels..

## 6.7 Process of Selection and its Classification Evaluation

All individual parts of the solution had to be unified to evaluate the selection of features by the classifier. The incorporating all these parts shows pseudo-code of the algorithm 6.7.

---

**Algorithm 3:** The overall process of selecting selection and classification.

---

**Input:** Matrix of features as `.csv` file.

**Output:** Classification accuracy.

- 1: load feature matrix
  - 2: load labels
  - 3: normalize the
  - 4: pass matrix of features and labels into selector
  - 5: evaluate feature selection
  - 6: **return** accuracy
- 

Usage of all parts together was in the files:

- `different_method_test.py`. This script tested two scenarios. The first one was a comparison of all feature selection methods (including different labels) with all feature classification methods to exclude the worst ones. Thus, all selection functions and classification algorithms were implemented here. Due to the fact that testing different label systems (*VA*, *N*, *Q*) requires differently implemented methods for selection, I divided them all into two parts. One of them for *VA* and *N* labels had functions named as `FSmRMR()`, `FSPCA()`, `FSLDA()`, `channels` and the other *Q* as `FSmRMR4()`, `FSPCA4()`, `FSLDA4()`, `channels4()`, because the functions worked with four labels.

The second scenario took only naive Bayes for classification, due to its best results and compared feature selection methods one more time. The results were accuracies for different channels and lobes for each selection method.

- `waves_separately.py`. In this file the impact of using waves separately was tested. LDA with NB ended up with the best score in `different_method_test.py`. Therefore, they are implemented here as the only methods. The files `waves_separately.py` and `proposed_solution.py` did not perform its process for all channels, but for only those located in the frontal lobe area.

- `proposed_solution.py`. This was the final solution, which was created from the results of previous tests. It had its own implementation of the LDA method because it differed significantly in its parameters and return arguments.

The necessary data files are loaded first. Then they were reduced to only specific features according to the location of the channel or their frequency band, which were needed for the next steps. The data were normalized and moved further into the selector. After selection, the resulting matrix is used as input to the classifier. The classifier evaluated the quality of the feature selection and the features used. The output was a `float` number that showed the accuracy of the selected channel and frequency band or a group of channels and frequency bands.

# Chapter 7

## Results

The main aim of this work was to design a feature extraction and feature selection model able to predict emotions. I studied EEG signals, their behaviour, and other related topics in the classifier learning specialization to develop this model.

When initially creating the proposal model, I considered the existing options and tried to use their great extent and combinations. From this came several ideas for the model's design, which were then compared to each other during experiments.

The best results were achieved by the Linear Discriminant Analysis algorithm for feature selection and evaluation by the Naive Bayes classifier. After its separate implementation and other tests, I reached a final result of 0.98 accuracy. Compared to other studies on this topic, I achieved noticeably better results. For this reason, my work also contributes to the further analysis of emotion detection.

### 7.1 Improvement Proposal

Evaluation of the results involves comparing predicted values with defaults for testing. In this case, the subject's self-evaluation corresponded to the primary valence and arousal distribution because if the model can distinguish two classes with such high accuracy, it could distinguish any other more complex labels. My solution would continue using this primary unit, which works with two classes, and assemble a structure from it with more units, which would form a complex emotion detection process. This process would be able to distinguish individual emotions precisely.

# Bibliography

- [1] AGARWAL, R., ANDUJAR, M. and CANAVAN, S. Classification of emotions using EEG activity associated with different areas of the brain. *Pattern Recognition Letters*. 2022, vol. 162, p. 71–80. DOI: <https://doi.org/10.1016/j.patrec.2022.08.018>. ISSN 0167-8655. Available at: <https://www.sciencedirect.com/science/article/pii/S016786552200263X>.
- [2] AL NAFJAN, A. Feature selection of EEG signals in neuromarketing. *PeerJ Comput. Sci.* PeerJ. april 2022, vol. 8, e944. DOI: <https://doi.org/10.7717/peerj-cs.944>. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9138093/>.
- [3] AL QAZZAZ, N. K., SABIR, M. K., ALI, S. H. B. M., AHMAD, S. A. and GRAMMER, K. Electroencephalogram profiles for emotion identification over the brain regions using spectral, entropy and temporal biomarkers. *Sensors (Basel)*. MDPI AG. december 2019, vol. 20, no. 1, p. 59. DOI: <https://doi.org/10.3390/s20010059>. Available at: <https://www.mdpi.com/1424-8220/20/1/59>.
- [4] ALLIBHAI, E. *Hold-out vs. Cross-validation in Machine Learning* [online]. October 2018. Accessed: 2023-4-13. Available at: <https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f>.
- [5] ATKINSON, J. and CAMPOS, D. Improving BCI-based emotion recognition by combining EEG feature selection and kernel classifiers. *Expert Systems with Applications*. 2016, vol. 47, p. 35–41. DOI: <https://doi.org/10.1016/j.eswa.2015.10.049>. ISSN 0957-4174. Available at: <https://www.sciencedirect.com/science/article/pii/S0957417415007538>.
- [6] BIN, N. W., AWANG, S. A., FOOK, C. Y., CHIN, L. C. and YING, O. Z. A Study of Informative EEG Channel and Brain Region for Typing Activity. *Journal of Physics: Conference Series*. IOP Publishing. nov 2019, vol. 1372, no. 1, p. 012008. DOI: [10.1088/1742-6596/1372/1/012008](https://doi.org/10.1088/1742-6596/1372/1/012008). Available at: <https://dx.doi.org/10.1088/1742-6596/1372/1/012008>.
- [7] CHENG, C. *Principal Component Analysis (PCA) explained visually with zero math* [online]. Towards Data Science, february 2022. Accessed: 2023-4-13. Available at: <https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d>.
- [8] CHOONG, W. Y., KHAIRUNIZAM, W., MUSTAFA, W. A., MURUGAPPAN, M., HAMID, A. et al. Correlation Analysis of Emotional EEG In Alpha, Beta and Gamma Frequency Bands. *Journal of Physics: Conference Series*. IOP Publishing. aug 2021,

vol. 1997, no. 1, p. 012029. DOI: 10.1088/1742-6596/1997/1/012029. Available at: <https://dx.doi.org/10.1088/1742-6596/1997/1/012029>.

- [9] COHEN, M. X. Where Does EEG Come From and What Does It Mean? *Trends in Neurosciences*. 2017, vol. 40, no. 4, p. 208–218. DOI: <https://doi.org/10.1016/j.tins.2017.02.004>. ISSN 0166-2236. Available at: <https://www.sciencedirect.com/science/article/pii/S0166223617300243>.
- [10] DE CHEVEIGNÉ, A. and NELKEN, I. Filters: When, Why, and How (Not) to Use Them. *Neuron*. 2019, vol. 102, no. 2, p. 280–293. DOI: <https://doi.org/10.1016/j.neuron.2019.02.039>. ISSN 0896-6273. Available at: <https://www.sciencedirect.com/science/article/pii/S0896627319301746>.
- [11] DEMPSTER, J. CHAPTER SIX - Signal Analysis and Measurement. In: DEMPSTER, J., ed. *The Laboratory Computer*. London: Academic Press, 2001, p. 136–171. Biological Techniques Series. DOI: <https://doi.org/10.1016/B978-012209551-1/50039-8>. ISSN 08924473. Available at: <https://www.sciencedirect.com/science/article/pii/B9780122095511500398>.
- [12] FUAD, N., JAILANI, R., OMAR, W., JAHIDIN, A. and TAIB, M. Three dimension 3D signal for electroencephalographic (EEG). In: *2012 IEEE Control and System Graduate Research Colloquium*. 2012, p. 262–266. DOI: 10.1109/ICSGRC.2012.6287173. Available at: <https://ieeexplore.ieee.org/document/6287173>.
- [13] GALVÃO, F., ALARCÃO, S. M. and FONSECA, M. J. Predicting exact valence and arousal values from EEG. *Sensors (Basel)*. MDPI AG. may 2021, vol. 21, no. 10, p. 3414. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8155937/>.
- [14] GANDHI, R. *Support Vector Machine — Introduction to Machine Learning Algorithms*. Towards Data Science, 07. june 2018 [cit. 2023-04-03]. Available at: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.
- [15] GONÇALES, L. J., FARIAS, K., KUPSSINSKÜ, L. and SEGALOTTO, M. The effects of applying filters on EEG signals for classifying developers' code comprehension. *J. Appl. Res. Technol.* Universidad Nacional Autonoma de Mexico. december 2021, vol. 19, no. 6, p. 584–602. Available at: [https://www.scielo.org.mx/scielo.php?pid=S1665-64232021000600584&script=sci\\_arttext&tlng=en](https://www.scielo.org.mx/scielo.php?pid=S1665-64232021000600584&script=sci_arttext&tlng=en).
- [16] GRAMFORT, A., LUESSI, M., LARSON, E., ENGEMANN, D. A., STROHMEIER, D. et al. MEG and EEG Data Analysis with MNE-Python. *Frontiers in Neuroscience*. 2013, vol. 7, no. 267, p. 1–13. DOI: 10.3389/fnins.2013.00267. Available at: <https://mne.tools/stable/overview/cite.html>.
- [17] HARRIS, C. R., MILLMAN, K. J., WALT, S. J. van der, GOMMERS, R., VIRTANEN, P. et al. Array programming with NumPy. *Nature*. Springer Science and Business Media LLC. september 2020, vol. 585, no. 7825, p. 357–362. DOI: 10.1038/s41586-020-2649-2. Available at: <https://doi.org/10.1038/s41586-020-2649-2>.

- [18] HERWIG, U., SATRAPI, P. and SCHÖNFELDT LECUONA, C. Using the international 10-20 EEG system for positioning of transcranial magnetic stimulation. *Brain Topogr.* Springer Nature. 2003, vol. 16, no. 2, p. 95–99. DOI: 10.1023/b:brat.0000006333.93597.9d. Available at: <https://pubmed.ncbi.nlm.nih.gov/14977202/>.
- [19] HUBENÝ, J. *Metody automatické detekce EEG epileptických hrotů*. Bachelor’s thesis. Available at: <https://dspace.cvut.cz/handle/10467/74686>.
- [20] HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering.* IEEE COMPUTER SOC. 2007, vol. 9, no. 3, p. 90–95. DOI: 10.1109/MCSE.2007.55. Available at: <https://matplotlib.org/stable/users/project/citing.html>.
- [21] KAMBLE, V. H. and DALE, M. P. Chapter 1 - Machine learning approach for longitudinal face recognition of children. In: SARANGI, P. P., PANDA, M., MISHRA, S., MISHRA, B. S. P. and MAJHI, B., ed. *Machine Learning for Biometrics*. Academic Press, 2022, p. 1–27. Cognitive Data Science in Sustainable Computing. DOI: <https://doi.org/10.1016/B978-0-323-85209-8.00011-0>. ISBN 978-0-323-85209-8. Available at: <https://www.sciencedirect.com/science/article/pii/B9780323852098000110>.
- [22] KAMEL, N. and MALIK, A. S. *EEG/ERP Analysis: Methods and Applications*. CRC Press, 2014. ISBN 9781482224719. Available at: <https://www.taylorfrancis.com/books/edit/10.1201/b17605/eeg-erp-analysis-aamir-saeed-malik-kamel-nidal>.
- [23] KATSIKIANNIS, S. and RAMZAN, N. DREAMER: A Database for Emotion Recognition Through EEG and ECG Signals From Wireless Low-cost Off-the-Shelf Devices. *IEEE Journal of Biomedical and Health Informatics*. 2018, vol. 22, no. 1, p. 98–107. DOI: 10.1109/JBHI.2017.2688239. Available at: <https://ieeexplore.ieee.org/document/7887697>.
- [24] KHOSLA, A., KHANDNOR, P. and CHAND, T. A comparative analysis of signal processing and classification methods for different applications based on EEG signals. *Biocybernetics and Biomedical Engineering*. 2020, vol. 40, no. 2, p. 649–690. DOI: <https://doi.org/10.1016/j.bbe.2020.02.002>. ISSN 0208-5216. Available at: <https://www.sciencedirect.com/science/article/pii/S0208521620300231>.
- [25] KUMAR, N. *Using Support Vector Machines Effectively*. Mar 2014. Available at: <https://neerajkumar.org/writings/svm/>.
- [26] LEE, W.-M. *Statistics in Python — Understanding Variance, Covariance, and Correlation* [online]. Towards Data Science, september 2021. Accessed: 2023-4-13. Available at: <https://towardsdatascience.com/statistics-in-python-understanding-variance-covariance-and-correlation-4729b528db01>.
- [27] LI, S., ZHOU, W., YUAN, Q., GENG, S. and CAI, D. Feature extraction and recognition of ictal EEG using EMD and SVM. *Computers in Biology and Medicine*. 2013, vol. 43, no. 7, p. 807–816. DOI: <https://doi.org/10.1016/j.compbiomed.2013.04.002>. ISSN 0010-4825. Available at: <https://www.sciencedirect.com/science/article/pii/S0010482513000905>.

- [28] LIGHT, G. A., WILLIAMS, L. E., MINOW, F., SPROCK, J., RISSLING, A. et al. Electroencephalography (EEG) and event-related potentials (ERPs) with human participants. *Curr. Protoc. Neurosci.* Wiley. july 2010, vol. 52, no. 1, p. 6.25.1–6.25.24. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2909037/>.
- [29] LIMITED, R. C. *PyQt6*. April 2023. Available at: <https://pypi.org/project/PyQt6/>.
- [30] LOPEZ GORDO, M., SANCHEZ MORILLO, D. and VALLE, F. Dry EEG Electrodes. *Sensors*. MDPI AG. Jul 2014, vol. 14, no. 7, p. 12847–12870. DOI: 10.3390/s140712847. ISSN 1424-8220. Available at: <http://dx.doi.org/10.3390/s140712847>.
- [31] LYASHENKO, V. and JHA, A. *Cross-validation in Machine Learning: How To Do It Right*. Jan 2023. Available at: <https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right>.
- [32] MALDONADO, K. A. and ALSAYOURI, K. Physiology, Brain. *Physiology, Brain*. In *StatPearls*. StatPearls Publishing. march 2023. Available at: <https://www.ncbi.nlm.nih.gov/books/NBK551718/>.
- [33] MAZZANTI, S. *Mrmr explained exactly how you wished someone explained to you* [online]. Towards Data Science, february 2021. Accessed: 2023-4-13. Available at: <https://towardsdatascience.com/mrmr-explained-exactly-how-you-wished-someone-explained-to-you-9cf4ed27458b>.
- [34] MAZZANTI, S. *MRMR* [online]. GitHub, december 2022. Available at: <https://github.com/smazzanti/mrmr>.
- [35] MCFEE, B., RAFFEL, C., LIANG, D., ELLIS, D. P., MCVICAR, M. et al. Librosa Audio and music signal analysis in python. In: *Proceedings of the 14th python in science conference*. 2015, p. 18–25. Available at: <https://librosa.org/doc/latest/index.html>.
- [36] MERIZALDE, Y., HERNÁNDEZ CALLEJO, L. and DUQUE, O. State of the Art and Trends in the Monitoring, Detection and Diagnosis of Failures in Electric Induction Motors. *Energies*. july 2017, vol. 10, p. 1056. DOI: 10.3390/en10071056. Available at: <https://www.mdpi.com/1996-1073/10/7/1056>.
- [37] MIKULSKI, B. *PCA — how to choose the number of components?* [online]. June 2019. Accessed: 2023-4-13. Available at: <https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components/>.
- [38] NAKISA, B., RASTGOO, M. N., TJONDRONEGORO, D. and CHANDRAN, V. Evolutionary computation algorithms for feature selection of EEG-based emotion recognition using mobile sensors. *Expert Systems with Applications*. 2018, vol. 93, p. 143–155. DOI: <https://doi.org/10.1016/j.eswa.2017.09.062>. ISSN 0957-4174. Available at: <https://www.sciencedirect.com/science/article/pii/S0957417417306747>.
- [39] OH, S.-H., LEE, Y.-R. and KIM, H.-N. A Novel EEG Feature Extraction Method Using Hjorth Parameter. *International Journal of Electronics and Electrical*

- Engineering*. 2014, p. 106–110. Available at:  
<https://www.semanticscholar.org/paper/A-Novel-EEG-Feature-Extraction-Method-Using-Hjorth-Oh-Lee/546864bcc8d88111893268eda6fc1a6cd74d1363>.
- [40] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, p. 2825–2830. Available at:  
<https://scikit-learn.org/stable/about.html#citing-scikit-learn>.
- [41] POR, E., KOOTEN, M. van and SARKOVIC, V. *Nyquist–Shannon sampling theorem* [online]. Universiteit Leiden, may 2019 [cit. 2023-25-02]. Available at: [https://home.strw.leidenuniv.nl/~por/AOT2019/docs/AOT\\_2019\\_Ex13\\_NyquistTheorem.pdf](https://home.strw.leidenuniv.nl/~por/AOT2019/docs/AOT_2019_Ex13_NyquistTheorem.pdf).
- [42] RADOVIC, M., GHALWASH, M., FILIPOVIC, N. and OBRADOVIC, Z. Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC Bioinformatics*. Springer Science and Business Media LLC. january 2017, vol. 18, no. 1, p. 9. Available at:  
<https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1423-9>.
- [43] SAME, M. H., GANDUBERT, G., GLEETON, G., IVANOV, P. and LANDRY, R. Simplified Welch Algorithm for Spectrum Monitoring. *Applied Sciences*. 2021, vol. 11, no. 1. DOI: 10.3390/app11010086. ISSN 2076-3417. Available at:  
<https://www.mdpi.com/2076-3417/11/1/86>.
- [44] SAWLA, S. *Linear Discriminant Analysis - Srishti sawla* [online]. June 2018. Accessed: 2023-4-13. Available at:  
<https://medium.com/@srishtisawla/linear-discriminant-analysis-d38decf48105>.
- [45] SEN GUPTA, S., SOMAN, S., GOVIND RAJ, P., PRAKASH, R., SAILAJA, S. et al. Detecting eye movements in EEG for controlling devices. In: *2012 IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*. 2012, p. 69–73. DOI: 10.1109/CyberneticsCom.2012.6381619. Available at: <https://ieeexplore.ieee.org/document/6381619>.
- [46] SHRIRAM, R., SUNDHARARAJAN, D. and DAIMIWAL, N. EEG Based Cognitive Workload Assessment for Maximum Efficiency. *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*. august 2012. Available at:  
[https://www.researchgate.net/publication/286371237\\_EEG\\_Based\\_Cognitive\\_Workload\\_Assessment\\_for\\_Maximum\\_Efficiency](https://www.researchgate.net/publication/286371237_EEG_Based_Cognitive_Workload_Assessment_for_Maximum_Efficiency).
- [47] STANCIN, I., CIFREK, M. and JOVIC, A. A review of EEG signal features and their application in driver drowsiness detection systems. *Sensors (Basel)*. MDPI AG. may 2021, vol. 21, no. 11, p. 3786. Available at:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8198610/>.
- [48] TEAM, T. pandas development. *Pandas-dev/pandas: Pandas*. February 2020. Available at: <https://doi.org/10.5281/zenodo.3509134>.
- [49] TUYCHIEV, B. *How to use pairwise correlation for robust feature selection*. Towards Data Science, Apr 2021. Available at: <https://towardsdatascience.com/how-to-use-pairwise-correlation-for-robust-feature-selection-20a60ef7d10>.

- [50] VALLAT, R. *AntroPy* [online]. GitHub, december 2022. Available at: <https://github.com/raphaelvallat/antropy>.
- [51] VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, vol. 17, p. 261–272. DOI: 10.1038/s41592-019-0686-2. Available at: <https://rdcu.be/b08Wh>.
- [52] WASKOM, M. L. Seaborn: statistical data visualization. *Journal of Open Source Software*. The Open Journal. 2021, vol. 6, no. 60, p. 3021. DOI: 10.21105/joss.03021. Available at: <https://doi.org/10.21105/joss.03021>.
- [53] WEI, Z., WU, C., WANG, X., SUPRATAK, A., WANG, P. et al. Using Support Vector Machine on EEG for Advertisement Impact Assessment. *Frontiers in Neuroscience*. 2018, vol. 12. DOI: 10.3389/fnins.2018.00076. ISSN 1662-453X. Available at: <https://www.frontiersin.org/articles/10.3389/fnins.2018.00076>.
- [54] WRIGHT, G. *What is peak-to-peak in a waveform?* TechTarget, Oct 2022. Available at: <https://www.techtarget.com/whatis/definition/peak-to-peak-pk-pk>.
- [55] XIAO, Y., KONG, W. and LIANG, Z. Short-Term Demand Forecasting of Urban Online Car-Hailing Based on the K-Nearest Neighbor Model. *Sensors*. 2022, vol. 22, no. 23. DOI: 10.3390/s22239456. ISSN 1424-8220. Available at: <https://www.mdpi.com/1424-8220/22/23/9456>.
- [56] YIU, T. *Understanding PCA (principal components analysis)* [online]. Towards Data Science, july 2019. Accessed: 2023-4-13. Available at: <https://towardsdatascience.com/understanding-pca-fae3e243731d>.
- [57] YU, L.-C., LEE, L.-H., HAO, S., WANG, J., HE, Y. et al. Building Chinese Affective Resources in Valence-Arousal Dimensions. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, p. 540–545. DOI: 10.18653/v1/N16-1066. Available at: <https://aclanthology.org/N16-1066>.