Stefan Tosic, BSc

EEG CLASSIFICATION OF BRAIN RESPONSES TO FAMILIAR AND UNFAMILIAR VOICES

BACHELOR'S THESIS

to achieve the university degree of

B.Sc

Bachelor's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

Valeria Mondini, Dott.ssa Dott.ssa mag.
and Shayan Jalilpourroudkoli
Institue of Neuroscience

Graz, August 2022

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present bachelor's thesis.

Date, Signature

Contents

List of Figures

List of Tables

Abstract

The goal of this thesis is to analyse the distinctive characteristics of perceiving voices of familiar and unfamiliar people, by decoding the collected data. In other words, the objective of this thesis was to examine whether it is possible to discriminate different familiar voices from an unfamiliar one by finding neural correlates related to the different speakers. For that to happen, different values were utilized as features, like the *Inter-site phase clustering* (ISPC) and *power spectral density* (PSD). These extracted features were then combined with classifiers to achieve the identification and classification of these familiar and unfamiliar voices. Electroencephalography (EEG) constituted as a bridge for investigating the human voice perception system in the brain. However, unlike other studies, within this research area, no event-related potentials (ERPs) were used for clarifying the difference between responses to unknown voices, intimately familiar voices, less familiar voices, and voices that the proband barely knows. The classification of the features revealed different responses for intimately familiar and unfamiliar voices. In fact, very familiar voices are getting better predicted by the classifiers than the unfamiliar ones. Also, specific frequencies, such as alpha and theta, occurred during the classification far more often than the other two sub-frequency bands.

1 Introduction

In our everyday lives, we encounter all different types of people that we categorize as familiar or unfamiliar based on their look, voice, smell, or overall appearance. Hence, this already implies that we can recognize a person based on those characteristics. Taking this concept one step further, it becomes apparent that people may also categorize familiar people into further different subgroups of familiarity. For instance, someone from work might be allocated as "familiar", but rather not categorized under the same familiar group as e.g. close family members who are likely to spend considerably more time together with one than work colleagues.
Based on the way someone acts in front of a person can also be used to determine how well they know this specific person. In general, it can be said that person recognition influences our behaviour, the way we speak as well as the way we feel when talking to differently known people. Depending on how someone categorizes another person, for instance, as a stranger or a friend, has crucial effects on the activity pattern in someone's brain. Different factors play a significant role here. For example, factors like the face, voice or even smell of the other person can determine if someone recognizes them or not. Numerous studies based on neuroscience and behaviourism have revealed that recognizing people is a complex cognitive process mediated by a series of face-, voice-, and verbal-specific brain processes [6]. There is even a distinction based on the activity between the two hemispheres while trying to recognize familiar and unfamiliar people. For instance, if the right hemisphere is more active than the left one, it indicates that the participant was talking to someone familiar [6].
However, unlike these studies, the main focus of this work is only to decode the data and see if there is a difference when comparing different familiar voices to an unfamiliar one, with relatively long stimuli and without ERPs. So, in total there were three different types of familiar voices taken into account. In order to determine if the stimuli results in various activity patterns, one must find the activity pattern hidden in the noisy, erratic neuronal data. After one has effectively deciphered the information from our brain's activity, it is then possible to "predict" the stimulus that activates the brain activity, and if the accuracy of "prediction" is greater than just random guessing (chance accuracy), it indicates the success of decoding. Besides, the focus here was not to see whether participants could recognize or identify a specific person, but rather to demonstrate if it is possible to differentiate between all these voices (conditions) without using event-related potentials (ERPs). Hence, it appears rather like reaching into the "raw" brain signals and trying to find variances between these conditions.
Only a few studies exist with a similar approach, yet most of them use much shorter stimuli that had been applied many times and in the end averaged, to get event-related potentials. A study with similar experimental procedure and methods can be found for

instance, in the papers Wenwen Chang et al [6] and Alberto Ara [2].
The entire procedure, from identifying another person to then seeing how they behave, is a very individual process for every participant, since it really depends on the cortical networks involved in retrieving human identities as well as on how fast the coactivation of visual, acoustic, and speech regions is. These factors, however, raise the overall complexity of this study, making the whole experiment far more challenging [6].

1.1 Workflow

As mentioned above, the conducted procedure in this thesis differs from the previously mentioned approach in the studies. Therefore, the general workflow of the paradigm looks different, as it can be seen in the following Figure 1.1. The first step starts in the same way as in the other studies: by collecting data by measuring the EEG from the participants with an electrode cap and an EEG amplifier. Afterwards, the collected data is pre-processed, with bandpass-, notch filters, and a few other processing techniques, like ICA to remove some of the artifacts, such as EOG, EMG or channel noise. Up to this point, the procedures are still very similar. The next step from the studies ([6] and [15]) would be to average over the trials to get phase-locked values, but this has been left out. Instead, the cleaned data is being epoched, and specific features are then extracted, which depicts the third step in Figure 1.1. Although before one could extract the features, some of the epoched trials were removed from the dataset, due to the trial's rejection criteria that had been fixed beforehand. With the help of these features then, it is possible to train a classifier, and determine how good the distinction between different voices is.

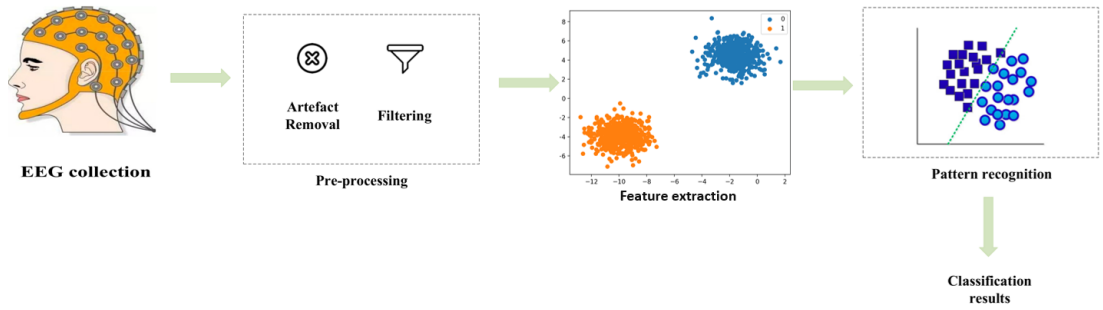


Figure 1.1: *Workflow of the thesis with its 5 simplified steps, whereby the electrode cap indicates the start point. Followed by the pre pre-processing step. Right after, the feature extraction begins and afterwards, the classification ends the workflow [22].*

1.2 EGG signal

One of the most famous brain imaging methods for recording these changes in our head is Electroencephalography (EEG). Due to its relatively simple, economical, and portable application, it is a very popular tool among such studies and for non-invasive BCI applications. Figure 1.2 illustrates an example of a few EEG signals from different channels.
The general EEG signal is produced by millions of neurons firing at the same time and creating tiny electrical fields throughout the brain. The signals are the result of postsynaptic potentials (PSP), which can be distinguished into excitatory postsynaptic potentials (EPSP) and inhibitory postsynaptic potentials (IPSP). But in principle, only the PSPs are responsible for creating the spikes. These electrical charges can then be detected with electrodes and to make them more visible, amplifiers are needed. The only problem left, is that the signal is very susceptible to technical as well as biological artifacts. Some examples of technical artifacts are the electromagnetic fields (e.g., 50 Hz), the amplifier noise, aliasing, and the AD converter quantization noise. Biological artifacts are produced by the participant himself, which include EOG, EKG as well as EMG. Unlike technical artifacts, biological ones cannot be removed with filters or electrical shielding. They only can be prevented up to a specific degree by adjusting the conditions for the participants or correcting the data afterwards, using computational methods.

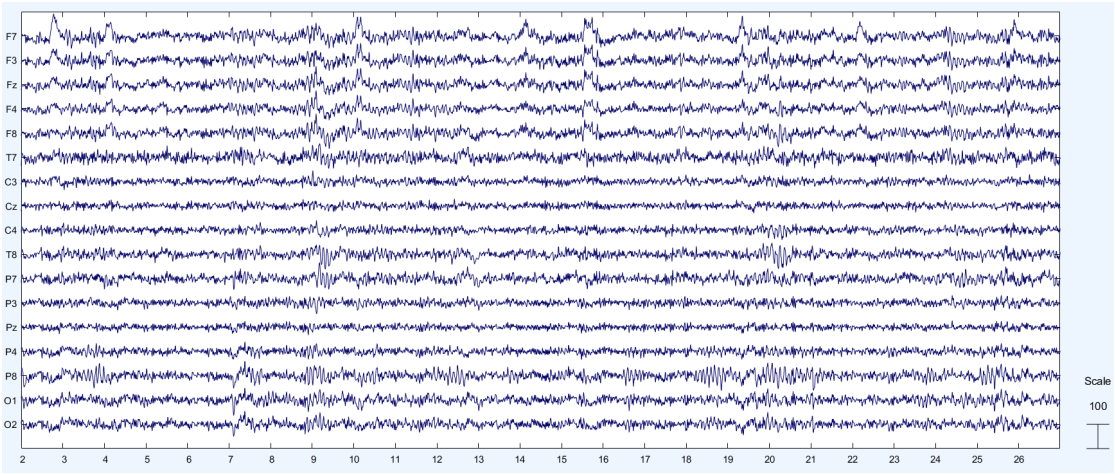


Figure 1.2: *EEG plots of one of the participants that took part in the experiment. There are in total 17 different EEG signals (channels) from different parts of the participant's scalp (location). The EEG is a time signal, which means the x-axis represents the time in seconds, while the y-axis tells how strong the brain activity at a specific time point is (in μV).*

The EEG signal is quite intricate and contains several frequencies that overlap with those from other areas of the human body. Therefore, very high or fast occurring spikes can occur, which is a sign of EOG or EMG artifacts. They are produced by the blinking

of the eye or by sudden movements of the body. Because of that, the signal analysis of the EEG data can be a quite challenging task, given its high non-stationarities and low signal-to-noise ratio of the EEG, which makes the classification task of the experiment prone to errors. There are some options on how to remove some of those artifacts, e.g. by using filters, ICA, ERPs or for biological artifacts try to blink and move as little as possible. However, getting rid of all of them is a very nerve-racking process.

1.3 Physiology of the human brain

The human brain is one of the most complex organs in our body. It consists of the grey and white matter. The brain is also responsible for the control of many functions, such as memory, emotion, touch, motor skills, vision, and a lot more. Together with the spinal cord, it builds the nervous system and can be structured into the cerebrum, cerebellum, and brainstem [3]. The greatest portion of the brain makes the cerebrum, which is divided into right and left hemispheres. This area of the brain performs a variety of tasks, including the interpretation of touch, vision, and hearing as well as communication, reasoning, emotions, learning, and fine motor control [3]. As indicated in Figure 1.3, each hemisphere contains then also four lobes: frontal, temporal, parietal, and occipital. Each one of them can then be divided into smaller areas, depending on its functionality [16]. For instance, the frontal lobe is responsible for communication, including speaking and writing (Broca's region), personality, behaviour and emotions [3]. On the other hand, language comprehension is a function of the temporal lobe, often referred to as Wernicke's region. Other functions of this lobe include memory, hearing, sequencing, and organizing [3]. It is noteworthy that lobes are often responsible for the same or similar functionalities. The reason for that is that the brain's lobes work together as a unit. Hence, the connections between the brain's lobes and its right and left hemispheres are extremely intricate.

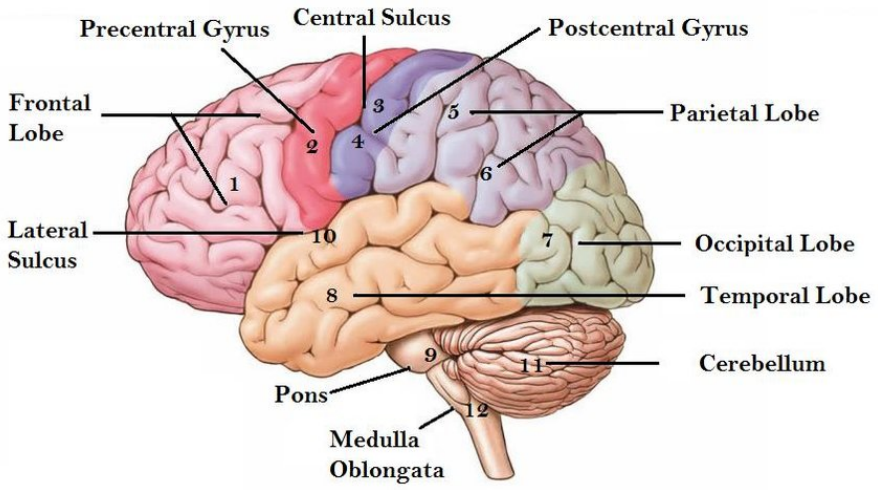


Figure 1.3: *Anatomy of the human brain with its major functional areas [16].*

The last part of the brain forms the cerebellum, which is located under the cerebrum. Its primary functions are to maintain balance, proper posture, muscular coordination, swallowing and more [3].
Now with the preceding explanation of the brain's structure, and activities in which it takes place in the human body, the perspective about the complexity of the detected signals from the brain, becomes clearer. Understanding these signals and categorizing them based on other people's voices is therefore a quite difficult and tricky task.

2 Method

The methods used for this bachelor thesis are similar to the ones from papers like *Rapid Brain Responses to Familiar vs. Unfamiliar Music* by Robert Jagiello et al., which focused on differentiating familiar and unfamiliar music in the EEG data or the paper from Julien Plante-Hébert about the processing of intimately familiar and unfamiliar voices. The experimental procedure has some exceptions that were done differently. An example is that the length of the stimuli is much longer in this experiment, also the stimuli all have different durations and are therefore not uniform. Another distinction is that in this work there was no use of event-related potentials. This means averaging over the trials was redundant, and instead the raw EEG data was used for classification. Also, while the study about music-evoked pleasantness [2] was only focused on extracting features in the theta region, the goal here was to extract features of all four sub frequency bands (delta, theta, alpha and beta), in order to subsequently evaluate which one of the sub frequency bands has the best correlation to the speakers.

2.1 Participants

For the purpose of this study, 5 male adult participants between the ages of 22 and 24 were selected. All of the participants were dominant right handed and showed good hearing abilities. None of the study participants had prior neurological or psychiatric conditions, like epilepsy or other diseases. Since all of the stories were in German, the only requirement they had to fulfill was to be fluent in German. The people who recorded the stories as audios were professors, family members, friends, colleagues, partners as well as complete strangers to the participants. The stories were recorded via a mobile phone. However, the participants were not allowed to read or listen to the elected stories prior testing.

2.2 Stimuli

The voice stimuli are recordings from commonly familiar German fables. Fables like *Der Rabe und der Fuchs von Jean de La Fontaine* or *Der Geizhals von Äsop* were used for this purpose (Table 2.1). However, there was a total of 20 stories and, since there were 4 different conditions to be examined, it was necessary to use 5 of the stories for one condition.
The stories have a duration between 1 and 2 minutes and before presenting them to the participants, they were trimmed to avoid too long intermissions at the start and end of each audio recording. Since every storyteller had a slightly different reading speed,

an uniform length of the recordings was not possible. That, however, did not create a problem, since event-related potentials were not of interest. Another crucial step was to normalize the loudness of these recordings, since the amplitude of the audio's varied. For this reason the amplitude was normalized to a Loudness Units Full Scale (LUFS) of -23, which is in accordance with the loudness guidelines for over the top television. This value basically measures the track's average loudness over time, adjusted for human hearing differences.

Table 2.1: *The table contains the names of the stories that were used for the stimuli as well as the names of the authors and the approximate duration. The origin of the stories is from a website for German fables:* `https://www.deutschland-lese.de/streifzuege/fabeln/`

Nr.	**Stories**	**Author**	**Duration in min**
1	Der Rabe und der Fuchs	Jean de La Fontaine	$\sim 1:00$
2	Stadtmaus und Feldmaus	Martin Luther	$\sim 1:20$
3	Fabel von dem Fuchs und der Katze	Hans Sachs	$\sim 1:30$
4	Der Zwerg und die Riesen	Florian Russi	$\sim 1:40$
5	Hundefreundschaft	Iwan A. Krylow	$\sim 1:20$
6	Der Alte und die drei Jungen	Jean de La Fontaine	$\sim 1:20$
7	Der Hahn und der Fuchs	Jean de La Fontaine	$\sim 1:00$
8	Die Auster und die zerstrittenen Pilger	Jean de La Fontaine	$\sim 1:20$
9	Die Grille und die Ameise	Jean de La Fontaine	$\sim 1:40$
10	Der Geizhals	Äsop	$\sim 1:00$
11	Die Kutsche und die Fliege	Jean de La Fontaine	$\sim 1:40$
12	Der Adler und der Fuchs	Äsop	$\sim 1:40$
13	Der Esel und die Ziege	Äsop	$\sim 1:30$
14	Warum nicht mich?	Florian Russi	$\sim 1:30$
15	Der Tanzbär	Christian F. Gellert	$\sim 1:00$
16	Die Milchfrau und der Milchtopf	Jean de La Fontaine	$\sim 1:20$
17	Die Katze und die beiden Spatzen	Jean de La Fontaine	$\sim 1:30$
18	Die beiden Pilger	Florian Russi	$\sim 1:00$
19	Die zwei Gorillas	Florian Russi	$\sim 1:30$
20	Der Grundstein und die Krähe	Florian Russi	$\sim 1:00$

2.3 Procedure

The presentation of the stories was as followed, each participant sat in front of a screen with over-ear headphones on. In order to avoid interference during the measurement, it was important that the over-ear headphones did not touch the electrodes on the cap. Also, before playing the actual audio of the stories, it was ensured that the volume was set to a pleasant level for the participant. A wireless mouse was also used, thus the participants could answer the questions after each story directly. After listening to the stories they had to complete a brief questionnaire, containing three questions, which was structured as described in Figure 2.1. The participants had to choose between one out of four answer possibilities, whereby the 4th one was optional and was only intended to be selected, if the participant wanted to skip a the question or if he could not remember the right answer. The participants were even encouraged to select this option, if they were unsure about the right answer, in order to avoid guessing the answers. During the whole audio presentation, there were 60 questions to answer in total. Per questionnaire the participants needed about 10 to 20 seconds to finish, which was about the half of the interval time. The layout for the questionnaire was obtained from GitHub and adjusted for this works purposes [19]. Also, in order to display the questions at a screen the *psychtoolbox* from matlab was used.

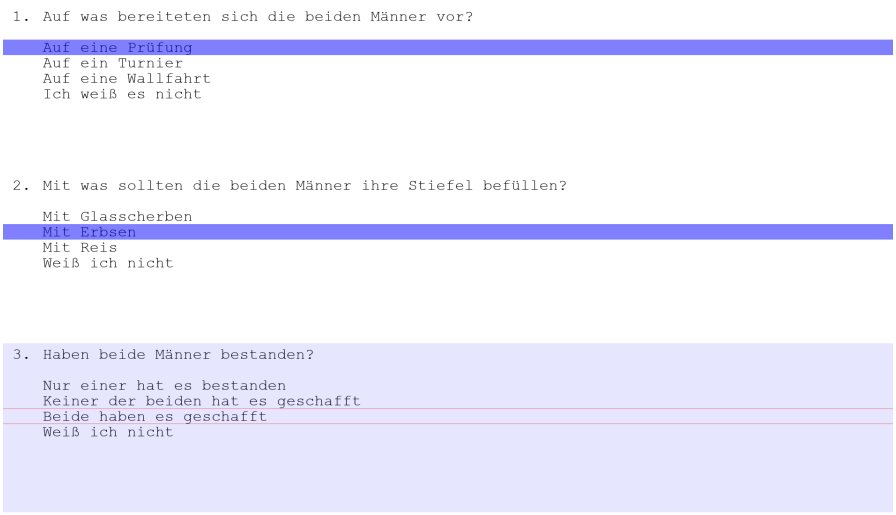


Figure 2.1: *The image depicts the layout that was used for presenting the questions at the screen, after each audio presentation. As visible, the selected answers got highlighted and later on saved as a .csv file locally.*

Participants were advised to keep during the audio their eyes open, blink as little as possible, and maintain a straight forward look onto a fixation cross at the screen. The experiment started with a 10 seconds *prerun* time. After that, the first story started playing, and would last approximately one to two minutes, depending on the individual story as well as the storyteller. In addition, the order of the stories and conditions was randomized, so the participants could not predict which person was about to speak.

Each participant had 40 seconds to complete this questionnaire before the next audio started. After the last story was played there was another 10 seconds pause (*prerun*) until the EEG recordings stopped. The whole audio presentation lasted around 41 minutes, with slight variations for each participant.

2.4 EEG acquisition and analysis

The EEG data was acquired in the lab of the Neural Engineering Institute of Graz with an ANT Neuro amplifier and by using a $64 - channel$ electrode cap. In this work only 19 channels, corresponding to the International 10-20 system were of importance: $FP1$, $FP2$, $F7$, $F3$, Fz, $F4$, $F8$, $T7$, $C3$, Cz, $C4$, $T8$, $P7$, $P3$, Pz, $P4$, $P8$, $O1$, $O2$. This can be seen in Figure 2.2. It should be noted, that the channel locations from the image correspond to the *MNI file* from EEGLAB, which contains the default channel location for electrode positions.
The ground (GND) electrode was put onto the forehead, while the data was referenced to *Cpz* and recorded with a sampling rate of 512 Hz. To keep the impedance below $1\,k\Omega$, an electrode gel was put between the electrode and the head of the participant. However, the channels *FP1* and *FP2* had to be removed from the dataset, since they produced a lot of channel noice and this caused a lot of trials getting removed.

The pre-processing part was done offline by using EEGLAB and Matlab. In order to extract the stories from the dataset, Matlab markers were used. The *prerun-marker* highlights the beginning of the measurement, while the *start- and end-marker* are responsible for labelling the start and end of the audio in the EEG data stream. To ensure a time synchronization between the EEG data streams and matlab markers, the Lab Streaming Layer (LSL) was used. Offline, the collected EEG data was then bandpass filtered with a high cut-off frequency of $40\,Hz$ and a low cut-off frequency of $1\,Hz$. To also get completely rid of the $50\,Hz$ noise, a notch filter was also applied between $48 - 52\,Hz$. Afterwards the data was resampled to $128\,Hz$, in order to reduce not only the computation time but also to decrease file read/write time as well as disk space usage for the later analysing steps. But before one could start with analysing the data a few preprocessing steps still needed to be done, like applying Independent Component Analysis (ICA), which is an approach for identifying hidden features in collections of random variables, measurements, or signals. It allows someone to remove/subtract artifacts embedded in the data, like EMG, EOG, Channel- as well as Line noise, without removing the affected data portions. This preprocessing step only works to a specific degree. For this reason, components that have been identified with an accuracy higher than 80%, were removed (see Figure 3.2).

After the data had been cleaned, the next logic step was to epoch the data and for that reason the part of the EEG data, where the participants listened to the audio, was extracted and subsequently cut into 8 seconds long segments. These so-called "segments" had an overlap of 4 seconds with other segments. Finally, it concluded a $3 - D$

tensor with the dimensions: *channels* × *time* × *trials*. Each trial contains 8 seconds worth of data points, which will be of use during the feature extraction process.

To avoid effects of surprise at the beginning and end of the stories, the first and last eight seconds were removed from the epochs. Every single one of these trials was then examined to see wether they fulfilled specific criteria. The first criteria is that the trials were not allowed to have a higher amplitude than $120\,\mu V$, otherwise they got rejected. This was also the reason for taking the channels *FP1* and *FP2* out from the study, since they were responsible for relatively high amplitudes in those trials. To catch other odd trials in the dataset, two measures were added: probability distribution and kurtosis. The oddness of a data trial is measured by its probability, for instance if the probability level is low, it implies that the activity levels in a data trial at a certain electrode are unexpected [8]. The goal is now to find these outlier trials by using the probability measure and setting a threshold in terms of a number of standard deviations (STD). The STD was set to 4, which means all the trials that had a normal distribution bigger than 4 were rejected. Furthermore, the purpose of the second measure was to determine the 'peakyness' of trials. This is the case if, for instance the trials contain strong transient muscle activity then the distribution of activation would be very peaky [8]. The threshold was once again set to a STD of 4.

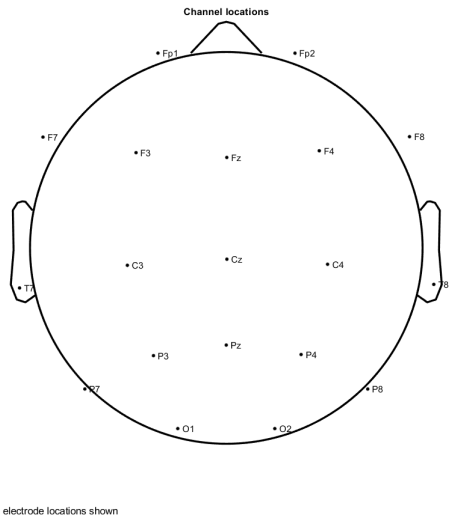


Figure 2.2: *The 19 used electrodes can be seen in this 2-D image. To keep in mind is that this are standardized position values and do not correspond to the exact values used in the study.*

2.4.1 Lab streaming Layer

The Lab Streaming Layer (LSL) is a system for collecting measurement time series in research experiments that manages both networking, time synchronization, (near-) real-

time access, and optionally centralized data collection, display, and disk recording. The LSL distribution includes the core library as well as a set of tools built on top of it. For this work, the Lab streaming Layer ensures a synchronization between the EEG data streams and the markers. This software solution saves the collected data of these different sources as an *.xdf* file, although the transport API does not recommend or supply a specific file format, but the provided recording tool (LabRecorder) returns this format [12].

2.4.2 Independent Component Analysis

ICA is a very powerful tool, besides using it for very known problems such as the cocktail party problem, one can also use its properties to detect and remove artifacts. The cocktail party problem is often explained in terms of audio recordings, where the goal is to isolate the sound of one specific person out of a stream of different speakers. In the context of EEG analysis, the result of an independent components analysis offers a set of weights for every electrode so that each component is a weighted sum of activity at all electrodes, and the weights are intended to separate the different components, like EMG or EOG [13]. In other words, it can be used to clean up EEG data by identifying components that isolate artifacts and then subtracting those components from the data.

To compute ICA components of the continuous EEG dataset, the *"Decompose data by ICA"* tool needs to be selected. This calls the function *pop_runica.m* to run ICA using the default options. For the ICA decomposition the Infomax algorithm (*runica*) from EEGLAB has been used.

One can either inspect ICA components by oneself and then remove them manually or apply an automated detection algorithm to get rid of of the artifactual ICA components. EEGLAB provides therefore already installed plugins, like *Classify components using ICLabel*. This automatically detects the 19 components in our EEG data and classifies them with an accuracy. There are six categories of components that can be classified: Brain, Muscle, Eye, Heart, Line Noise, Channel Noise, and Other. After all components have been classified, another plugin has to be selected to flag the unwanted components, which have been classified with more than 80% probability as artifacts. These marked components are then subtracted from the data.

2.5 Feature extraction

For the feature extraction part, the epoched and filtered data was used in order to extract specific information from every single trial. The two values that got extracted here are the *Power Spectral Density* (PSD) and *Phase-Synchronization* (PS). The PSD basically reflects the distribution of signal power over a specific frequency or a frequency range [13]. In order to calculate the PSD, the *Welch's* method was therefore used. On the other hand, the phase-synchronization value describes the consistency in phase difference between two signals (channels) over time, for a frequency band of interest [13]. In order to provide a comprehensive assessment of the synchronization between two signals, PS can also be calculated via successive frequency bands. The most popular PS measurement and the one used in this work is called *Inter-site phase clustering* (ISPC). Both have already been used for classification in studies before. In a recent study, where fronto-temporal theta phase-synchronization was analysed, in order to study music-evoked pleasantness, good results were obtained [2]. Another study used PSD values to analyse the specific EEG channels of a user, while they were performing a typing task with a laptop [21]. The promising results from the studies was criteria for selecting these features. Nevertheless, it needs to be borne in mind that this thesis focus was on solely the extraction of the features from the "raw" EEG data and not, however, the application of event-related potentials. If ERPs come into play, then one must consider that they are direct responses to a specific stimulus (event-locked). The trials have an invariable latency and shape. They are used to cancel out the random brain activity, by conducting many trials and then averaging it, so that only the relevant ones remain. The only downside of that, is that in order to measure it correctly and accurately, many trials are required and the stimuli need to be uniform. Also, if one intends to use ERPs to draw conclusions about cognitive processes, they should understand the problems with component overlap, component quantification, acceptable interpretation, and statistical methods [13].

2.5.1 Phase-Synchronization

As mentioned above, one of the features used in this work is the ISPC or PS, but in order to understand what this means, one has to get familiar with the Intertrial Phase Clustering (ITPC) first. The definition by Cohen [13] describes the value quite good: "ITPC measures the extent to which a distribution of phase angles at each time-frequency-electrode point across trials is nonuniformly distributed in polar space." [13]. Since phase values are circular (=measured in radians), one cannot simply average the values in the same way as power or voltage is averaged [13]. This problem can be bypassed if phase angles are represented as vectors with unit length on a circle, which can be described mathematically with the Euler's formula (e^{ik}). Likewise, it must be mentioned that these phase angles are obtained by a point in complex space resulting from the convolution between a complex Morlet wavelet and the EEG data itself. The ITPC is now measured by taking the average of these phase vectors. This averaged vector is always smaller than the unit length vector of the phase angle. The basic thumb rule

here is, the further apart the phase angle vectors are, the smaller the result vector is and by taking the length of that result vector (averaged vector) one basically gets the ITPC. The values are set between 0 and 1, whereas zero describes completely uniformly distributed phase angles and 1 depicts completely identical phase angles.

As mentioned above, this leads to the actual value used in this thesis, the ISPC. The two of them are very similar, but with the difference that instead of averaging phase angles, ISPC takes the average of the variations in phase angles between two electrodes over time [2]. To do so, one has to start by computing the time-frequency decomposition on each trial, using complex Morlet wavelets with defined frequency bands (see Figure 2.3). From this decomposition, the phase values for each electrode and frequency were obtained over time and then used to calculate the ISPC by using the following formula:

$$ISPC_f = \left| n^{-1} \sum_{t=1}^{n} e^{i(\varphi_i - \varphi_j)} \right| \tag{2.1}$$

whereby φ_i and φ_j are the phases of two given signals at a given time point, n is the total number of time points, and f is the current frequency. At the end, this was calculated for all four frequency bands and for every possible channel combination [2].

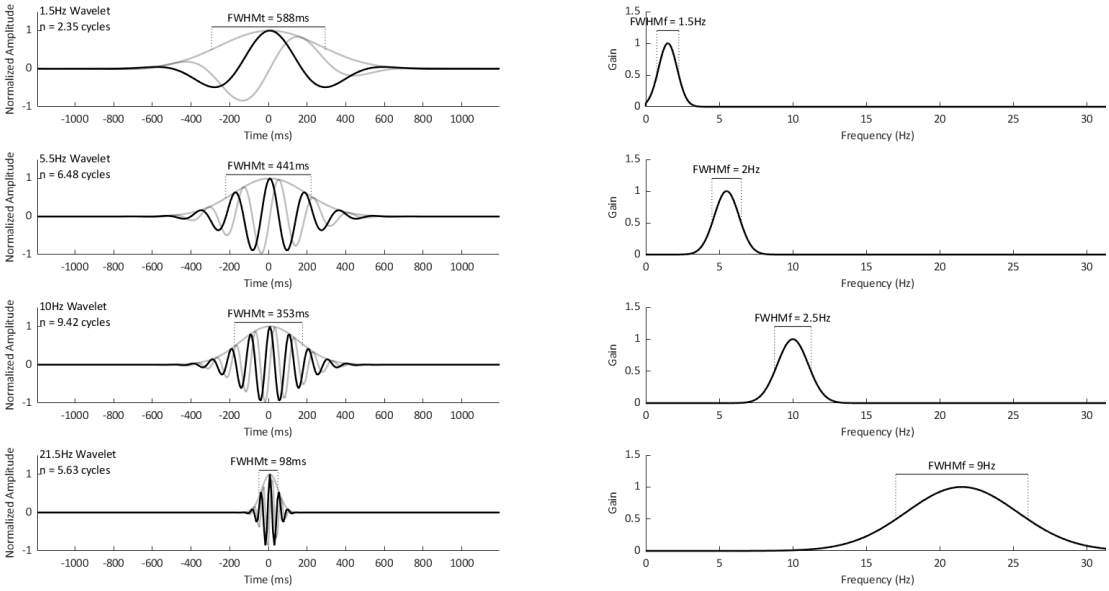


Figure 2.3: *These user defined complex Morlet wavelets were used for time-frequency decomposition. Since the goal was to inclose all four frequency bands, only the frequency component was of interest (right side). FWHMf was always set in such a way, so the whole sub frequency range was covered, which also explains why the wavelets all have a different number of cycles.*

2.5.2 Power Spectral Density

The second feature used as a feature extraction method in this work was the power spectral density (PSD). This value is an essential representation of the signal spectrum that depicts the power measurement content over a frequency. PSD is typically used to characterize broadband random signals and has a variety of usages in all kinds of research areas [17]. There are two appraoches on obtaining the PSD, one of them is by using parametric and the other one by using non-parametric methods. The moving-average model and the autoregressive model are two instances of parametric techniques. While the non-parametric methods are based on Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT) or Welch's method [17].
Welch's method involves breaking the time signal into sequential blocks, creating the periodogram for each block, and then averaging it to estimate the power spectra [21]. This overlapping aids in reducing information loss due to tapering and displays a more dependable periodogram. Each block is subjected to the Fourier transform calculation, and the result is then squared. This result serves as the foundation for the PSD computation and by averaging all the periodogram results from all the blocks, one gets the PSD estimate. Mathematically this can be expressed as followed, where first the DFT is calculated for each window:

$$S_i[\vartheta] = \sum_{m=1}^{M} s_1[m] \cdot w[m] \cdot e^{\frac{2\pi j m \vartheta}{N_f}} \tag{2.2}$$

with w as the window vector, s_1 is the first interval/block of the time signal, and N_f the DFT size. Afterwards the periodogram values are calculated by the squared value of the absolute DFT samples:

$$P_i[\vartheta] = \frac{1}{C}\left|S_i[\vartheta]\right|^2 \tag{2.3}$$

where C is just the normalization factor over the windowing vector w. The last step that is left is averaging the periodogram values. This leads to the PSD for a certain frequency:

$$PSD(\vartheta) = \frac{1}{K}\sum_{i=1}^{K} P_i[\vartheta] \tag{2.4}$$

Estimating the power of the signal throughout the whole signal period with the Welch's method leads to a more smoothed frequency spectrum [21].

2.6 Feature selection method

After all the features have been extracted from the EEG data, the next step was to concatenate all these feature values from a trial to one feature vector. This allows one to label the trials to a specific class. Combining all these values results in a vector with a size of 612 feature values. Since using all of them would not yield a good classification result, it is necessary to apply a feature selection method, which is a process of selecting a subset of preliminary features [10]. In fact, by eliminating pointless and unnecessary features, the dimensionality decreases. The classifier benefits from this, because not only does the performance of the prediction improve, but also the classifier works faster and is more cost-effective [10].
In general, feature selection techniques fall into one of three families: filter-based, wrapper-based, or embedding methods [10]. The ones used in this thesis are filter-based methods, also known as classifier-independent approaches since they use some predetermined assessment criteria to choose the optimal subset of features. To note, this method is not reliant on the learning process or classifier, while on the other site, wrappers are classifier-specific, which means they use the learning algorithm as an evaluation function to look for the optimal collection of features [7].

The selection of features using a filter-based method is typically reduced to a binary choice that maximizes a certain performance requirement [10]. While there exist various performance criteria, the one selected for the purposes in here, is the the Fisher score. It is one of the most widely used criteria, because of its good performance. The whole process of finding the best features is actually a combinatorial optimization problem, and finding the global optimal solution for that is NP hard [10]. Solving this problem requires a lot of computational effort, but one method which approaches this problem is the heuristic algorithm. It works by calculating each feature's score, in accordance with the Fisher criteria and then selecting the top ranked features with the highest scores. The main goal of the Fisher score is to identify a subset of features such that the distances between data points belonging to different classes in the data space are as great as possible, whereas the distances between data points belonging to the same class are as small as possible [10]. By considering this, one can then compute the Fisher score with the given training vectors x_k and the positive n_+ and negative n_- instances (two classes) as followed [7]:

$$F(i) = \frac{(\bar{\mathbf{x}}_i^+ - \bar{\mathbf{x}}_i)^2 + (\bar{\mathbf{x}}_i^- - \bar{\mathbf{x}}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^+ - \bar{\mathbf{x}}_i^+)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^- - \bar{\mathbf{x}}_i^-)^2} \tag{2.5}$$

where $x_{k,i}^+$ and $x_{k,i}^-$ are the ith feature of the kth positiv or negative instance. $\bar{\mathbf{x}}_i^+$ and $\bar{\mathbf{x}}_i^-$ are the averages of the ith feature of the overall data sets of the two instances. The greater the Fisher score is, the greater the likelihood that this feature is more discriminative [7].

2.7 Classifier

Classification is an important instrument when it comes to setting apart various classes. A classifier in data science is a type of machine learning algorithm used to categorize data input [9]. To train the classifier one has to use labeled data, whereby in this case the data is represented by the calculated features. The classifier takes training data (the features from trials) and assigns them to a certain class. After the training process, it is possible to give unlabeled data to the trained classifier, which will then output the classified labels of the input data. The way it works is by making predictions about the chance of a data input being categorized in a particular class [4]. Therefore, classifier algorithms make use of complex mathematical and statistical techniques and each classifier has its own method to make the prediction. In this work, the classifier statistically predicts whether a trial of a story is likely to be from a friend, a family member, a teacher or an unknown person. There is only a binary classification, which means out of the 4 datasets, only two conditions at the time are being classified. Because of that the classifier has to run 6 times for each participant, since there are 6 combinational possibilities to create 2 pairs from these 4 conditions.

By using different classifer algorithms it becomes more visible whether there are big variations in the results and it is possible to determine which one of the classifiers performs the best. Hence, the following three classifiers were used: Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), k-Nearest-Neighbours (kNN). All of them followed the same learning pattern and used the same data. The whole dataset was split by the 80/20 rule, where 80% of the dataset is training data and 20% test data. In other metrics, for each condition, one can assign 4 stories to the training set and the fifth story to the test set. Important to mention is also that every story in the condition has been considered as test data, which means each classier performed additional 5 times. The results from these five runs were then averaged and then represented as the classification result of the respective classifier.

2.7.1 Support Vector Machine (SVM)

SVM are a promising tool for data classification as well as for regression and fall under the category of generalized linear classification. They only use a subset of data samples for classification, these samples are also known as "support vectors". Another property of SVMs is that they are able to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary [9].

The basic idea behind SVMs is to map data onto a higher dimensional space and determine the optimal decision boundary -also known as hyperplane- by maximizing the margin between both classes. The margin refers here to the distance between the closest data points to the decision boundary, which is also the reason why they are sometimes called Maximum Margin Classifiers [9]. In order to map training data into a higher dimensional space, kernal functions come into play. There is variety of kernel functions

out there, but the one used in this work is the Radial Basis function (RBF) kernel, which is a Gaussian without normalization [4]:

$$k(\mathbf{x}, \mathbf{x'}) = exp(-\gamma||\mathbf{x} - \mathbf{x'}||^2) \tag{2.6}$$

RBF has an especially good perfomance, when it comes to non-linearly separable data, which is one of the reasons why they were selected for the classification task. The other reason is that they are able to penalize wrong data points on the wrong side of the hyperplane, which allows one to create a much better decision boundary.

2.7.2 Linear Discriminant Analysis (LDA)

Compared to SVM, LDA works exactly the other way around. Instead of mapping the data into a higher dimension, like the SVM algorithm does, it instead projects the data into a space of lower dimension M ($M < D$) [4]. This method is very similar to Principal Component Analysis (PCA), but with the difference that LDA is a supervized technique and PCA not, which means that LDA takes the class labels of the feature vectors into account. This allows one to do classification tasks.

The useful information in the data is represented in the variance of the data samples [4]. Therefore, one needs a projection vector $\mathbf{u}$ that accounts for as much of the variability in the data as possible after projection, while simultaneously also maintaining the information of the classes [4]. To do so one has to select a projection vector $\mathbf{u}$ in such a way that the distance between the two class means of the projected data is maximized (between-class covariance matrix), but on the other hand, minimize the covariance of the data within a particular class (within-class covariance). This ratio is also known as the Fisher criterion:

$$J(\mathbf{u}) = \frac{\mathbf{u}^T \sum_b \mathbf{u}}{\mathbf{u}^T \sum_w \mathbf{u}} \tag{2.7}$$

where $\sum_b$ is the between-class covariance matrix and $\sum_w$ the within-class covariance. Putting it all together, the goal is to maximize the ratio $J(\mathbf{u})$:

$$\mathbf{u} = \underset{u}{argmax}\, J(\mathbf{u}) \tag{2.8}$$

Solving it results into a generalized eigenvalue/eigenvector problem, and by taking the eigenvectors of $\mathbf{u}$ that match to the largest eigenvalues λ, it is possible to project the data onto a lower dimensional space:

$$y = \mathbf{u}^T X \tag{2.9}$$

2.7.3 k-Nearest-Neighbours (kNN)

The k-Nearest-Neighbours algorithm is compared to the other two classifier a relative simple, but also effective classification method. It takes k many neighbours of the trainig data points into consideration in order to determine the class/label of the test data. If

we now want to classify a specific data point in the data, the algorithm looks for the k nearest neighbours in this dataset area and then decides (based on the distance to the neighbours) to which class it belongs to. In order to use kNN, it is mandatory to select a proper value for k, since the classification's performance is highly reliant on this number. Taking a too small value leads to disrupted clusters with many small regions, while for too large values of k the classification performance suffers [4]. Running the algorithm several times with various k values and selecting the one with the greatest performance is a pretty simple way to determine a suitable k value. This was also the works procedure for selecting the k value, which in the end was 5.

2.7.4 Chance accuracy

The chance accuracy can be depicted as the performance baseline for a classification algorithm [5]. An algorithm that achieves a classification accuracy below chance level has no skill on the dataset, whereas an algorithm that achieves a score above that has some skill on the dataset [5]. Imagine having a random performing classifier and a dataset with a balanced distribution of 0s and 1s. In this scenario, the classifier's so-called chance accuracy would be 50%. The baseline accuracy of a classifier really depends on the class distribution, so whether it is balanced or imbalanced. If it is imbalanced then one class has more trials than the other, which means that it cannot be simply said that the chance level in a simple 2-class paradigm is exactly 50% [14]. Therefore, two methods are presented, that were used to calculate the chance accuracy.

The first method makes use of a naive classification model, which is based on the distribution of the dataset. In order to calculate the chance accuracy, the number of trials in each class has to be determined. For instance, if one class has 50 trials and another one has 40 trials, one takes the majority as the nominator and the sum of both to define the denominator:

$$chance\ \ accuracy = 100 \cdot \frac{50}{40 + 50} = 55.55\,\% \tag{2.10}$$

which results then in a chance accuracy of 55.55 %.
The second method is completely different from the first one. It is based on the approach depicted in the paper of Mueller-Putz *"Better than random: a closer look on BCI results"* [14]. The basic idea behind it is to make use of the confidence interval (with a certain significance) for the accuracy obtained with a random classifier. The procedure looks like followed, where first a fake set of labels from the two classes is created. The size of the labeled dataset depends on the number of trials in these two classes. The first half of the dataset should contain the labels of one class (e.g. 0), and the other half the labels of the second class (e.g. 1). Then by randomly shuffling the labels N times, with N being 10000 in this case, it is possible to create a distribution of random accuracies. The accuracy is calculated after each shuffle as followed:

$$accuracy = \frac{Number\ \ of\ \ correct\ \ predicitions}{Total\ \ number\ \ of\ \ predicitions} \tag{2.11}$$

Finally, one takes the $alpha = 5\%$ upper confidence interval of the chance level, which means nothing more than taking the 95th percentile of the distribution. By doing so, one gets left with the chance accuracy value for a specific data distribution.

2.8 t-SNE

The t-Distributed Stochastic Neighbor Embedding or just t-SNE is a technique to visualize high dimensional data, in our case in form of a scatter plot in a two-dimensional space. It is just like PCA, but with the difference that t-SNE is an unsupervised, non-linear dimension reduction method, while PCA only a linear technique [20].

The initial step of stochastic neighbor embedding is to translate the high-dimensional euclidean distances among data samples into conditional probabilities that indicate how similar the points in the high dimensional space are [20]. If data points are nearby it means a higher probability, while widely separated data points mean a lower conditional probability. The following step is nearly identical to the previous one, except that this time instead of using a Gaussian distribution, one uses a Student t-distribution with one degree of freedom (Cauchy distribution) [20]. As a result, a second set of probabilities in the small-dimensional space has been created. The next step is to ensure that these probabilities from the low-dimensional space match, as closely as possible those from the high-dimensional space and this is done by using Kullback-Liebler divergence (KL). The exact procedure of KL is not of importance, however, the only thing of relevance is the fact that it is an asymmetrical method that tries to efficiently compare the two probability sets. In the end, gradient descent is applied to minimize the KL cost function [20].

In conclusion, it must be kept in mind that since t-SNE is not a clustering method, the values might frequently change in between runs. That is also the reason why it cannot be used for clustering and is only intended for experimentation and visual interpretation of the data.

3 Results

This section contains all the results from the described procedures and used methods from the chapter 2 above. In here the results are being compared and depicted with the help of various tables and figures. The interpretation and discussion of these results is then in chapter 4.

3.1 Questionnaire

The answered questionnaire of each participant was saved in a text file and then evaluated. Meaning that the right, wrong, and skipped questions from each participant were counted.
The distribution of right and wrong selected answers can be seen in Figure 3.1, in which also the number of skipped questions is depicted. Most participants made only one mistake and/or skipped one of the questions. To add here is also that nearly all of the participants made mistakes on different types of questions and not the same one.

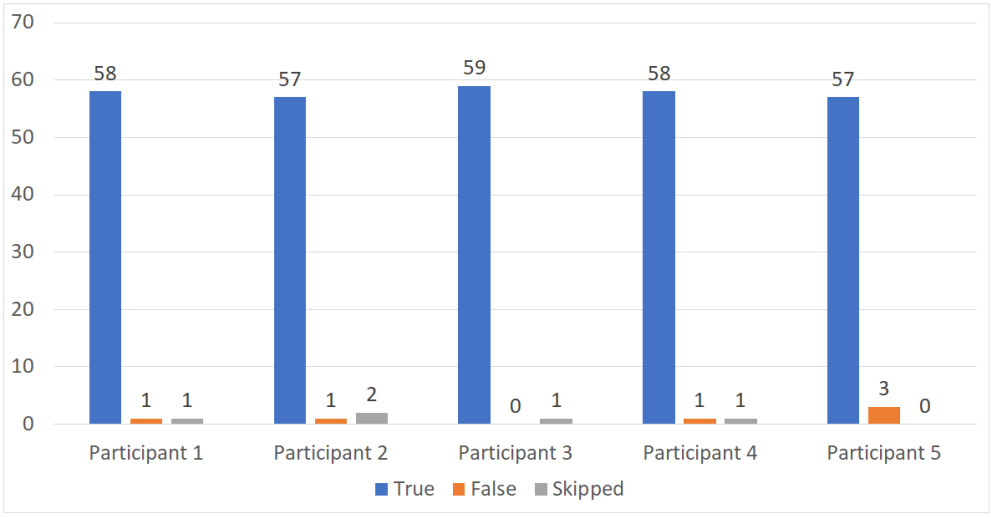


Figure 3.1: *Comparison of the numbers of right (blue) and wrong (orange) answered questions, as well as on how often the participant skipped a question (grey).*

The mouse button marker was used to find out how long each participant, on average, took to answer the questionnaire. For that reason, the mouse button- and the start markers from the dataset were extracted. The first thing to do was to find the time difference between the start marker and the nearest mouse button marker. Doing so made

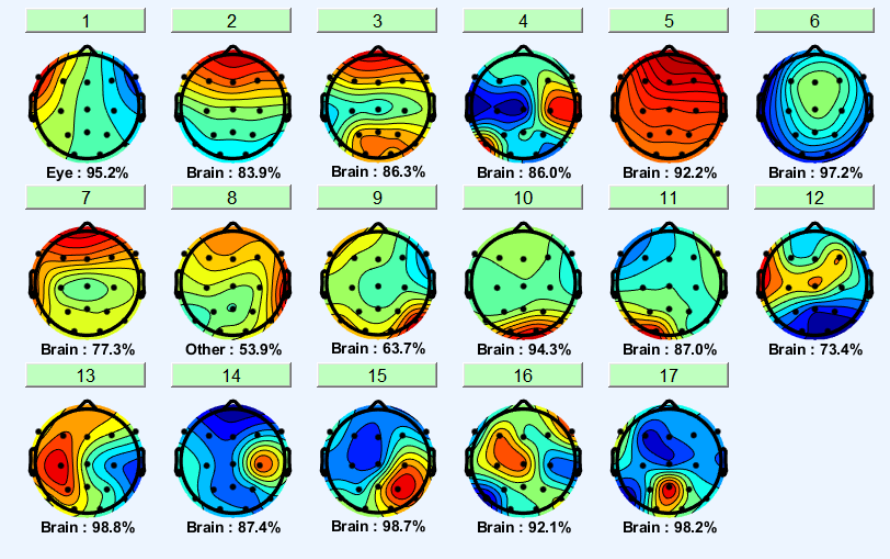
it possible to determine how much time needed to pass until the next story started. Also, since the interval time between the audios was known (40 seconds), it was possible to calculate the time on how long each participant took for each questionnaire. The only thing to do was to subtract the prior estimated time difference from the 40 seconds.
The downfall of this method is that one could only calculate the time difference between the first 19 stories. The reason for that is that the end marker, which indicates when the whole experiment ends, was set too near to the end marker of the last story so that in the end, it was not possible to determine when the last mouse button was pressed after the very last questionnaire.
But despite that, there are still 19 values left, which was more than enough to estimate the average answering speed.

Table 3.1: *Averaged answering speed over the first 19 questionnaires. The average time over all the participant is also calculated.*

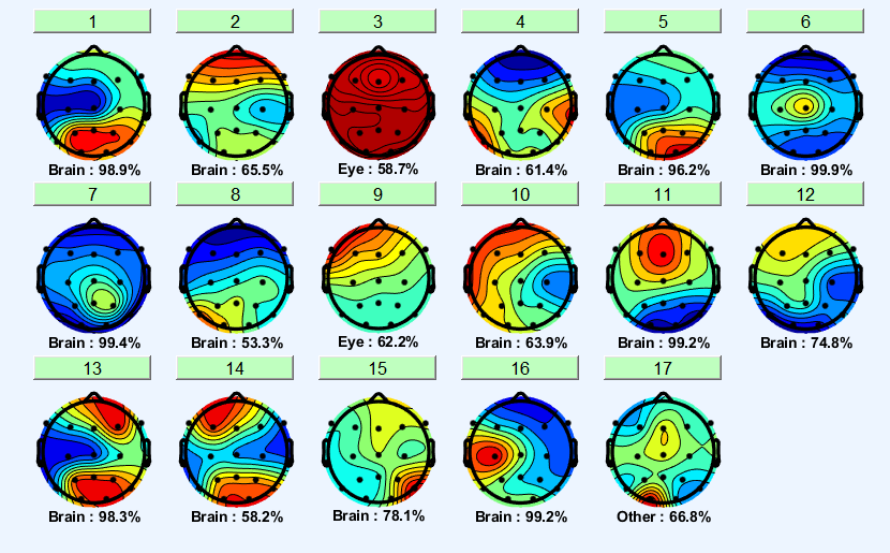
	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	**mean value**
answering speed (in seconds)	17.13	7.86	14.77	15.08	13.23	**13.61**

3.2 ICA components

To remove artifacts like EOG, EMG, line- and channel noise embedded in the data while simultaneously not removing the actual data portions, ICA comes in very handy. The topography plots of each component after ICA decomposition and labeling can be seen in Figure 3.2. The reason for only having 17 and not 19 components is due to the fact that in a previous step, the two channels, *FP1* and *FP2*, had to be removed from the dataset because of high channel noise.
However, only one component was actually removed from the two participants. For the participant in Figure 3.2 (a), it was an EOG artifact. For the other one (b), there was none since the classification accuracy of the artifacts was just too low and therefore not enough to get them removed from the dataset. The other participants had very similar results, with one or two artifacts more than the ones shown below.



(a)



(b)

Figure 3.2: *ICA component plots after ICA decomposition and ICA labeling from the second and fourth participants. (a) For the second participant, only one EOG artifact was detected and later removed. (b) The 4th participant had two EOGs and some other artifacts, but they were only detected with a certainty of around* 60%*, which was not enough to get them deleted.*

3.3 Classification datasets

With the help of t-SNE plots, it is possible to visualize higher dimensional data in the form of 2-D plots and then investigate, learn, or evaluate the segmentation of the data. t-SNE allows us to assign each label with its features to a specific coordinate in an image. The following t-SNE plots of the participants already tell how good or bad the performance of the individual classifier will be. Just by looking at the images, it stands out that for some condition pairs, the data is nicely clustered into two sections, whereby for other conditions, the data points are somehow distributed throughout the image. To illustrate the visual differences between datasets, two very opposite t-SNE plots of the data are displayed below. The first one is from the best-looking dataset (Figure 3.3), and the second one contains the poorest data distribution (Figure 3.4).

Figure 3.3, with its best-looking datasets, belongs to the second participant. What especially catches the eye here is that the first row has the cleanest segmentation between the two classes/conditions, except for some outliers. In addition, the first row only contains familiar conditions, and every time the unfamilar one is involved, the dataset gets less clustered. The good segmentation from earlier disappears, and the dataset (blue and red dots) is distributed somehow through the image, as seen in the lower rows of Figure 3.3.

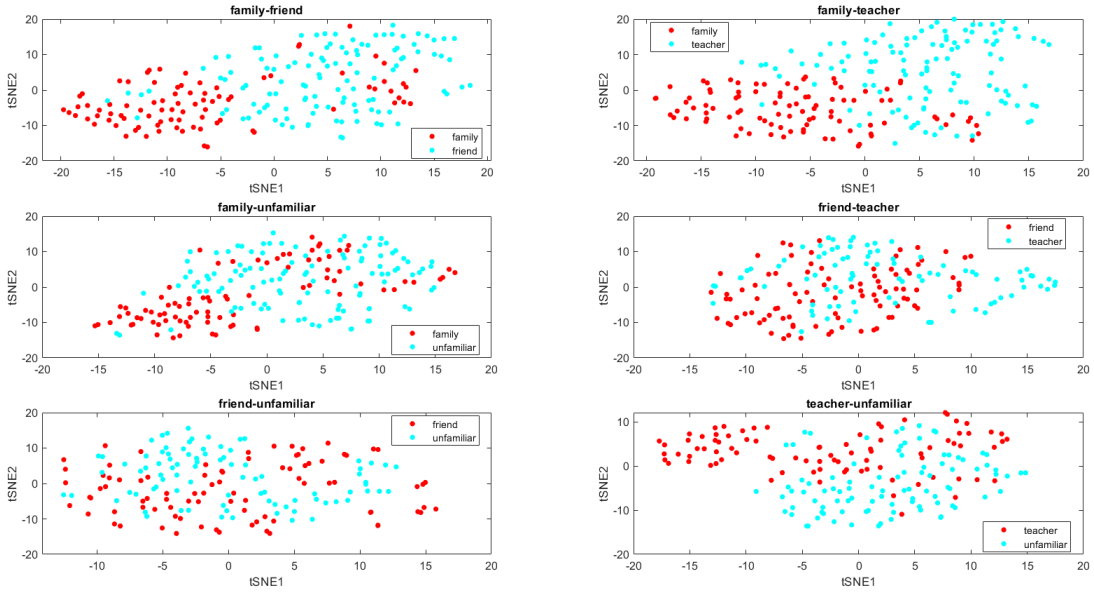


Figure 3.3: *Visualised data from Participant 2. Every possible combination of conditions has been considered and plotted. The dots represent the labels or the two different classes that are going to be classified.*

The dataset with the poorest distribution was from the 4th participant, and compared to Figure 3.3, the data now looks much more chaotic. It is hard to tell whether there

will be good classification results since the amount of outliers is much higher than in Figure 3.3. However, there are still some areas where one can detect some clusters of red or blue points, which gives hope of getting at least some good classification results.

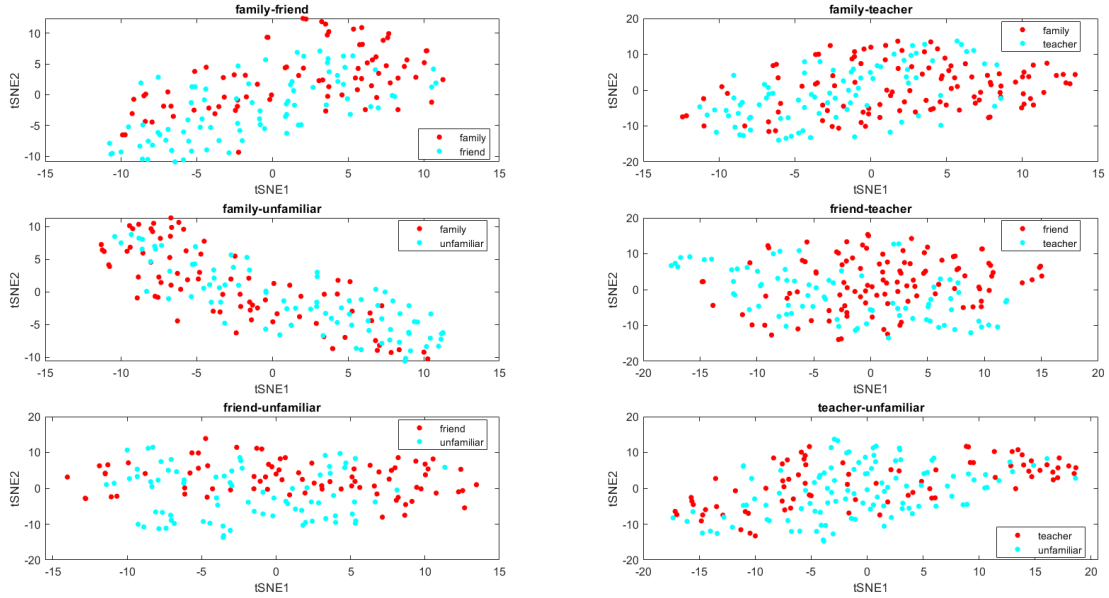


Figure 3.4: *Visual representation of the data from Participant 4 with every possible combination of conditions.*

3.4 Chance level

As mentioned in the previous chapter, two different approaches have been used for calculating the chance accuracy. The first one was by using a naive classification model, where the focus was set on the majority class and used to calculate the chance accuracy according to formula 2.10. Since the distribution is not perfectly balanced between two classes, it is therefore not possible to get a chance accuracy of precisely 50%. Instead, it varies between 50% to 64%, as Table 3.2 shows. The higher the chance accuracy is, the more unbalanced the two classes are. This is especially the case between the two conditions, friend and unfamiliar, where the chance is always around 60%.

On the other hand, the upper confidence limit of a chance result for a 2-class classification problem is closer to the theoretical 50% than the previous method. There is only a small variance of 4% to the 50%. The highest chance level here reaches a value of 54%, while the other method got higher values with over 60%. What stays the same are the positions in the tables, where the chance accuracy rises. If a value in table 3.2 is high, so is also the same cells in table 3.3. The table 3.3 still has most of the chance level results for most participants and condition pairs, around 50%.

Table 3.2: *The classification results correspond to the probability of a naive classifier, that always predicts the class with more likelihood.*

Participant	**chance accuracy**					
	family-friend	**family-teacher**	**family-unfamiliar**	**friend-teacher**	**friend-unfamiliar**	**teacher-unfamiliar**
1	52,50%	60,00%	62,56%	57,57%	60,18%	52,69%
2	57,14%	55,23%	50,87%	51,93%	56,28%	54,36%
3	56,59%	50,70%	55,02%	57,30%	61,46%	54,31%
4	51,55%	56,31%	58,91%	57,83%	60,40%	52,65%
5	64,96%	65,38%	58,28%	50,45%	57,03%	57,48%

Table 3.3: *Results from the upper confidence interval for the accuracy of a random classifier with* $alpha = 0.05$.

Participant	**upper confidence interval** ($alpha = 5\%$)					
	family-friend	**family-teacher**	**family-unfamiliar**	**friend-teacher**	**friend-unfamiliar**	**teacher-unfamiliar**
1	50,15%	52,01%	53,17%	51,18%	52,08%	50,17%
2	51,07%	50,57%	50,03%	50,09%	50,80%	50,42%
3	50,91%	50,04%	50,52%	51,06%	52,64%	50,40%
4	50,05%	50,83%	51,61%	51,27%	52,17%	50,16%
5	54,49%	54,74%	51,40%	50,08%	51,02%	51,13%

3.5 Accuracies of the classifiers

There are three different classification algorithms to determine how good the distinction between the four conditions is. All of them use completely different approaches to get accuracy results. Using these different classifiers makes it possible to tell the best performing one and see whether there are significant differences in the results. For instance, the SVM and kNN classifier performed better than the LDA one, but with the overall performance still being very similar. The SVM and kNN reached only half of the accuracy results higher than the chance level. The kNN classifier, therefore, achieved more often accuracy results higher than the chance accuracy from Table 3.2. In fact, kNN is the only classifier containing 10 values over chance accuracies (from Table 3.2).

The family-friend condition pair and family-teacher are the best condition pairs with the highest mean accuracies. Unfortunately, it was not possible to get the accuracies for the rest of the pairs much higher or even over the chance level. Those who were higher are marked in the tables below, with green meaning that the accuracy is higher than chance level from Table 3.2 and blue being higher than chance accuracy from Table 3.3. Of course, there are some exceptions where it was possible to obtain results as high as 76%. This is noticeable for the second and fifth participant, who received relatively good results in general (see table 3.4). Another thing that catches the eye is that these "high"

values are especially visible when trying to differentiate between family and friend or family and teacher speakers, so basically in the region of familiar voices.

Table 3.4: *Accuracy results from the SVM classifier and the mean accuracy from condition pairs. Blue marked cells indicate that the accuracy is higher than the upper confidence interval from table 3.3. Green highlighted cells mean that the accuracy is higher than the chance accuracy from table 3.2.*

Participant	family-friend	family-teacher	family-unfamiliar	friend-teacher	friend-unfamiliar	teacher-unfamiliar
1	48,82%	53,27%	50,01%	54,48%	52,99%	49,69%
2	**70,92%**	**76,53%**	51,47%	59,97%	36,68%	**57,66%**
3	57,19%	51,09%	44,06%	41,92%	**59,59%**	35,97%
4	49,34%	47,31%	38,19%	43,69%	42,45%	43,40%
5	57,45%	74, 73%	**60,07%**	**63,24%**	49,67%	34,71%
mean accuracy:	56,75%	60,59%	48,77%	52,66%	48,28%	44,29%

Nevertheless, half of the accuracy results still stay under chance accuracy, especially for the 4th participant, where as good as no accuracy values got over chance level, only in table 3.6 it barely reaches the threshold.
Another thing that stands out is that low accuracies are obtained whenever an unfamiliar condition is involved. Then, the mean values of each condition pair only reach about 40%. In comparison, the average for the familiar condition pairs attains up to 60.59% accuracy.

Table 3.5: *The accuracy results of the LDA classifier, between different conditions and with mean accuracy results from the condition pairs. The highest occurring value in a condition pair is always bold.*

Participant	family-friend	family-teacher	family-unfamiliar	friend-teacher	friend-unfamiliar	teacher-unfamiliar
1	46,98%	51,63%	**62,82%**	**56,16%**	52,02%	52,94%
2	**66,20%**	64,63%	55,48%	51,95%	**57,93%**	50,47%
3	59,09%	44,79%	42,82%	48,07%	54,02%	41,35%
4	44,31%	48,97%	39,58%	45,88%	51,19%	41,54%
5	51,05%	**67,75%**	49,91%	47,03%	41,20%	**55,52%**
mean accuracy:	53,53%	55,56%	50,13%	51,97%	52,28%	47,42%

On the other hand, the LDA classifier has the lowest performance out of the three, with only 13 values being higher than chance accuracy. There are some other differences too. For instance, a few condition pair values from the previous table that were much lower for some participants are now higher (and vice versa) and have an increase or decrease of 20%. These changes in accuracy can be seen especially on the right-hand site of the tables (last four condition pairs). Despite these differences, the overall performance is still very similar, which confirms the value of the mean accuracies over the participants since it only changes a few percent.
The remaining classification results from the kNN classifier are visible in table 3.6. It

achieved slightly better overall accuracy results than the SVM algorithm, with the best results being between the familiar conditions. On the other hand, the accuracies from the SVM have more isolated cases, where the accuracy reaches more than 70%. In fact, 3 cases reach over 70% with the SVM, while only one gets a value of 73.85% with the kNN classifier. The only classifier that steps out of line here is the LDA classifier because no value got over 70%. The highest accuracy reaches only a value of 67.75%. Also, the differences between the familiar condition pairs and unfamiliar ones are not as significant as with the other two classifiers, since the mean accuracy only reaches 55%.

Table 3.6: *Accuracy results of the kNN algortihm with mean accuracy results from the condition pairs. The highest occurring values are again bold.*

Participant	**family-friend**	**family-teacher**	**family-unfamiliar**	**friend-teacher**	**friend-unfamiliar**	**teacher-unfamiliar**
1	54,97%	54,34%	49,26%	48,00%	55,12%	48,91%
2	**68,25%**	**73,85%**	48,20%	56,99%	34,16%	**59,74%**
3	64,87%	52,02%	47,62%	46,38%	**57,05%**	36,80%
4	51,01%	49,52%	41,24%	46,26%	40,27%	47,04%
5	59,34%	67,71%	**63,44%**	**66,04%**	49,89%	41,67%
mean accuracy:	59,69%	59,49%	49,95%	52,74%	47,29%	46,83%

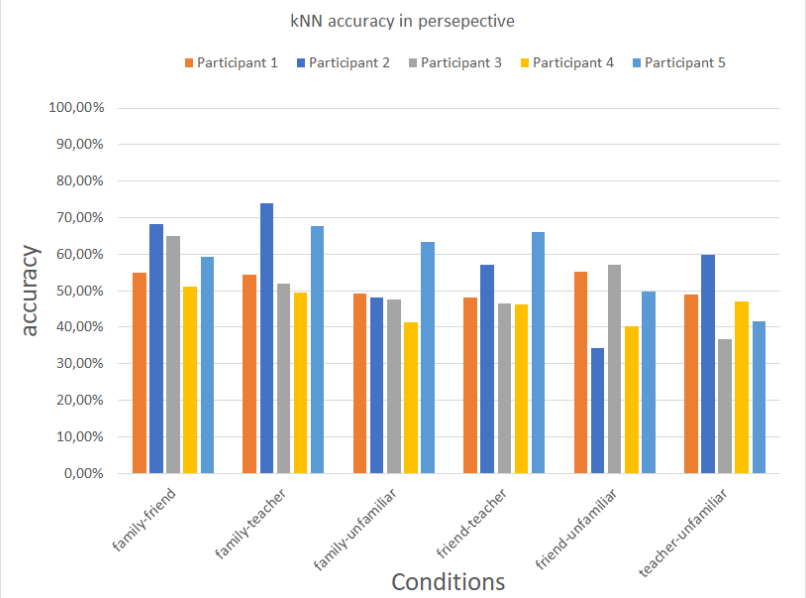


Figure 3.5: *Plot of all the accuracy results from table 3.6*

When putting all the accuracy results of the kNN classifier into perspective, it becomes much more apparent how high and good the classifier's performance is (see Figure 3.5). The differences between the condition pairs are now more visible, especially when trying

to distinguish between the unfamiliar condition and the teacher condition, then shallow accuracy results are obtained. Also, these high-occurring accuracies that reach over 70% can be seen better now and put into contrast to other condition pairs.

3.6 Frequency bands

Since only the 100 features with the highest Fisher score have been selected for classification, it was possible to determine which frequency band dominated during which classification problem. To find out how often the features from a specific frequency band occurred, it was necessary first to label all of them and then only extract the ones that belonged to this frequency band. The whole process was done for every participant, and the amounts were summed up for each classification task.

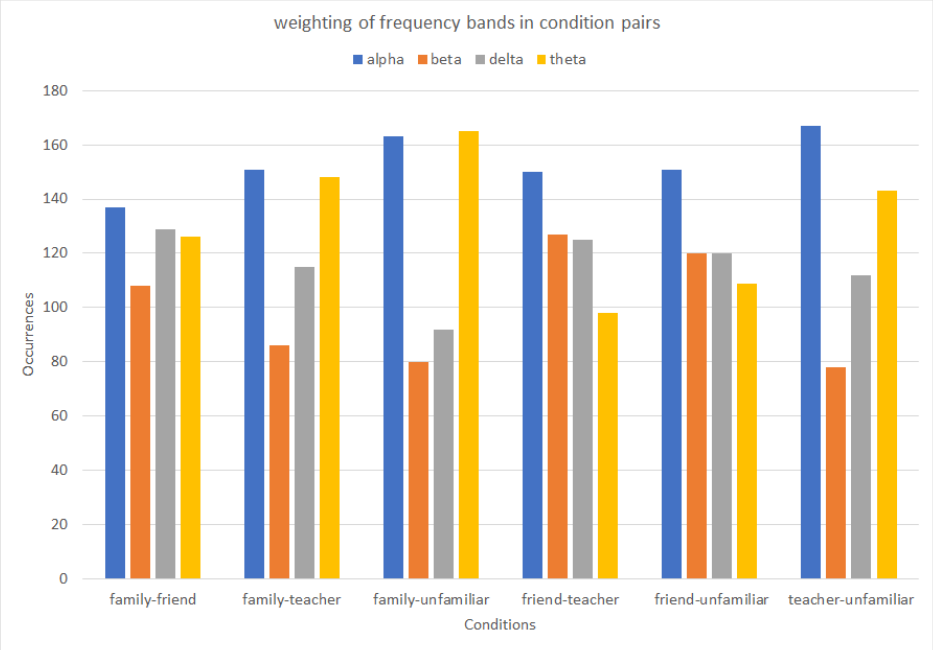


Figure 3.6: *Sub frequencies that occurred during different classification tasks. The y-axis contains the amount of how often a feature from a frequency band appeared. The x-axis depicts the other condition in which the classification task was performed.*

When trying to distinguish between a family member and an unfamiliar person, the features from the alpha and theta frequency band occurred more often. The same behaviour is also to see between the conditions teacher and unfamiliar. Only between the friend-unfamiliar pair the result is not the same case, in there only really the alpha frequency band is dominating. The last condition pair that also has an increase of features from the alpha and theta frequency band is the family-teacher pair. To note is that these two conditions look very similar to the family-unfamiliar one.

4 Discussion

A lot of research has been already done in the fields of neural responses of speaker recognition and identification. It is no secrete that when talking to a friend, a family member or a complete stranger, one not only behaves differently but also feels different when they are around them. By decoding the EEG data stream and finding specific activity patterns in the signals (with features), allows one to predict the stimulus that causes the activation in our brain.

Regarding the classification results from this work, one can observe that they get less accurate if no ERPs are in use. The attempt, in general, was to analyze the relationship between phase synchronization of EEG signals in all sub-oscillatory bands and the distribution of power of frequency components and then use this information to classify the different voices. Unfortunately, the classification did not achieve as high results as they probably could have with ERPs.

Before going into more detail, regarding the classification accuracy results, the questionnaire's outcome also needs to be discussed. Concerning the results from Figure 3.1 about the answered questionnaire, one thing can be said with certainty: all the participants paid attention throughout the process. On average, out of 60 questions, only two were skipped or answered incorrectly. In addition, every participant answered different questions wrong, which means one can not blame the difficulty on one particular question. Also, each study participant needed 13 seconds to answer a questionnaire on average, which leaves the amount of time for relaxation (until the next audio presentation started) about 48 seconds. That confirms that each participant had more than enough time to answer the questions and then prepare for the next audio presentation, in which they needed to focus again and move as little as possible.
Based on only the wrong and skipped questions, it is impossible to determine if participants lost focus or were tired at the end of the experiment since they all answered completely different questions wrong at random time points.
Nevertheless, at the end of the measurement, the participants were asked how they felt and experienced it. Nearly every one of them thought that the questions were doable, but regarding the measurement duration, they all were on the same page: "it was pretty exhausting". They had difficulty concentrating for that extended period, despite their relaxation time between the questionnaires and audios. A solution to that problem could have been to break the whole experiment into blocks, so participants would have time to make small breaks and regenerate. Breaks that lasted more than only 20 seconds and where they would have the opportunity to determine by themselves when they wanted to continue with the experiment.

Another inconvenience was that the stories had significantly different durations. They lasted from 1 to 2 minutes, which caused a problem, especially during the making of the dataset for the classification part (after epoching). For example, some of the stories had a whole minute of EEG data more than others, resulting in unbalanced datasets since every story was epoched the same way. The consequences of this are visible in Table 3.2 and Table 3.3. Although, depending on which method one uses for calculating the chance accuracy, the results can quite vary. So are the results from the naive model (Table 3.2) more sensible to the class distribution than the ones from the upper chance level (Table 3.3). However, this does not mean that one method is better than the other. It just serves as a comparison and to clarify that depending on the model one chooses, the results might vary.
In general, the unbalanced data distribution causes the chance accuracy to rise, which means the overall performance of the specific classifier drops. To overcome this dilemma, one could select the stories to be more uniform, which is not such a simple task since finding 20 stories of the same length is quite challenging. An alternative could also be to trim those stories with a long duration, but without losing too much of information about the story.

The results from the classification part are not as promising as the ones of Wenwen Chang et al. [6]. Their study reached classification accuracies up to 90%, whereas this work only reached results as high as 74%. Although, this was predictable since not only completely different features were used, but also the experimental procedure differed a little bit from this works ones. Their study selected, for efficient feature extraction of the multichannel EEG signal, directed functional network parameters, signal complexities, and Hjorth parameters. In this work, however, only phase synchronization and PSD values were of interest. That would be an idea to consider for future measurements, to select different features for the classification, or to try a combination of the used features and new ones. The basic idea behind the phase synchronization was that since it produced pretty good results in the study about music evoked pleasantness [2], the thought was that it would also adapt quite well to this paradigm.
Based on Table 3.6, the accuracy varies depending on the participant. So are the best results for participants 2 and 5, while for number 4, the accuracy did not even reach the chance level. One explanation for the relatively bad performance of participant 4 could be his lousy posture during the EEG experiment. While all the other participants sat comfortably and leaned back during the audio presentation, participant 4 slightly leaned forward, which may have caused the noisy results. Despite that, the collected EEG signals were good, except for some channel noise.

The degree of success of the experiment is hard to tell since only a small number of people took part in the study. To give more ensuring results, one has to carry out the same audio presentation for a bigger group of participants. However, based on only the 5 study participants, it leads to the conclusion that it is possible to differentiate between the different conditions since classification results sometimes even reach 20% over chance

level. This just confirms that distinguishing the responses of different voices in the EEG data is even possible without ERPs, although it has to be said that with ERPs the accuracy would probably be better.
What is noticeable in the results is that it is easier to classify familiar voices than unfamiliar ones. The best results were obtained by trying to distinguish between familiar voices, like a family member and a friend. At the same time, the lowest accuracies were between unfamiliar and teacher voices, with which the participant had the poorest relationship. This is no surprise since it is already known that familiar talkers are more intelligible than unfamiliar ones [11] and therefore probably the processing of the voice in our brain more intense.

Although the accuracy results were not as promising as expected, there is still room for improvement. A possibility would be to improve the result by just creating more data, so instead of 5 stories per condition, one could increase it up to 10. By having more data to classify, the results would be more accurate since outliers in the dataset would not have such a significant influence on the results. It was only possible to gather about 350-400 trials per participant, resulting in a relatively small dataset. In addition, there was also the problem that some of the trials got removed after epoching, due to the high amplitude that some of the trials had. The cause of that was the high channel noise in the data, which could not be completely removed without losing important information. On average, only about 20 trials got rejected from each participant's dataset. At first, it does not sound much, but considering the small dataset, such a data loss has quite an impact on the results.

Besides the classification accuracy, there are also the results from the frequency bands. What stood out was that the features from the alpha and theta frequency band appeared the most during the classification. The study from Wenwen Chang et al. [6] used a different approach for selecting the best frequency band. They separated the features by frequency band before applying the classifier and then only used the features from one specific frequency band for training the classifier. This resulted to the point that the best classification accuracies were obtained with the features from the delta frequency band (90%). The following places are alpha and delta with 86% , and the beta band had the lowest accuracy with only 78.86%.
The results from this work are different from the study's ones. Mainly alpha and theta were the ones that dominated, and delta and beta were more in the background. However, one could argue that these differences are because other features were used or the general use of ERPs.

In general, theta activity is categorized as a "slow" activity and ranges from 3.5 to 7.5 Hz. It is often connected or categorized for memories, feelings, and experiences and is associated with creativity, daydreaming, and fantasizing [18]. While the alpha oscillatory band (7-12Hz) is mostly active in adults in their normal relaxed state and often described as the bridge between the conscious to the subconscious.
It would have been interesting to also investigate the behaviour of the gamma (30-50 Hz

or higher) frequency band, since it is already known they have stronger electrical signals in response to visual stimulation [1]. Taking the gamma band into consideration, would be a great proposal for future measurements.

Bibliography

[1] M. Abo-Zahhad, Sabah M. Ahmed, and Sherif N. Abbas. "A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals". In: *International Journal of Intelligent Systems and Applications* 7.6 (2015), pp. 48–54. ISSN: 2074904X. DOI: 10.5815/ijisa.2015.06.05.

[2] Alberto Ara and Josep Marco-Pallarés. "Fronto temporal theta phase synchronization underlies music-evoked pleasantness". In: *NeuroImage* (2020). DOI: 10.1016/j.neuroimage.2020.116665.

[3] Aravind Prasad. "Feature Extraction and Classification for Motor Imagery in EEG Signals". Master. Kaunas University of Technology, 2016.

[4] Christopher M. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. New York, NY: Springer Science, 2006. ISBN: 978-0-387-31073-2.

[5] Jason Brownlee. "What Is the Naive Classifier for Each Imbalanced Classification Metric?" In: *Machine Learning Mastery* (13.1.2020). URL: https://machinelearningmastery.com/naive-classifiers-imbalanced-classification-metrics/.

[6] Wenwen Chang et al. "An EEG based familiar and unfamiliar person identification and classification system using feature extraction and directed functional brain network". In: *Expert Systems with Applications* 158 (2020). ISSN: 0957-4174. DOI: 10.1016/j.eswa.2020.113448.

[7] Yi-Wei Chen and Chih-Jen Lin. "Combining SVMs with Various Feature Selection Strategies". In: *Feature extraction*. Ed. by Isabelle Guyon et al. Vol. 207. Studies in Fuzziness and Soft Computing. Heidelberg: Springer, Berlin, Heidelberg, 2006, pp. 315–324. ISBN: 978-3-540-35487-1. DOI: 10.1007/978-3-540-35488-8{\textunderscore}13.

[8] Arnaud Delorme, Scott Makeig, and T. Sejnowski. "Automatic Artifact Rejection For EEG Data Using High-Order Statistics And Independent Component Analysis". In: *ResearchGate* 7.10 (2002).

[9] Srivastava Durgesh and Bhambhu Lekha. "Data classification using support vector machine". In: *Journal of Theoretical and Applied Information Technology* 12.1 (2010), pp. 1–7. ISSN: 1817-3195.

[10] Quanquan Gu, Zhenhui Li, and Jiawei Han. "Generalized Fisher Score for Feature Selection". In: *arXiv* (2012). DOI: 10.48550/ARXIV.1202.3725.

[11] Emma Holmes, Ysabel Domingo, and Ingrid S. Johnsrude. "Familiar Voices Are More Intelligible, Even if They Are Not Recognized as Familiar". In: *Psychological science* 29.10 (2018), pp. 1575–1583. DOI: 10.1177/0956797618779083.

[12] Christian Kothe et al. *Introduction — Labstreaminglayer 1.13 documentation*. 2019. URL: https://labstreaminglayer.readthedocs.io/info/intro.html.

[13] Mike X Cohen. *Analyzing neural time series data: Theory and practice*. Issues in clinical and cognitive neuropsychology. Cambridge, Mass. u.a: MIT Press, 2014. ISBN: 9780262319546.

[14] Gernot Mueller-Putz et al. "Better than Random? A closer look on BCI results". In: *International Journal of Bioelectromagnetism* 10.1 (2008), pp. 52–55. ISSN: 1456-7857.

[15] Julien Plante-Hébert, Victor J. Boucher, and Boutheina Jemel. "The processing of intimately familiar and unfamiliar voices: Specific neural responses of speaker recognition and identification". In: *PloS one* 4 (2021). DOI: 10.1371/journal.pone.0250214.

[16] Md Rahman and Rasel Ahmmed. "An Advanced Algorithm Combining SVM and ANN Classifiers to Categorize Tumors with Position from Brain MRI Images". In: *ResearchGate* 3 (2018), pp. 40–48. DOI: 10.25046/aj030205.

[17] Mohammad Hossein Same et al. "Simplified Welch Algorithm for Spectrum Monitoring". In: *Applied Sciences* 11.1 (2021), p. 23. ISSN: 2076-3417. DOI: 10.3390/app11010086.

[18] Seidi Suurmets. *Neural Oscillations - Interpreting EEG Frequency Bands*. 2018. URL: https://imotions.com/blog/neural-oscillations/.

[19] TOSHINAKI et al. *MATLAB_PTB_Questionnaire: 2016/08/06 first release*. 2016. URL: https://github.com/Toshinaki/MATLAB_PTB_Questionnaire.

[20] Andre Violante. "An Introduction to t-SNE with Python Example - Towards Data Science". In: *Towards Data Science* (29.8.2018). URL: https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1.

[21] Wei Bin Ng et al. "PSD-Based Features Extraction For EEG Signal During Typing Task". In: *IOP Conference Series: Materials Science and Engineering* (2019). DOI: 10.1088/1757-899X/557/1/012032.

[22] Jun Yang et al. "4-Class MI-EEG Signal Generation and Recognition with CVAE-GAN". In: *Applied Sciences* 11.4 (2021), p. 1798. ISSN: 2076-3417. DOI: 10.3390/app11041798.