Independent project, 15 credits, for the degree of
Master of Science in Computer Science with Specialization in
Embedded Systems
Spring Semester 2018
Faculty of Science

Automatic Control of a Window Blind
using EEG signals

Marijana Teljega
Author
Marijana Teljega

Title
Automatic control of a window blind using EEG signals

Supervisor
Dr. Fredrik Frisk, HKR

Examiner
Professor Eric Chen, HKR

Abstract
This thesis uses one of Brain Computer Interface (BCI) products, NeuroSky headset, to design a prototype model to control window blind by using headset’s single channel electrode. Seven volunteers performed eight different exercises while the signal from the headset was recorded. The dataset was analyzed, and exercises with strongest power spectral density (PSD) were chosen to continue to work with. Matlab’s spectrogram function was used to divide the signal in time segments, which were 0.25 seconds. One segment from each of these eight exercises was taken to form different combinations which were later classified.

The classification result, while using two of proposed exercises (tasks) was successful with 97.0% accuracy computed by Nearest Neighbor classifier. Still, we continued to investigate if we could use three or four thoughts to create three or four commands. The result presented lower classification accuracy when using either 3 or 4 command thoughts with performance accuracy of 92% and 76% respectively.

Thus, two or three exercises can be used for constructing two or three different commands.

Keywords
NeuroSky, EEG signal processing, Extract Features, Classification, ThinkGear library, BCI
Table of Contents

1 INTRODUCTION ................................................................................................................................. 5
  1.1 BACKGROUND ............................................................................................................................... 5
  1.2 LITERATURE STUDY ..................................................................................................................... 6
    1.2.1 Literature Review .................................................................................................................... 6
    1.2.2 Signal acquisition and segmentation ..................................................................................... 8
    1.2.3 Feature extraction and classification methods ................................................................. 9
    1.2.4 Literature search findings summarized ............................................................................. 10
  1.3 RESEARCH QUESTIONS ................................................................................................................. 10
  1.4 THESIS ORGANIZATION ............................................................................................................... 10
  1.5 ACKNOWLEDGEMENTS ............................................................................................................... 10

2 METHODOLOGY AND EXPERIMENTAL DESIGN .................................................................................. 11
  2.1 OVERVIEW OF EXPERIMENTAL SETUP .................................................................................... 11
  2.2 MATERIALS AND METHODS ...................................................................................................... 12
  2.3 DATA ACQUISITION ..................................................................................................................... 12
  2.4 ANALYSIS OF EEG SIGNALS, FEATURE EVALUATION AND EXTRACTION ... 13
  2.5 MODE CLASSIFICATION ............................................................................................................. 14
  2.6 COMMANDS/ MODE IDENTIFICATION .................................................................................... 15

3 IMPLEMENTATION AND RESULTS .................................................................................................... 16
  3.1 MODEL DEVELOPMENT .............................................................................................................. 16
  3.2 ANALYSIS OF SPECTROGRAM ................................................................................................. 16
  3.3 FEATURE EXTRACTION AND CLASSIFICATION ........................................................................ 19

4 DISCUSSION ........................................................................................................................................... 23

5 CONCLUSION .......................................................................................................................................... 24

6 FUTURE WORK ...................................................................................................................................... 25

7 REFERENCER ......................................................................................................................................... 26

8 APPENDIX A EEG TECHNOLOGY ........................................................................................................ 28
  8.1 BRAIN AND ITS ACTIVITY ......................................................................................................... 28
  8.2 HUMAN NERVOUS SYSTEM AND THEIR FUNCTIONAL RELATIONSHIP ... 28
  8.3 EEG SIGNAL MEASUREMENT .................................................................................................. 30
  8.4 ERP- EVENT RELATED POTENTIALS ....................................................................................... 30
  8.5 INTERNATIONAL 10-20 SYSTEM ELECTRODE PLACEMENT ............................................. 31
  8.6 FREQUENCY BANDS IDENTIFYING BRAIN ACTIVITY PATTERNS ............................................. 31
  8.7 DIFFERENT TECHNIQUES TO MEASURE BRAIN ACTIVITY .............................................. 32

9 APPENDIX B BCI- BRAIN COMPUTER INTERFACE ............................................................................... 34
  9.1 WHAT IS BCI? ............................................................................................................................. 34
  9.2 MODELING SINGLE NEURON .................................................................................................... 34
  9.3 CONCLUSION TO DRAW ABOUT THE MODELING .................................................................. 35
  9.4 CONSTRUCTION OF THE BCI SYSTEM ..................................................................................... 35
  9.5 DATA ACQUISITION AND SEGMENTATION .......................................................................... 35
  9.6 FEATURE EXTRACTION ............................................................................................................. 36
  9.7 FEATURE EXTRACTION PROCESS DESIGN ............................................................................. 37
  9.8 CLASSIFICATION ....................................................................................................................... 38

10 APPENDIX C THINKGEAR™ .............................................................................................................. 39
  10.1 THINKGEAR PACKET STRUCTURE ......................................................................................... 39

11 APPENDIX D FEATURES AND CLASSIFICATION RESULTS .......................................................... 41

12 APPENDIX E ANALYSIS AND EVALUATION ..................................................................................... 49
  12.1 EXPERIMENTAL DESIGN AND ANALYSIS ......................................................................... 49
  12.2 COMPUTATION OF EFFECTS .................................................................................................. 49
  12.3 ALLOCATION OF VARIATION ................................................................................................... 50
List of Figures

Figure 1. 1 Automatic control of a window blind ................................................................. 5
Figure 2. 1 Processes the system needs to handle for each incoming data .................. 11
Figure 2. 2 The NeuroSky MindVawe Mobile ................................................................. 12
Figure 2. 3 Signal processing design ............................................................................. 14
Figure 2. 4 Flowchart of activities in the system .......................................................... 15
Figure 3. 1 Exercise “Up” ................................................................................................. 17
Figure 3. 2 Exercise “Down” .......................................................................................... 17
Figure 3. 3 Exercise ”Balloon” .................................................................................... 17
Figure 3. 4 Exercise ”Rectangle” .................................................................................. 17
Figure 3. 5 Exercise ”Closed eyes” ............................................................................. 18
Figure 3. 6 Exercise ”Relax” ....................................................................................... 18
Figure 3. 7 Exercise ”Count back” ............................................................................... 18
Figure 3. 8 Exercise ”Point” ....................................................................................... 18
Figure 3. 9 Feature extraction and construction of classification matrix .................. 19
Figure 3. 10 Classification performance of all exercises .......................................... 20
Figure 3. 11 Performance of all task combinations using 0,25 seconds original segment ... 21
Figure 3. 12 Classification performance of four exercises ........................................ 21
Figure 3. 13 Classification performance of three exercises ....................................... 22
Figure 9. 1 Representation of membrane using electrical circuit ......................... 34
Figure 10. 1. Example of output in streamLog.txt file ........................................ 40
Figure 10. 2 Corresponding output in dataLog.txt file ............................................ 40
**List of Tables**

Table 2. 1 Recorded dataset information
Table 3. 1 The best accuracy classifiers with all eight exercises
Table 3. 2 The best accuracy classifiers for combinations with two and with three exercises
Table 11. 1 Classification with 8 exercises
Table 11. 2 Measured classifier accuracy for combinations with "Up" exercise
Table 11. 3 Measured classifier accuracy for combinations with "Down" exercise
Table 11. 4 Measured classifier accuracy for combinations with "Balloon" exercise
Table 11. 5 Measured classifier accuracy for combinations with "Rectangle" exercise
Table 11. 6 Measured classifier accuracy for combinations with "Eyes closed" exercise
Table 11. 7 Measured classifier accuracy for combinations with "Relax" and
Table 11. 8 Measured accuracy with four and three exercises
Table 12. 1 Measured classifier accuracy for the two features
Table 12. 2 Result of effect analysis
Table 12. 3 Estimated accuracy for all combinations
Table 12. 4 The errors for all observations
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Auto Regressive Model</td>
<td>23</td>
</tr>
<tr>
<td>ARMA</td>
<td>Auto Regressive Moving Average model</td>
<td>23</td>
</tr>
<tr>
<td>BC</td>
<td>Bayesian Classifier</td>
<td>25</td>
</tr>
<tr>
<td>BCI</td>
<td>Brain Computer Interface</td>
<td>5</td>
</tr>
<tr>
<td>CHKSUM</td>
<td>Checksum packet</td>
<td>26</td>
</tr>
<tr>
<td>CNS</td>
<td>Central Nervous System</td>
<td>15</td>
</tr>
<tr>
<td>CT</td>
<td>Computerized Tomography</td>
<td>19</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalogram</td>
<td>5</td>
</tr>
<tr>
<td>ERP</td>
<td>Event Related Potentials</td>
<td>9</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
<td>8</td>
</tr>
<tr>
<td>FL</td>
<td>Fuzzy logic</td>
<td>25</td>
</tr>
<tr>
<td>fMRI</td>
<td>Functional Magnetic Resonance Imaging</td>
<td>19</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
<td>25</td>
</tr>
<tr>
<td>IF/ LIF</td>
<td>Leaky Integrate and Fire neuron model</td>
<td>22</td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbor</td>
<td>25</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
<td>25</td>
</tr>
<tr>
<td>MEG</td>
<td>Magnetoencephalography</td>
<td>20</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
<td>19</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Networks</td>
<td>25</td>
</tr>
<tr>
<td>P100</td>
<td>Positive potential peak occurred 100 milliseconds after event is triggered</td>
<td>18</td>
</tr>
<tr>
<td>PAYLOAD</td>
<td>Packet data payload</td>
<td>26</td>
</tr>
<tr>
<td>PET</td>
<td>Positron Emission Tomography</td>
<td>19</td>
</tr>
<tr>
<td>PLENGTH</td>
<td>Packet length value</td>
<td>26</td>
</tr>
<tr>
<td>PNS</td>
<td>Peripheral Nervous System</td>
<td>15</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
<td>8</td>
</tr>
<tr>
<td>SEE</td>
<td>Sum of Squared Errors</td>
<td>37</td>
</tr>
<tr>
<td>SPECT</td>
<td>Single-photon Emission Tomography</td>
<td>19</td>
</tr>
<tr>
<td>SS0</td>
<td>Sum of Squared of the mean</td>
<td>37</td>
</tr>
<tr>
<td>SSA</td>
<td>Variation due to factor A</td>
<td>37</td>
</tr>
<tr>
<td>SSB</td>
<td>Variation due to factor B</td>
<td>37</td>
</tr>
<tr>
<td>SST</td>
<td>Sum of Squared Total</td>
<td>38</td>
</tr>
<tr>
<td>SSY</td>
<td>Sum of Squared of all observations</td>
<td>38</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machines</td>
<td>25</td>
</tr>
<tr>
<td>SYNC</td>
<td>Sync packet values</td>
<td>26</td>
</tr>
<tr>
<td>TGAM</td>
<td>ThinkGear ASIC Module</td>
<td>12</td>
</tr>
<tr>
<td>WEKA</td>
<td>Waikato Environment for Knowledge Analysis</td>
<td>9</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

The brain computer interface (BCI) research have grown in the last 20 years from being only three groups, to six-eight groups 10 years later. Today there are more than 100 active groups around the world doing scientific research in BCI [1].

One of the products used in the BCI research is NeuroSky mindset [2]. NeuroSky mindset reads brain activity signals, where the signals are realized by only one electrode attached to the headset. The NeuroSky has been available for several years now, and applied to different kinds of games, learning/school applications and recently in health applications [3].

The purpose of this thesis is to create a prototype to control the window blind by using NeuroSky headset by thinking tasks that shapes one of four different commands: “Start the application”, “Window blind up”, “Window blind down” or “Window blind stop”, see Figure 1.1. The performance of the created prototype will be examined.

Recordings of brain signals, named electroencephalogram (EEG), are traditionally lab–based, where a person wears a cap with many electrodes glued on the scalp for accurate signal reading. Issues with this kind of EEG signal reading has been recognized as not very convenient and not comfortable to use, especially because people with disabilities need to wash their hair after the recordings [4]. Compared to the traditional signal reading, NeuroSky mindset with only one electrode records the signal, gives limited signal/data result with lower reliability when eyes are opened [5]. While the data acquired from the NeuroSky mindset signal is enough to encourage creating applications to help people with disabilities in their daily life.

The creation of the prototype includes the processing of the signal. EEG signal processing is done in two main steps; feature extraction, where several features are chosen to identify the EEG signals, and classification based on a set of features to get the commands from the features [6].

It is important to decide how many features are needed to be implemented in the classification algorithm without loosing high precision of the result, but keeping a balance between the correct amount of features and the correct approximation [7].

When implementing too many features, the result will be more accurate but it will take too long time for the algorithm to process all features, which will result in taking too long time to control the window blind, which will not be an usable product.

When implementing too few features, the result will be processed fast but there is a risk that the result will not be enough accurate to recognize the correct command and thus useless.

![Figure 1.1 Automatic control of a window blind](image-url)
1.2 Literature Study

This work starts with a lot of questions in the area.

- How do neuroscience and computer technology work together in the area?
- How is EEG signal processed? What different kind of measurements of the signal exist, both theoretically and practically?
- What measurements can we do by ourselves to avoid going to hospital using heavy machine for getting EEG signals?
- What developing environment is used for reading the EEG signal, processing it, and which environments are good to start with as a beginner in BCI field?

Those questions are the starting point for this thesis, building the foundation for the research questions of this work.

A systematic literature review is realized to identify relevant literature for accomplishing this thesis work. Starting with a big picture, an overview of the project, several research areas were identified and work as a starting point for deeper research. The next step was to narrow down the article selection in each area until interesting articles for this particular work have been found. It is important to mention that all chosen articles are peer-reviewed.

Depending on the area of research, different queries were written for searching. For example, queries about the EEG theory have important content that is written a long time ago, and should be narrowed down by importance of the trusted material and not focusing only on the latest 5 years. But the information about the NeuroSky headset is quite new and the searched articles starts from the year after the device has been developed and used by others.

Another important point to mention is the use of Google search in understanding possible problems that others have had and wanted to share together with some solutions for those issues. Another good kind of search that Google offers is searching for pictures, i.e. the presentation of the graphs gathered during different mode stages. Images that have been found on Google pictures, gave a link to scientific articles as well, which was very useful and saved a lot of time for getting there correctly by ourselves. Wikipedia homepage served for pointing out the references used to related articles, in the area of search.

We were searching for one good written book, covering the theory and how to work in practice when analyzing EEG signals, their issues and solutions, filled with lots of references pointing to scientific articles for further reading. The book should also be written to persons having mathematical, signal processing and machine learning knowledge, as well as basic knowledge in neuroscience, at least on the level of advanced undergraduate student. To find a book with those characteristics was a challenge. It was also important to search for a book that has got good recommendations from the readers. This search was made outside the Summon database; on Google, Matlab forums, journals covering the BCI and their references, other university and their forums, as well as what book was used in Neuroscience and machine learning courses in different universities for example Stanford and Harvard University. As a result, one book was the obvious winner; the name of the book is “Analyzing neural time series data- Theory and practice”, written by Mike X Cohen [8].

Part of the information about EEG theory and some parts of BCI technology come from three books; “Analyzing neural time series data” [8], “Neuroscience” [9], and “Fundamentals of computational Neuroscience” [7].

1.2.1 Literature Review

Database named Summon which is provided by Kristianstad University is connected to numerous popular databases in the world, and because of that, it was chosen as the main tool for the article search for this thesis work.
The four main keyword areas used for searching are: EEG theory, BCI technology, Classification of the EEG signal, and Ethical aspects. Those areas were searched several times resulting in getting more specific titles, as presented here:

**EEG theory**: “brain activity”, “nervous system”& “functional relationship”, “EEG signal measurements”, “frequency bands”& “brain activity”, “electrode placement” and “measurement techniques”. This research gave 87 articles to use in further work.

**BCI technology**: “survey”, “Time domain versus frequency domain”, “EEG raw data signal analysis”, “Filtering the EEG signal”. This search resulted in 69 articles.

**Classification of the EEG signal**: “Feature selection and extraction”, “classification algorithm” resulted in 3 articles.

**Ethical aspects**: “EEG recording”, “Save”, “Use”, “Human”, “Sweden”, and “World” resulted in 3 articles.

Out of the result, 87 articles were found during the search. They were further narrowed down to smaller result, 15 articles, by reading the content in the abstracts and choosing those articles that contain relevant content.

The NeuroSky headset was developed in 2004, but there might be some interesting facts about other similar products, and because of that the searching was limited back to year 2000. Unfortunately, this approach didn’t eliminate any article.

The search of articles was limited to those articles published during 2000 – 2017, with a jump from year 2013 to 2014 with 15%. After 2014 the research in BCI area holds the same high level for the last three years. Most of article titles present the research in area that handles designing of BCI for those people who suffered from stroke and came back to normal life with the help of rehabilitation, by using BCI.

One of the research problems, is the reliability of a single electrode used for EEG raw data recordings. The article [5] presents that there is not enough proof showing that NeuroSky headset is reliable and that more research is desired. This article is used as one of several bases for this thesis work to answer the question how NeuroSky can be used to control a window blind. The Neurosky was tested for four different tasks to do: two of them in relaxed state and two were active tasks to achieve attention state. This research presents that the signal has better reliability when the eyes are closed than when the eyes are opened. However, we made a further contribution by proving that the use of a single electrode is reliable; we have designed several more tasks for the user of the headset to perform.

Another article presents in [10], how to process and record the EEG signal, both theoretically and how to implement it. Discussion about the NeuroSky library and its useful methods are presented, as well as Fast Fourier transform that has been used to transform the signal from time domain to frequency domain. Two mode states were tested; relaxed and concentrated. The research was successful in recognizing two modes. The signals which were presented in graphs, both in time and frequency domains, have been analyzed and discussed. The limitation of the research presented in this article is that no classification has been performed. Our contribution shows that segmenting data with Matlab’s function named Spectrogram is desirable where using Hamming-window. We put the overlapping parameter value to 75% of the whole window, which divides the signal in time-segments that are not too small to miss important data but still contain enough of data to perform a classification. The classification resulted in 97% accuracy.

The prototype design was carried out with the use of NeuroSky headset and presented with the goal to implement the system, which recognizes two mental tasks, relaxation and concentration [13]. The result of the two mental tasks was analyzed and divided in frequency band values. The EEG signals have been amplified and filtered in several steps, several times to get the signal of interest removing unnecessary signals, where Arduino is used in between to receive the signal before it is transferred to the computer. The research result was successful in identifying the two mental modes. We contribute with our experiment which shows that the successful result is achieved without filtering the raw data.
Another research has been performed on characterization of EEG signals, proposing new developed algorithm for distinguishing between relaxing and focusing mode. The measurements were based on Power Spectral Density (PSD), within 10-50Hz range. During the relaxing mode, the value was higher than during focusing mode, and this was the base for the new proposed algorithm. The result presents that the signal showed fine differences in relaxation and focused EEG signals, and as well artefacts between 40 and 60 Hz created by the device [14]. In this thesis, we make further contribution by investigating the PSD with strongest value in frequencies that only exceeds 30 Hz showing that it is sufficient data to be used for the experiment, while we decrease the memory consumption and computational time as well.

1.2.2 Signal acquisition and segmentation

In the case of NeuroSky headset, there is only one channel that takes care of recording EEG data. This also means understandably that the result will be thinner/ smaller in comparison when using multiple electrodes for measuring the signal.

When being in the attention/ focused state, the signal is found in the frequency range from 12- 30 Hz, also named beta wave. The relaxed mode signal is found in the frequency range from 7.5- 12 Hz, also named alpha wave. Brief information of the frequency ranges is given in Appendix A.

The measurement of alpha wave can be carried out both in front of and back of the head. Beta wave is found in central and frontal area of the head. The strength of the signal is individual. The research shows that the back of the head gives overall stronger alpha signal effect [15].

For the recording time length, there is nothing directly showing that there is a specific time length to use when measuring alpha or beta waves. Instead, articles point out that the length depends on which particular task to solve and connects the length of that particular problem to how it has been analyzed and to the method that gives the best result. One example is that there should be the acceptable length for doing FFT analysis [16]. The article stated that the focus should not lie on the length or how many trials, but it should lie on finding the best individual signals that the user can use for controlling and then do an analysis method. This is especially true for people having some part of the brain damage, where they must use the best kind of signals they can achieve.

To be able to record alpha or beta EEG signals with a successful result, we need to get into relaxing and focusing mode state during the recordings. We can get into focusing mode while reading specific text or calculating math mentally, by taking a quiz containing simple math problems, or by reading an article [14]. Relaxing state was accomplished by sitting at rest, trying to not think, breathe deep, relax muscles and close eyes. If the eyes are closed the recordings don’t have to deal with the ocular artifacts.

Researches and studies show that it is important to find out what exercise combination is giving EEG reaction during the focusing and relaxing mode. There is as well statistics which presents that around 11% of people is not able to show alpha EEG signal [15]. For this project, we need to concern this fact as a possibility to happen before we get into the classification process.

If both mode activities were recorded first alone, and later recorded together in the same session, it makes it easier to see the differences between those two state modes [14]. In this study, four people participated in the recordings. The length of the trials and how many times recordings were repeated, for both modes, are different in research articles (if presented). The study case in [10], conducted 400 seconds of one trial, and later divided that in 100 equal parts getting 4 seconds of each part. Here, the author does not do a classification, but only does the Fourier transform to get the power/effect for the purpose of analyzing the graphs in time- and in frequency-domain.
The work [3] conducted 20 trials for visual calculating, i.e. focused concentration with closed eyes, and 20 with opened eyes, and found that signal wave is stronger when eyes are closed, showing as well alpha wave present.

When dealing with recording blinks, it was enough to record 5 samples, with 1280 milliseconds to present visually a major variation in the signal [3]. Blinking or eye movement, heart or muscle related activity are all biological and environmental artifacts showing changes in power/ effect in the EEG signal [11].

Finally, the literature [8], suggests than there is no magical number of trials to use but 50 trials per condition per person should give a good level of signal-to-noise ratio. But if we only look for frequency effect on ERP response, it is statistically reliable with 14 trials (see Appendix A).

1.2.3 Feature extraction and classification methods

Brief theory about the feature extraction and classification methods in the area of this thesis work, is given in Appendix B.

The study performed by [14] used 512 Hz as a sampling frequency (NeuroSky own sampling frequency), to get the data in time-domain. Because no differences could have been seen in the signal in time-domain, the data was segmented by Matlab’s function Spectrogram to present each segment in frequency-domain. Hamming window was used as a time-window, where the writer states the choice of that particular window is that it is widely used for basic Fourier analysis. The chosen time-window size was 0.4 seconds, i.e. 200 samples per window, where the size is based on how good the signal looks in Matlab’s spectrogram. The next step is about finding the average power at each frequency, and for that PSD function was used. The experiment shows that more energy is found between 10-50Hz, where the resting state has the higher value than value in reading state. Because of that this range was chosen to be used as feature for further work.

The study [5] performed mental tasks dividing between relax and concentrate as well, but in this case the feedback from the user was expected, by pressing computer keyboard when instruction says to do that. The experiment was using the pauses (2000ms) between the tasks which were presented to the user in 500ms. Auditory stimulus was as well performed in this study, by listening to some tones which occurred randomly and pressing the keyboard as a reaction. Here, the target-noise consisted of 1000Hz with 1 second interval, and the tone, which was not important to give feedback consisted of 500Hz. The raw data was sampled at 128 Hz, band-pass filtered at 0.5-30Hz, and the search was done manually for artifacts which were taken away in further processing. Four seconds Hamming time-window was used in further Fourier transformation. The result is a power spectrum, where absolute power in all frequency bands are summarized to get total power. In the next step each frequency band power was divided by total to get the existing power percentage for each individual band. The experiment found that the differences between modes are not so clear, but they confirmed as other studies did, that alpha wave is stronger when eyes are closed in contrast to when opened. They suggest that more research need to be done to have a better distinguishing result between different modes.

In a study by [17], 20 participants attended recordings of 5 different activities: reading a short story and relaxing, listening, watching short movie and a task including problem solving- a Sudoku game. Based on those tasks features were extracted to make a classification for the relaxing and concentrating state. The sampling frequency was 512Hz, and moving time-window was set to 1 second, using Matlab. The features are found by using Fourier transform on original time-domain signal to band-pass frequency bands. Averaging of each frequency bands gave the following features, the average of focusing and relaxation values, and the average of power band values for the frequency bands. Classification has been successful using Bayesian network, with recognizing reading with 97% accuracy and recognizing relaxing mode with 79% accuracy. WEKA [18] was used for feature selection and classification.
1.2.4 Literature search findings summarized
The literature review states that further research using one dry electrode is desirable, to be able to find better methods that more clearly divide the attention and relaxation modes. The study cases also provided the PSD method as one good technique to use for finding power variation in frequency bands. Many cases don’t present classification method, rather their results present if states are possible to distinguish. The Hamming time-window was mostly used in study cases in the articles.

1.3 Research Questions
The main purpose of this work is to design, implement and analyze the performance and usability of the resulting prototype, i.e. perform analysis on how NeuroSky headset can be used to control window blind. In this thesis, we will try to answer the following research questions:
- How responsive is the final prototype?
- What are the weaknesses and advantages of using NeuroSky mindset with one electrode for controlling window blind?

1.4 Thesis Organization
The organization of the report has been divided in three main parts; the first part is introducing the reader to the thesis idea, including the background information. The second part of the report presents methods and resources used to develop the system design. The third part reviews and summarizes the result of the implementation and makes suggestions for further work.

The reader can find brief theoretical background concerned with this project in the appendix parts.

1.5 Acknowledgements
This master thesis is the result of knowledge gained from master courses in embedded systems and from neuroscience course.

I would like to thank anyone who was directly or indirectly related to this project, colleagues, friends and family for giving me push forward in times when I mostly need it, especially to Professor Åke Arvidsson and Dr. Dawit Mengistu. Thanks to my supervisor Dr. Fredrik Frisk, and my examiner Professor Eric Chen.
2 Methodology and Experimental Design

To answer the research questions we will do an experimental work with observations and quantitative data analysis. We need to investigate what kind of exercise combination gives the best pattern i.e. the highest classification performance.

The following sections will present exercises that are used in this experiment. The acquisition of EEG signal’s raw data is performed with NeuroSky headset, and the same data need to be converted to voltage data before it can be used for further work with feature detection and extractions. Matlab will handle signal processing, feature extraction and classification.

Machine learning is applied in this experiment to give the best classifiers that can work in the final system. The following section will present each process in recognizing the best exercise combination for this experiment.

2.1 Overview of Experimental Setup

Our design of the system aims to make it possible to read raw EEG data, pre-process it, and make use of recognized mode to form command that are used in the system. We use Bluetooth as a communication protocol between the server and the client to handle communication from headset to window blind.

With a correct design of the system it is supposed that the reaction of the system will result in giving feedback automatically; the person sees directly if the chosen mode is successfully rolling up or down the window blind.

Because there is only two modes that can be caught by the headset, the system will use blinking in different time spectrum to form the rest of the commands that needs to exist in the system.

The Matlab application will listen for EEG signals from the client headset, to further communicate with Java server telling it what is the mode or if it is the blinking. The Java-based server application will handle Bluetooth communication to the window blind, see Figure 1.1. with following commands:

1. Start application,
2. Window blind up,
3. Window blind down,
4. Window blind stop.

The Figure 2 shows processes that are carried out for all commands.

Figure 2.1 Processes the system needs to handle for each incoming data
2.2 Materials and Methods
Matlab is used to collect data from the signal acquired from the headset. The classification result is identifying the classes of the focused or relaxed commands. From the same classification result, Matlab generates the code which is transferred to the server to be used for recognizing commands.

The device we used in this experiment is NeuroSky MindWave, Figure 3, works on 3.3 V and uses a dry sensor technology, meaning that there is no physical preparation to apply the electrode on the head [19]. The headset integrates TGAM, ThinkGear ASIC Module, single chip EEG sensor, which is placed in the headset, to receive EEG signals, filter out the noise, amplify and convert to digital power. The headset uses 512 Hz as a sampling frequency.

The EEG sensor is placed on the arm of the sensor, which rotates so it reaches the left part of the forehead to read the small changes in activity. The second contact point is placed on the ear lobe connected to the headset, and serves as the reference point and the ground.

Figure 2.2 The NeuroSky MindWave Mobile

For the purpose of this project we need to parse data stream to get bytes (raw values) and use it further for the classification purpose. More information about functionality of the ThinkGear is found in Appendix C. To convert raw value unit to the voltage value, we use the equation [2] presented in Appendix C.

2.3 Data Acquisition
In this section, we show how to get into focusing and relaxing mode state, how many trials are done and how long time each trial is taking. From the overall design perspective, the person that is intended to use this headset will not first get stimulated visually or auditorily, i.e. with the light flash or sound tones. Instead the person is getting by himself in one of two mode states.

It is not practical to need to prepare first and then control the window blind going up or down. Using the headset is not intended to include preparation time before using it, which means that the person doesn’t need to prepare before using it, i.e. no need to listen to calm music to get into the relaxed state.

Seven persons, including the author, participated in doing eight different tasks. The only reason that we are seven persons in this study is that all participants were volunteers, and there is no direct relation in regard to the old, race, culture according to the found literatures. Volunteers are between ages 35-66, healthy, with no known concentration issues.

The first four tasks are performed with opened eyes, and the last four with closed eyes. Each trial is 20 seconds long, and the recordings are repeated 5 times.

Tasks with opened eyes:
1. “Up” - Imagine and think how you with your mind move the window blind up successfully. Imagine that you see more and more of the window.
2. “Down” - Imagine and think how you with your mind move the window blind down successfully. Imagine that you see less and less of the window.
3. “Balloon” - Think of one white balloon. Imagine that the wind blows and plays with the balloon. Concentrate on the movement of the balloon and its white color.
4. “Rectangle” - Think of one blue square. The contour line has thick black color. Follow the contour line and concentrate on the blue color.
Tasks with closed eyes:
1. “Eyes Closed” - Think of whatever with eyes closed.
2. “Relax” - Close the eyes and relax in your muscles.
4. “Point” - With closed eyes imagine that you have a black dot between the ears in your head. Concentrate on that dot.

Some exercises are chosen to be performed with opened eyes with meaning, because we want to see if that is possible to recognize some pattern just by having opened eyes and imagining something to perform.

Several studies presented that when eyes are closed there will be stronger alpha waves than when eyes are opened. This needs to be analyzed for this particular headset to see if that is the case in this experiment as well. The negative with this approach is that the user will not see if the window blind is moving, but the user can still hear the sound of window blind going up or down.

Here as well, we want to see if there is some difference or if there is a pattern showing that the exercises performed with closed eyes are stronger if the person also relaxes muscles, or if the person concentrates on something specifically like a dot between the ears in the head or if counting backwards.

Another analysis presented in the research is that eye blinks give muscle artifacts, which should not be filtered.

As presented in the Dataset Table 2.1. below, the total time of data getting from the exercises is 1 hour and 55 minutes, where the total number of exercises are 280.

<table>
<thead>
<tr>
<th>No. of Classes</th>
<th>No. of volunteers</th>
<th>No. of exercises</th>
<th>Time for all exercises</th>
<th>Number of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Blind Up</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>2. Blind Down</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>3. Imagine Balloon</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>4. Imagine rectangle</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>5. Close the eyes</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>6. Close the eyes and relax</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>7. Eyes closed and calculate backwards</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>8. Eyes closed and focus on a dot</td>
<td>7</td>
<td>35</td>
<td>35x20sec= 700 sec</td>
<td>35 x 10240</td>
</tr>
<tr>
<td>9. Total</td>
<td>280</td>
<td>5600 sec = 1,55 hours</td>
<td>280 x 10240</td>
<td></td>
</tr>
</tbody>
</table>

2.4 Analysis of EEG signals, Feature Evaluation and Extraction
Firstly, as presented in the Figure 2.3. in step 1, recordings are made for each volunteer. Here we get 280 exercises to evaluate. In step 2, we want to reduce the amount of exercises so that redundant exercises don’t consume memory and computational time and to only use the strongest signal in further work. We choose one trial of each 8 exercises, reducing the exercise amount to 8 exercises. This selection is analyzed by inspecting power spectrum for each signal of the exercises.
The function used for analysis in step 2, is Matlab’s function named spectrogram [20], which divides the signal in segments and gives estimated power spectrum density for each segment.

In step 3 we continue to use spectrogram function’s output to find what segment have the strongest PSD. The size of the segment is calculated based on input parameters obtained from the literature review. This means that we now will have 8 segments where each of them represents the frequencies with strongest PSD for each exercises. The 8 segment’s PSD, frequency and time will be further used as a features in the classification searching for the best exercise combination to be used as commands.

Inside the extract feature vector process, in step 4 and 5, one exercise combination for each exercise will be placed in the Exercise dataset in step 5, which results in 28 combinations.

For the general signal processing design, see design presented in Appendix B.

Figure 2.3 Signal processing design

2.5 Mode Classification

There are 8 classes (exercises) that are combined in each possible combinations of two and two. They will be classified to decide if and how they can be used to form commands. Better to say that it is to expect to find at least one different classification pattern combination that gives high classification accuracy, because the headset can only detect two kind of modes, but we will perform more experiments to verify if more pattern can exist. The total amount of combinations are 28. Following are the possible combinations:

1. “Up” and “Down”
2. “Up” and “Balloon”
3. “Up” and “Rectangle”
4. “Up” and “Eyes closed”
5. “Up” and “Relax”
6. “Up” and “Count back”
7. “Up” and “Point”
8. “Down” and “Balloon”
9. “Down” and “Rectangle”
10. “Down” and “Eyes closed”
11. “Down” and “Relax”
12. “Down” and “Count back”
13. “Down” and “Point”
14. “Balloon” and “Rectangle”
15. “Balloon” and “Eyes closed”
16. “Balloon” and “Relax”
17. “Balloon” and “Count back”
18. “Balloon” and “Point”
19. “Rectangle” and “Eyes closed”
20. “Rectangle” and “Relax”
21. “Rectangle” and “Count back”
22. “Rectangle” and “Point”
23. “Eyes closed” and “Relax”
24. “Eyes closed” and “Count back”
25. “Eyes closed” and “Point”
26. “Relax” and “Count back”
27. “Relax” and “Point”
28. “Count back” and “Point”
Appendix B presents 7 classifiers used when working with EEG signals depending on the relative task, but in this experiment, classifiers that will be used are the classifiers that Matlab offers in Classification Learner toolbox, which are in total 23 classifiers. Depending on the classification accuracy result we will continue experimenting with the best combinations to find out the best pattern for the system.

2.6 Commands/ Mode Identification

The overall system will have Java server which will receive signals from Matlab that reads the signals which come from the headset. This application will be running constantly, listening if a blinking signal is received then it will be translated into the corresponding command which requests the system to start the application. If the same blinking signal is received twice, this will request the application to stop but the Java server will continue to run.

Once the application is started, the user can give commands up or down. The system expects input signal from the headset which comes as a raw data. The time that the system will use while expecting incoming data will be decided by the spectrogram segmenting output. Based on the received data, the trained model will decide if the command “up” or “down” will be used. The user sees the result of moving window blind up or down, and if satisfied, the user will use blink command again, but this time to close the application. The Figure 2.4. shows the flowchart of activity steps used in the system.

![Flowchart of activities in the system](image-url)
3 Implementation and Results

Here the implementation of machine learning model with the proposed approach discussed in the methodology part is presented. In section 3.1 we discuss model development, and in section 3.2 we discuss analysis of spectrogram. Finally in section 3.3 the feature extraction and the classification are discussed.

3.1 Model Development

The implementation of the model was carried out in Matlab. Different Matlab tools was used for different purposes. Signal Analyzer tool box was used to analyze the frequency spectrum to inspect if there are useful information. Another Matlab tool, Classification Learner application, was used for the purpose of classifying signal.

The raw data signal is 20 second long. The signal is analyzed with Signal Analyzer that segments the signal and apply Fourier transform. Each segment’s length is the result of spectrogram function; this function will use parameters that are based on the founding we discussed in the literature review. Following are the parameters: the input signal, the Hamming window, the number of overlap between the segments, sample frequency and the length of the signal. The Hamming window is chosen among other windows, because it doesn’t cut the edges of the signal, which means we don’t lose the data from the signal during the overlapping process. The sampling frequency is set to 512 Hz in Matlab, because NeuroSky headset samples at 512 Hz. The overlapping is set to some value based on the fact presented in Appendix B, Brain Computer Interface, which presents that we can use some value between 50-90 %, and we choose to set it to 75%. The spectrogram function outputs one matrix that contains segments of time and estimated power spectrum for each segment.

The data for all persons and all exercises, saved in tables that are in total of 280 * 10240 data points, but after feature detection and extraction, the extracted feature vector size will consist of compressed matrix, as presented in Figure 2.3. in methodology part. The resulting dataset contains one time-segment of each of the eight exercises.

After the extraction part all exercises are put in one table together, class names are labeled for each combination presented in methodology part and features are labeled as “PSD” and “Frequency”.

The next step and before the classification, we need to decide the validation technique which will divide dataset for training and for testing. Here we decide to use 80% data points for training and 20% for validation.

As discussed in methodology part, different classifier families are used, where we have evaluated in total 23 classifiers with 28 different datasets to find what classifier gives the best exercise combination.

The best trained combination model, based on the classification result will be converted by Matlab to a Matlab function. This function will be implemented on the server to classify incoming data.

3.2 Analysis of Spectrogram

After we have done our experiment with the seven volunteers, we find out that the output from the spectrogram function gives us time vector, frequency vector and PSD matrix. We notice that the time vector uses segments of 0.25 seconds and we continue to use this time-segment further in the classification process. Figures 3.1- 3.8 are spectrograms computed on all 8 exercises and zoomed in for detailed view.
Exercises with opened eyes:

- Figure 3.1 Exercise “Up”
- Figure 3.2 Exercise “Down”
- Figure 3.3 Exercise “Balloon”
- Figure 3.4 Exercise “Rectangle”
Exercises with closed eyes:

Figure 3.5 Exercise "Closed eyes"

Figure 3.6 Exercise "Relax"

Figure 3.7 Exercise "Count back"

Figure 3.8 Exercise "Point"
3.3 Feature Extraction and Classification

We can clearly observe that exercises with closed eyes give the PSD of the signal more focused which makes it easier to decide in what frequency range it exists. The PSD of each signal that is computed with opened eyes, presents that there are ocular artefact which gives spread PSD over frequency ranges.

From each exercise, one time segment that contains strongest PSD, is picked out and placed in the classification matrix, Figure 3.9. We can see in Figure 3.1-3.8 that PSD data under 1 HZ seems to relate to some artifact which is not usable in the classification. The PSD which exists in frequency range up to 30 Hz, contains data that relates to focusing and relaxing modes, alpha and beta values. The PSD data under 1 Hz and above 30 Hz is removed to work only with data that is interesting for our cases.

![Figure 3.9 Feature extraction and construction of classification matrix](image-url)
Classification with all exercises at the same time gave low performance accuracy, where the highest accuracy of 70.0% are presented in Figure 3.10, and the best classifiers are presented in Table 3.1.

<table>
<thead>
<tr>
<th>Classifier Model Type</th>
<th>Classifier Name</th>
<th>Accuracy All exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>Fine Tree</td>
<td>44.0%</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>Quadratic Discriminant</td>
<td>15.6%</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>Fine Gaussian SVM</td>
<td>28.6%</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>Medium KNN</td>
<td>29.6%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>Bagged Trees</td>
<td>70.0%</td>
</tr>
</tbody>
</table>

We continue the classification with combinations of two exercises, as planned.

We can notice that three combinations with the highest accuracy are “Count back/Point”, “Close eyes/Point” and “Relax/Point”. All three are combined with “Point” exercise, and because of that we will see if we can use “Close eyes”, “Relax”, “Count back” and “Point” to be classified as 4 commands.
The classification of the four exercises has been performed with 76% as the highest accuracy. This is quite low accuracy result, so we test further to see if we can get better accuracy, with at least around 90% if we classify three tasks. From the Figure 3.12, we observe that “Count backwards” is giving the lowest accuracy and we choose to continue classification without it.

Figure 3.13 shows that the three exercises, “Point”, “Close eyes” and “Relax”, performs classification with 92.0%.

---

Figure 3.11 Performance of all task combinations using 0.25 seconds original segment

Figure 3.12 Classification performance of four exercises

Figure 3.13 Classifier accuracy of two task combinations
The Table 3.2. presents the result of the 2 exercise combinations with the highest accuracy and their corresponding classifiers. Appendix D includes further results with all combinations.

<table>
<thead>
<tr>
<th>Classifier Model Type</th>
<th>Classifier Name</th>
<th>Accuracy Point and Count back</th>
<th>Classifier Name</th>
<th>Accuracy Point, Relax and Close eyes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>Fine Tree</td>
<td>90.9%</td>
<td>Fine Tree</td>
<td>75.9%</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>Logistic Regression</td>
<td>40.9%</td>
<td>Logistic Regression</td>
<td>-----</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>Quadratic Discriminant</td>
<td>55.2%</td>
<td>Quadratic Discriminant</td>
<td>36.8%</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>Fine Gaussian SVM</td>
<td>78.0%</td>
<td>Fine Gaussian SVM</td>
<td>60.1%</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>Fine KNN</td>
<td>97.0%</td>
<td>Fine KNN</td>
<td>78.2%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>Bagged Trees</td>
<td>92.7%</td>
<td>Bagged Trees</td>
<td>92.0%</td>
</tr>
</tbody>
</table>

In Table 3.2, we can see that the best accuracy, 97.0%, is computed with combination of two exercises, “Point” and “Count back”. This is very good rate indicating that we can use the two exercises for controlling the window blind. Further, we notice as well that combinations that uses three exercises gave lower performance, 92.0% accuracy, and they are good candidates to implement and use for three commands for controlling the window blind.
4 Discussion

In Chapter 3, we could read about results which present measurements with features, the two differently compressed dataset. Here the analytical model will be created by using the same parameters to investigate if the results are giving similar approximation of factors that are significant for the system. Even here blinking commands will not be considered because of time limit for this project.

If implementing the whole project, possible system services could be: recognizing the 4 commands and turn window blind up and down. The system services are affected by different system parameters, and if implementing the whole system we need to deal with some of the following system services: dataset size, classification model, features, communication speed, and reliability of the network, how fast is computation done by computer.

We choose to investigate how features and classifiers affect system performance. The reason we chose features and classifiers is because we assume that they mostly affect the performance of the system, and since we investigated the same parameters in the previous part of the report we want to see if results are justified. Accuracy is the performance metric that will be used in this analysis.

The experimental design is created with two-factor full factorial design, where 2 factors are used; 5 classifiers and 3 different amount of commands. The calculations are presented in Appendix E.

The accuracy with Ensemble classifier is 16.29% higher than the average accuracy. If we look at accuracy with Nearest Neighbor it has 29.88% lower accuracy than the overall average accuracy. The highest effect difference between the two classifiers are 46.17%.

The use of 2 commands have 13.88% higher accuracy than the average accuracy, while using 3 commands shows 2.64% lower accuracy than the average. And if using 4 commands the accuracy is even lower than the average, with 11.24% lower accuracy. The highest effect difference between the highest and lowest amount of commands used in the system is 25.12%.

Choosing the correct factor of commands has 49.19% importance on the system performance, while choosing the correct factor of classifiers have influence with 50.80%. There is as well, unexplained small variation of errors with 0.01% importance on the system performance.
5 Conclusion

This experiment took longer time than expected. From the start, we conducted knowledge about human brain and EEG to exclude possible errors and misunderstandings while designing our experiment. That was the most time-consuming part of the entire project leading to have neither sufficient time to analyze the blinking mode nor to implement a full working prototype.

We could see from the conducted knowledge about brain and EEG that 11% of people are not able to show alpha EEG signal reaction. Because of that we saw the need to record signals from more than one participant, and we have recorded signals from seven volunteers including the author.

By using Matlab’s Signal Analyzer toolbox we could see that the recorded signals of two volunteers were not showing power effect reaction while doing the exercises. Another two volunteers signal recordings gave significantly strong power effect, and the rest of the volunteers gave weak but visible power effect reactions.

The decision of how the exercises should be performed was based on the literature review, as well how we can construct the embedded system with Java server, Bluetooth communication and Matlab functions.

For this experiment we have used headset’s signal to find at least 2 classification patterns that could build commands to use in the full implemented system. The features that were used in classifications are segment’s frequencies and their PSDs obtained from each exercise, and the time of the segment. The best classification performed with 97.0% accuracy with Nearest Neighbor classifier. It was computed from the exercises; when a person is counting backwards and when a person is focusing on the point between the ears in the head. We investigated as well, the possibility of using several exercises to generate more commands, but the classification accuracy gives more accuracy with less number of exercises.

The analysis of classification result shows that the importance of choosing correct classifier is 50.80%, and choosing the correct command factor have 49.19% impact on the developed model. Unexpected variation based on the experimental errors has lower impact on the developed model with 0.01% error rate.

How many exercises can be recognized as commands?

- the classification with 8 exercises that could form 8 commands give 70% accuracy performance
- the classification with 4 exercises that could form 4 commands give 76% accuracy performance
- the classification with 3 exercises that could form 3 commands give 92% accuracy performance
- the classification with 2 exercises that could form 2 commands give 97% accuracy performance

We can conclude, out the answer of the research questions, that the two exercises can be used as two commands to control the window blind, i.e. to take down and up the window blind. The rest of commands can be computed by recognizing how many times a person blinks. As well as, we want to point out that the system can be built with only three commands, as illustrated in the flowchart in Figure 2.4. which means that we create two commands with thinking and one command with blinking.

The advantage of our experimental proposal is that it successfully design three or four commands to control the window blind. But the weakness is that there is still a need to create the model where the 11% of the people whose signals are not detectable by the headset, can use the headset as well.
6 Future work

Future work that we will do is to analyze the blinking and implement the full system, like it was planned from the start. Now we have seen that if we use features “PSD”, “Time” and “Frequency”, we will get high accuracy.

We can as well experiment with different features and time-windows to get higher classification accuracy. As well, we can use more segments of strong PSD for classification, but the data that is not interested should be filtered out.

Another aspect for the future work is to use other metrics, especially when analyzing the full system prototype. In that case we can use response time, reliability, energy efficiency, just some to mention.

The recorded data used for developing this system is saved and used only by the author. The data is marked with special combinations of letters and signs where only author knows who the volunteer for that particular dataset was. For future work, when implementing the full system we need to consider integrity of recorded data, and if there exist some research reasons to save the data for further project investigations. For example if we want to minimize errors in the system we would need to start with analyzing the recorded dataset.

Finally, we can state that this experiment introduced and opened world of other possible experiments as well. Just to mention one, we can investigate how the headset can be used for the 11% of people that don’t show EEG activity.
7 Referenser


8 Appendix A EEG Technology

8.1 Brain and its activity

Human brain consist of approximately 100 billion neurons. Neuron organizations and networks are shaped based on communications between neurons and their function purposes.

Three basic kinds of neuron cells, afferent neurons, efferent neurons and interneurons neurons have their own special functionalities when sending and receiving signals.

Afferent neurons take care of signals from periphery to the brain, efferent neurons carry information from the brain out to the periphery, while interneurons neurons work with signaling in local network circuits.

**Action potential** means that neuron is producing action potential established on chemical reaction in neuron membrane, commonly named that neuron “fires”. While in resting state the neuron is negatively charged in compared to the outside. The membrane potential in resting state is named resting potential and is around -70mV. When the neuron “fires”, the threshold is reached, named depolarization, and information is propagated along neurons. The action potential strength rises 100 mV (from -70mV to 30mV), see Figure 8.1. [21].

The information can be send between neurons via graded or action potentials. **Graded potential** is different from action potential in that way that the information is locally spread based on stimulus location. Graded potential have no threshold to reach, but the strength of potential depends on intensity of stimulus and is as a rule week. Another difference is that graded potential is becoming weaker and weaker i.e. consumed during the propagation of the information signaling, but the action potential is keeping the same strength during the propagation.

![Figure 8.1 Action potential.](image)

**Brain activity** refers to activity that is produced during communication between neurons where we can follow the signal that is creating **activity pattern.** Brain activity is also named **oscillatory activity.**

8.2 Human nervous system and their functional relationship

Anatomically, the nervous system is divided in two main systems presented as in Figure 8.2. [21]:

1. CNS central nervous system consists of the brain and spinal cord.
2. PNS peripheral nervous system consists of sensory neurons found on the surface of the body, peripheral, which have the purpose to sense and send signals to CNS for further information processing.
The brain consists of following parts: cerebrum (cerebral hemispheres), diencephalon (thalamus and hypothalamus), brain stem and cerebellum. The diencephalon is not visual in the Figure 8.3.

The cerebral cortex is the biggest area of the cerebrum which consists of grey matter containing neural cell bodies, which creates electrical activity we further use in scientific measurements. The four main areas of the cerebral cortex represent different sensory, motor and cognitive functions. Areas are named lobes, and have following names: frontal lobe, parietal lobe, occipital and temporal lobe [9] presented in Figure 8.3. [22]. Functions relative to the four areas are presented in the Table 8.1.

<table>
<thead>
<tr>
<th>Functional area- lobe</th>
<th>Associated functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frontal lobe</td>
<td>• Thinking skills</td>
</tr>
<tr>
<td></td>
<td>• Concentration</td>
</tr>
<tr>
<td></td>
<td>• Voluntary muscle movement</td>
</tr>
<tr>
<td></td>
<td>• Personality</td>
</tr>
<tr>
<td></td>
<td>• Language production</td>
</tr>
<tr>
<td></td>
<td>• Planning response for stimuli</td>
</tr>
<tr>
<td></td>
<td>• Planning and decision making</td>
</tr>
<tr>
<td>Temporal lobe</td>
<td>• Memory retrieval</td>
</tr>
<tr>
<td></td>
<td>• Recognize auditory stimuli</td>
</tr>
<tr>
<td></td>
<td>• Recognize olfactory stimuli</td>
</tr>
<tr>
<td>Occipital lobe</td>
<td>• Visual stimuli and visual interpretation</td>
</tr>
<tr>
<td>Parietal lobe</td>
<td>• Attending to stimuli (touch, pain, temperature, vibration)</td>
</tr>
</tbody>
</table>
8.3 EEG signal measurement

With electrophysiological (EEG) signal recording we can measure neural electrical activity in two different ways; placing the electrode inside the nerve cell or placing it near the nerve cell. By choosing to place the electrodes near the neuron cell, named extracellular recording, we can observe pattern of actions created, which in their turn represents different kind of behavior (see section Frequency bands identifying brain activity patterns). When placing electrode inside the nerve cell, we can observe smaller changes in electrical potential during neuron communication.

Hans Berger, German psychiatrist, (21 May 1873 – 1 June 1941) was the first to measure human EEG signals in 1920s, see Figure 8.4. [23], [9].

![Figure 8. 4 The first EEG recordings made by Hans Berger](image)

When measuring EEG, the electrodes are placed on the scalp. Only few electrode is enough to use for measurement. But to get more accurate result the number of electrodes used for measuring goes up to several hundreds. Other solutions for placing electrodes exists as well, like for example recording with one electrode, i.e. single-channel EEG recording [5].

Gained signal voltages are weak and must be amplified, by taking difference between one more electrodes that serves like a reference point placed on the head, or connected directly to the part of the head, i.e. earlobe.

For the purpose of mapping the signal/event to specific brain area, i.e. to explain behavior (mode), the signal is amplified, and send to the software for further processing like division in the frequency bands, calculated with Fourier transform formula (further explained in section BCI- Brain Computer Interface).

8.4 ERP- event related potentials

Three main classifications of systems of neural circuits exists: sensory system, motor system and associational system. Sensory system process information sensed from the environment. Motor systems respond to information captured from sensory systems, and associational system take care of complex brain functions behaving like a mediators between inputs and outputs system.

Any electrophysiological response of sensory, motor or associational neuron activity (event) is named ERP event-related potential, and the signals are measured with electrophysiological (EEG) signal recording, Figure 8.5. [24].

![Figure 8. 5 ERP- event related potential](image)
EEG signals exist always in the brain, measuring internal motivation and revealing the overall state of the brain. ERP signals are created in the EEG wave when the brain is reacting on some stimuli from the outside as well as inside stimuli.

While EEG signals are not measured in time to any particular event, the ERPs are telling us about specific cognitive function that appears which we can measure based on time. The ERP signal needs to be traced in time because of many changes in the potential that appears after the triggering. The ERP event is triggered when the time is 0, following with the signal that changes in positive and negative potential.

Positive potential peaks are represented with letter P, and negative peaks with letter N see Figure 8.5. The suffix number is telling us about the peak time in milliseconds, i.e. P100 (or P1) means positive potential peak occurred 100 milliseconds after event is triggered. The P300 is the one that is used for the purpose of creating/developing BCI (see section BCI) [3].

As presented in the Figure 14, the vertical axes is plotted with negative upward- this is because of historical reasons. It is important when doing research to examine what is negative versus positive indication on the graph.

ERP signals are weaker than the EEG signals, and the background noise deteriorates the quality of the signal. To overcome those issues we need to do multiple trials to extract ERP signal. The extraction process is done by finding repeated occurrences in the underlying raw EEG signal, created by the same event, and averaging the occurrences [9].

For the purpose of mapping the signal/event to specific brain area, i.e. to explain behavior, the signal containing the ERPs is amplified, and send to the software for further division in the frequency bans, calculated with Fourier transform formula (further explained in Appendix B).

### 8.5 International 10-20 system electrode placement

10-20 international system describes electrode placement locations of the scalp areas. 10-20 means that the distance is either 10 % or 20 % between electrodes counted from front to back of the scalp, and from right to left on the scalp Figure 8.6.

The letters are indicating cortical regions on the scalp. Suffix numbers are based on the midline of the scalp, where the odd numbers represent left side and even numbers the right side. The midline is represented by small letter z, Figure 8.6., (Purves, et al., 2012).

F- frontal area, C- central, P- parietal, O- occipital, T- temporal

![Figure 8. 6 The 10-20 system electrode](image)

### 8.6 Frequency bands identifying brain activity patterns

Since Hans Berger realized EEG recordings, the researchers have been studied activity patterns and their frequency patterns and found that different frequency ranges represent different stages of consciousness. Frequency ranges of the brain waves changes between 1-30 times per second. The observations were done by instructing the subject to do some task, while EEG signals are measured. There are 5 main frequency bands: gamma, beta, alpha, theta and delta waves, they all presented in Table 8.2.
Table 8.2 Frequency bands, associative activity and location

<table>
<thead>
<tr>
<th>Frequency band</th>
<th>Activity and location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gamma waves</strong>, ranging from 31 Hz and up.</td>
<td>This range represents activity associated with attention, perception and cognition.</td>
</tr>
<tr>
<td><img src="image" alt="Gamma waves" /></td>
<td>[25]</td>
</tr>
<tr>
<td><strong>Beta waves</strong>, ranging from 12 and 30 Hz.</td>
<td>This range is associated with focused concentration, and its activity is localized in central and frontal area.</td>
</tr>
<tr>
<td><img src="image" alt="Beta waves" /></td>
<td>[26]</td>
</tr>
<tr>
<td><strong>Alpha waves</strong> ranging from 7.5 to 12 Hz.</td>
<td>This range is associated with relaxation. The activity is localized in the back of the head and in the frontal area.</td>
</tr>
<tr>
<td><img src="image" alt="Alpha waves" /></td>
<td>[27]</td>
</tr>
<tr>
<td><strong>Theta waves</strong> ranging from 3.5 to 7.5 Hz.</td>
<td>This range represents the fine line between being awake and the sleep state.</td>
</tr>
<tr>
<td><img src="image" alt="Theta waves" /></td>
<td>[28]</td>
</tr>
<tr>
<td><strong>Delta waves</strong> ranging from 0.5 to 3.5 Hz.</td>
<td>This range represents sleep state.</td>
</tr>
<tr>
<td><img src="image" alt="Delta waves" /></td>
<td>[29]</td>
</tr>
</tbody>
</table>

8.7 Different techniques to measure brain activity

Different techniques exist for studying and measuring the activity and the structure. The different techniques are shortly presented hereunder. The measurement used for this projects, electrophysiological (EEG) recording, is already described in section EEG signal measurement.

**Computerized Tomography (CT)** - scanning technique forming 2D or 3D image presenting the brain structure with spatial resolution. The scanning is performed in hospital environment.

**Magnetic Resonance Imaging (MRI)** - uses strong magnets to create radio waves which results in released energy creating 3D image. The MRI is as well performed in hospital environment.

**Positron Emission Tomography (PET)** and **Single-photon Emission Computerized Tomography (SPECT)** - Here the particular reagents are injected into blood stream and later traced to get 3D image of how the metabolic activity looks like. In the PET the injection is for example water or glucose, and in SPECT the injections consists of radioactive compound.

**Functional Magnetic Resonance Imaging (fMRI)** - this technique is the same as MRI, but here we search for the area in the brain that represents some particular activity. When some task is performed we can see that oxygen level has increased in the activated brain area.
Magnetoencephalography (MEG)- in this technique the patient wear head helmet which reads magnetic changes of brain electrical activities. This technique is using same procedure for analyzing neural time series data as EEGs, but the recordings need to be performed in a hospital environment.
9 Appendix B BCI- Brain Computer Interface

9.1 What is BCI?
The Brain Computer Interface (BCI) system is a name for an interface which reads EEG signals and translate them to desired application placed on the device with specific function purpose.

Two main categories of BCI systems exist, invasive and non-invasive [7], [9]. In the invasive system the EEG signals is red from the electrodes implanted in the brain tissue. More about ethical aspects and laws is found in section Ethical aspects. The non- invasive system read EEG signals from the electrodes placed on the scalp. There is as well a mix of those two categories, where the BCI is half-invasive, or partial- invasive. In this case some of the electrodes are placed underneath the skull creating slightly better signal that non-invasive system.

9.2 Modeling single neuron
In the year 1939, Alan Hodgkin and Andrew Huxley, has done experiment on giant axon of a squid trying to understand action potentials. The result of the experiment was a success. The description of action potential was redrawn from an oscilloscope picture, describing a sharp positive increase of the potential and directly after a sharp decrease; which describe the membrane resting potential, depolarization and hyperpolarization (see Figure 8.1. action potential in the beginning of this chapter). Besides that, the experiment also describes mathematically models and equations.

The Figure 9.1. shows minimal mechanism for modeling membrane and action potential with an electrical circuit [30]. The model consist of three resistances, where $R_n$ and $R_k$ vary with time and $R_l$ is constant, as well as $C_n$. The $R_l$ and $E_l$ are together representing the leakage channel with constant conductance and resistance.
The capacitor $C_n$ is used to represent the equilibrium which is the goal for the membrane. Outside represents current coming from outside the membrane, and inside for intracellular currents, representing together signal direction.

The physical shape of the neuron is described with comparison to the long cable with its diameter and how resistive it is. The name for this theory is cable theory.

To model one neuron action potential takes time and is done when there is need for understanding the neuron in detail.
When modeling spike actions, the model is not created in detail, but is an approximation and simplification of the concerned action. For example, we don’t need to know the exact shape of the spike, but we need to know how long time the spike is lasting, as well as information about the input and/or output processes; before the spike is created and the process after the neuron is recovering/resetting after the spike.
Neural Models have been realized during the years, and the research field is still active. The most important models are presented here.

- **IF neuron** or **LIF neuron**, *leaky integrate and fire neuron model*- is model of the neuron when inputs are “leaking”, summarizing them and finally if the threshold is met, the firing of the neuron creates the spike. The resetting time after the firing is realized as well.

- **Spike-response model**- is kind of an IF model, but the difference is that IF model uses constant inputs and here the inputs streams are arbitrary, where time for input pulses plays a roll, which in their turn need to be calculated in different way.

- **The Izhikevich neuron model**- this model is easier to compute, and simulate, but at the same time keeps the same quality of variety of responses of real neurons. This model gave birth to three new models; *fast describing* (basic spiking similar to IF with small time constant), *regular spiking* (more common spike rate in the cortex) and *bursting spiking*. This model is used in network simulations.

- **The McCulloch- Pitts neuron model** - Edgar Douglas Adrian, was one of the first scientist to use first models of microelectrodes to record signals (in the 1920s). For his work, he got Nobel Prize in Psychology or Medicine. The model uses logical units for summarizing inputs, to calculate and compare the sum with the threshold; if the threshold is 1 the logical unit is activated, otherwise it is 0. The positive with this model is that we can measure the amount of spikes during an interval, but the negative is that time is not captured and thus cannot be used for more detailed calculations.

### 9.3 Conclusion to draw about the modeling

When modeling the neuron we can observe that different kind of models exist as well as the complexity, which makes a discussion of how difficult it can be to record the brain activity? Different places on the head can give the same signal, but the signal strength can have different levels of strength.

### 9.4 Construction of the BCI system

Several steps are performed during signal processing; data acquisition of the raw EEG signal, feature extraction and classification.

### 9.5 Data acquisition and segmentation

The sampling frequency shall meet Nyquist theorem, in other words to be able to capture frequency band the signals should be twice the signal, also being able to capture fast neural signals [31].

The book of [8], suggest that using 20-30 frequencies gives a good frequency range. Choosing more frequencies gives a possibility to select the appropriate frequency ranges to what we search for, but it also creates a large result matrices which take longer time to process.

If the work is to focus on for example alpha frequency band, the lowest frequency to analyze data can go down to 5 or 6Hz, and the highest frequency can be both theoretically and practically not higher than the Nyquist frequency. The literature of [8] suggest that many data points per cycle will increase signal-to-noise ratio. So, if sampling rate is 500Hz, and choosing the maximum frequency of 125 Hz provides 4 sample per cycle to analyze. In case that we want to analyze only alpha frequency band, the range of the closest low and high frequencies will be of interest to analyze, and not for example broad range of 150Hz.

How many trials are enough for analysis and how long each trial should be depends on the related research question. The literature of [8], suggest that approximately 50 trials per condition per person is giving a good level of-signal-to-noise ratio. The same book proposes that choosing the length of the signal, or how to record the signal length, should depend on the task to accomplish. A study of [6] recommends to record 5 to 10 examples per class to create good feature vector, but this is not practical because if using for example one electrode and sample at 250Hz, one trial, dimensionality will be 250 * 5= 1250 times for user to repeat the state. Of course this is not achievable, especially if recordings are made with multiple electrodes.
The raw data is cut in the segments or divided in time-windows differently depending on the research question, i.e. on how the signal shall behave in the experiment. Besides time-window, segment is also named epoch. If there is a stimuli, e.g. visual stimuli, the time-window should not just start before the stimulus onset, but it should also extract a baseline period before the stimuli onset. Also the length of the post stimuli depends on the research question. After the raw data is recorded it is saved in 2D matrix containing time and electrodes, but after the segmentation is performed the data is saved in a 3D matrix containing electrodes, time, segments, where the time is the length of each segments, and segments are the number of segments that the signal consist of. Usually, in the beginning and the end of the signal, there are edge artifacts which needs to be considered during segmentation [8].

Several different models exist to segment EEG data. The article of [32] discusses two methods: fixed-length segmentation and adaptive segmentation. Fixed –length segmentation consists of 4 steps: the signal is divided into equal minimal segments, where the next step will identify all and each of segments based on specific set of features, for example spectral estimation. In the third step, some of statistical methods are used to divide the segments into different classes, but in the last step boundaries between segments will be deleted. The resulting segments can contain stationary properties and others don’t, for example because of the transition state.

The second kind of segmentation process, adaptive segmentation is applied to the segments which consist of different variable length. Here two methods exists to be applied: parametric and non-parametric methods. Parametric methods, are applied when the pieces in EEG signal are of stationary structure, and is good to use when the process of the experiment in the study is known. Here the most used models are a kind of (AR) auto regressive model named (ARMA) auto regressive moving average model, as well as Kalman filter.

Non-parametric methods don’t have priori information about the sequence distributions probability. In this case, the method uses a strategy of change point detection, and the process is accomplished in 5 following steps: find random sequence which will serve as a sequence of detection of changes, creating a similarity hypothesis, do a preliminary check of change-point, continue with rejection of unclear change-points, and in the last step make an estimate of change-points [32].

9.6 Feature extraction

Different feature extraction algorithms exist, where 3 main sources of information can be used for further filtering/ extracting. The sources can be retrieved like spatial information, spectral information and/or temporal information.

- **Spatial information** is focusing on what information is retrieved from what EEG recordings channel, i.e. choosing features from channel of interest. The translation algorithms are performed filtering information from several channel sources with focus on one active area on the brain. Most used algorithms are Bipolar montage, Common average reference, Laplacian method and Common Spatial Patterns.

- Retrieving featuring from **temporal information** means that the features are retrieved directly from the EEG signal describing how the signal varies at different time points and in different time windows. This kind of features are found in time-domain. Mostly used algorithms are: Mean value, Standard deviation, Maximum Peak Value and Cross Correlation [32].

- **Spectral information** or information about the frequency; the features are choose based on the increase/decrease of power in frequency bands. The features can be found in frequency-domain and/or in time-frequency- domain.

**Time-frequency domain features**
- The features are describing how the power spectra varies over time.
- Most used transformation algorithms are Short Time Fourier transform and Wavelets transform [32].

**Frequency-domain features**
- this kind of features are also named band-power features
- the features are found as a power in the signal in specific frequency-band
• most used feature translation algorithms are
  o AR - Auto Regressive coefficients
  o PSD – Power Spectrum Density
  o Band-power
  o Asymmetry Ratio PSD

**Feature selection** is a kind of dimensionality reduction; the feature vector that is created after feature extraction process needs to reduce feature dimensionality, i.e. to delete features that are redundant. The resulting vector is named *reduced feature vector* [32]. Usually, the selection is made by dividing the feature vector in subsets, and to choose the best subset according to some specific criteria.

Another way of dimensionality reduction is named *feature projection*, where projection is the best chosen combination of the original feature vector values, further projected on a covariance matrix, on the eigenvectors, forming uncorrelated feature set.

### 9.7 Feature extraction process design

A study of [6] presents two basic BCI designs for extracting features; one to apply to oscillatory brain activity and the other to apply to ERP activity.

Feature extraction from oscillatory activity is performed in the way where the EEG signal is first band-pass filtered from frequency band from where we search power that varies. This process, to retrieve the power in interested frequency band, squares the same signal amplitude to calculate the signal power. And the third step is to average the signal over time, for example over a time window. The result is one feature, i.e. one band-power for that particular channel.

**Band– pass filtering step**
The recorded raw EEG data contains some noises and those need to be filtered away. Usual noise that follow with the recordings, is the noise from the electricity power, and is around 50Hz. Other filtering are concerned with filtering away low and high frequency ranges that are not of interest.

Before Fourier transform, the signal shall be time-segmented/ time-windowed, and use window function, like for example Hann window, Hamming window or Gaussian window [8]. There is no rule for how much the overlapping of the time-segments value should be, but recommendations are values between 50 and 90% of the length of the time-segments [8].

To analyze what frequencies exist in the EEG data, i.e. to extract frequencies in frequency domain we use Fourier transformation algorithm. We also get a voltage/amplitude that exists in the signal, i.e. the signal is presented as power of the frequencies which exists in the signal.

**Power squaring step**

After data acquisition and band-pass filtering, resulting vector contains the voltage which can be presented both in time-domain and in frequency -domain. The same vector is power squared achieving better view making the signal stronger/higher. Because the signals power is squared it is easier to further draw conclusion on what frequency band have a power that vary, increase or decrease.

Literature by [8] states that the effect/power of amplitude should be measured instead of the measuring the signal in time domain. The mean value of the signal in time-domain is giving smooth result, which means there are no effect peaks, as when using Fourier transform the amplitude is squared to the power, resulting in outstanding peak values at specific frequencies.

**Averaging the signal over time window**

This part of the processing is concerned with feature extraction so that feature selection vector can be created for further classification. Averaging of the signal over time is performed by moving time-windows.
9.8 Classification

Now the raw EEG data have been filtered and only features that are characterizing different classes are placed in the reduced vector. The next step is to use the reduced vector in classification algorithm to identify the class that the data belongs to.

There are two kind of classification learning, supervised (features can be identified with already existing algorithm) and unsupervised (features cannot be identified because there is no algorithm that will label the features to particular class).

There are different kind of classification algorithms used in BCI area. 7 classification algorithms are presented as mostly used: Neural Networks (NN), Bayesian classifier (BC), Fuzzy logic (FL), Linear discriminant analysis (LDA), Support vector machines (SVM), Hidden Markov model (HMM), K-nearest neighbor (KNN) and combinations of classifiers [32].

Following content explain briefly the characteristics of above mentioned 7 classification algorithms.

1. **Neural Networks (NN)**
   
The human brain consists of $10^{12}$ neurons, and the work to reconstruct neural network that behave like the real brain is a complex task to do. There are several different calculations applied for calculating, but the backpropagation is mostly used. The backpropagation algorithm is built on supervised learning, repeating the learning process and taking into account the error from the answer, adjusting the weights on input communication in next cycle of learning. This classifier can handle both linear and non-linear relationship, as well it is good at handling real-time constraints, which makes it very popular classification method.

2. **Bayesian classifier (BC)** - this is probabilistic classifier, where features are classified based on probability to belong to a certain class.

3. **Fuzzy logic (FL)** - this is a classifier which divides features to classes based on fuzzy logic, i.e. if not belonging to one class and not to the other it will belong to a class between, or the third one.

4. **Linear discriminant analysis (LDA)** - as already stated in this report, the LDA is very much used for feature extraction and dimension reduction of features.

5. **Support vector machines (SVM)** - this classifier is used for both the classification and regression. It is dividing features based on statistical linearity.

6. **Hidden Markov model (HMM)** - this model relies on Markov process model, where states are hidden (unknown), but the results from the states are known.

7. **K-nearest neighbor (KNN)** - this classifier is based on measuring the distance between features.

8. **Combinations of classifiers** - depending on the task different classifiers can be combined to get the best result.
Appendix C  ThinkGear™

ThinkGear Connector is a middle component that receive EEG signals through an open socket. The technology is placed on the headset to read EEG signals, to amplify signals, translate to binary raw data and send further through the ThinkGear Connector to present them to the developer by using its API. ThinkGear Connector needs to be started manually, and before the headset connection. The work is placed in the background thread, listening for headset connections request.

Through ThinkGear Stream Parser the developer can get the information, not just about the raw data, but also information about the subject’s mood, attention and meditation [19]. When receiving the EEG signal the ThinkGear uses an algorithm which calculates the signals and present the mood, or frequency band, or raw data that developer is asking for. Possible outputs [2] are presented:

- **poorSignalLevel**- integer value in range 0-200 indicating the quality level of the signal. If 0 the signal is good, if 200 indicate that the subject don’t wear the headset.
- **rawEEG**- the raw data that is either an integer or floating value. The raw data is actually a signed 16-bit integer, where the value can be in the range from -2048 and 2047. Streaming communication is at 57 600 baud rate.
- **eSense**- is a package representing meditation or attention levels based on the integer values, ranging from 1 to 100. Meditation level correspond to integer values found in the first half of the interval 1-100, and attention in the second half of the same interval.
  - attention- increased attention gives higher integer value.
  - meditation- increased meditation gives smaller integer value.
- **eegPower**- is a package containing 8 different EEG frequency bands. The output is represented with eight 4-byte floating point numbers
  - delta- (0,5- 2,75) Hz
  - theta- (3,5- 6,75) Hz
  - lowAlpha- (7,5 – 9,25) Hz
  - highAlpha- (10- 11,75) Hz
  - lowBeta- (13- 16,75) Hz
  - highBeta- (18- 29,75) Hz
  - lowGamma- (31- 39,75) Hz
  - highGamma- (41-49,75) Hz

10.1 ThinkGear packet structure

ThinkGear packet structure consists of three parts: Packet Header, Packet Payload and Payload Checksum, as presented in Table 10.1. The Header part is divided in two sync packet values (SYNC) and one packet length value (PLENGTH). The PLENGTH tell us the number of bytes (the length) of data payload. The (PAYLOAD) data can be up to 169 bytes long. The checksum packet (CHKSUM) is one byte long and is calculating the checksum to verify the validity of data payload packet parsing.

<table>
<thead>
<tr>
<th>Table 10.1 ThinkGear packet structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
</tr>
<tr>
<td>[SYNC]</td>
</tr>
</tbody>
</table>
The Figure 10.1. shows the output packets after streaming the signal and saved in streamLog.txt.

- The AA and AA are used for synchronizing the start of a new packet.
- 04 is (PLENGTH) indicating that the following data (PAYLOAD) will come in 4 bytes.
- 80 02 indicates that the raw wave value comes, and following columns represent the 16-bit hexadecimal code 00 0A for number 10 (see first row in the Figure 10.1.).
- The last part is the checksum, represented by number 73 in the first row in the Figure 10.1.

The raw value 00 0A found in the first row in the Figure 10.1. is displayed in the first row in dataLog.txt file, presented in Figure 10.2. In the same row we see decimal number 10 that represents 00 0A, just before it, marked with green ellipse.

For the purpose of this project we need to parse data stream to get bytes (raw value) and use it further for the classification purpose.

To convert raw value unit to the voltage value [2] use following equation:

\[
\text{Voltage value} = \left( \text{Raw Value} \times \frac{1.8}{4096} \right) \times \frac{2000}{2000}
\]
Appendix D Features and Classification Results

Classification have been done first with the 8 exercises (0.25 sec) and the accuracy results are presented in Table 11.1. Tables 11.2-11.7 present classification with two exercises. The number of combined exercises to classify are 28. More information about the exercises is found in Methodology part. In the end of Appendix D, there is Table 11.8 presenting accuracy performance on four exercises and on three exercises.

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>All 8 exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>44.0%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>26.2%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>20.2%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>14.9%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>15.6%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>15.8%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>9.8%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>12.8%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVN</td>
<td>28.6%</td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>20.2%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>18.8%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>29.4%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>29.6%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>16.8%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>15.9%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>29.4%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>28.7%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>30.1%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>70.0%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>15.3%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>0.0%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>29.3%</td>
</tr>
</tbody>
</table>
### Table 11. 2 Measured classifier accuracy for combinations with “Up” exercise

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>85.3%</td>
<td>90.1%</td>
<td>86.2%</td>
<td>83.6%</td>
<td>89.1%</td>
<td>89.7%</td>
<td>87.1%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>65.1%</td>
<td>85.8%</td>
<td>80.2%</td>
<td>77.2%</td>
<td>77.6%</td>
<td>78.9%</td>
<td>72.0%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>53.0%</td>
<td>73.7%</td>
<td>74.1%</td>
<td>64.2%</td>
<td>69.8%</td>
<td>71.1%</td>
<td>54.7%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear</td>
<td>45.7%</td>
<td>56.5%</td>
<td>55.2%</td>
<td>49.6%</td>
<td>55.2%</td>
<td>50.0%</td>
<td>50.4%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>45.7%</td>
<td>61.2%</td>
<td>59.1%</td>
<td>49.1%</td>
<td>55.6%</td>
<td>50.0%</td>
<td>50.1%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>49.1%</td>
<td>55.6%</td>
<td>56.0%</td>
<td><strong>55.2%</strong></td>
<td>56.5%</td>
<td><strong>53.4%</strong></td>
<td>50.0%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>46.1%</td>
<td>60.8%</td>
<td>62.9%</td>
<td>48.3%</td>
<td>55.2%</td>
<td>52.6%</td>
<td>52.6%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>53.4%</td>
<td>49.6%</td>
<td><strong>64.3%</strong></td>
<td>59.1%</td>
<td>50.9%</td>
<td>57.3%</td>
<td>51.3%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>49.6%</td>
<td>50.4%</td>
<td>51.7%</td>
<td>53.9%</td>
<td>45.3%</td>
<td>44.0%</td>
<td>50.0%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVM</td>
<td>54.7%</td>
<td><strong>68.5%</strong></td>
<td>62.5%</td>
<td><strong>70.3%</strong></td>
<td><strong>68.5%</strong></td>
<td><strong>81.0%</strong></td>
<td><strong>62.5%</strong></td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>48.7%</td>
<td>62.9%</td>
<td>64.2%</td>
<td>61.6%</td>
<td>66.4%</td>
<td>67.7%</td>
<td>56.9%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>47.0%</td>
<td>61.6%</td>
<td>58.6%</td>
<td>52.6%</td>
<td>55.2%</td>
<td>55.6%</td>
<td>53.4%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>68.1%</td>
<td>74.6%</td>
<td>66.8%</td>
<td><strong>81.5%</strong></td>
<td>77.2%</td>
<td><strong>86.2%</strong></td>
<td><strong>84.9%</strong></td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>66.8%</td>
<td>71.6%</td>
<td>70.3%</td>
<td>79.3%</td>
<td><strong>77.6%</strong></td>
<td>83.2%</td>
<td>76.3%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>44.4%</td>
<td>62.5%</td>
<td>58.2%</td>
<td>62.1%</td>
<td>66.4%</td>
<td>64.2%</td>
<td>49.6%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>49.1%</td>
<td>50.4%</td>
<td>56.0%</td>
<td>65.1%</td>
<td>71.1%</td>
<td>69.0%</td>
<td>58.2%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>66.4%</td>
<td>71.1%</td>
<td><strong>69.4%</strong></td>
<td>78.4%</td>
<td>77.2%</td>
<td>82.8%</td>
<td>75.9%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td><strong>69.4%</strong></td>
<td><strong>75.0%</strong></td>
<td>64.7%</td>
<td>81.0%</td>
<td>76.7%</td>
<td>83.2%</td>
<td>83.2%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>67.7%</td>
<td>87.5%</td>
<td>82.8%</td>
<td>80.2%</td>
<td>78.0%</td>
<td>85.8%</td>
<td>75.4%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>87.9%</strong></td>
<td><strong>91.8%</strong></td>
<td><strong>90.1%</strong></td>
<td><strong>87.1%</strong></td>
<td><strong>89.2%</strong></td>
<td><strong>89.2%</strong></td>
<td><strong>89.2%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>44.8%</td>
<td>55.2%</td>
<td>54.7%</td>
<td>55.2%</td>
<td>50.4%</td>
<td>53.4%</td>
<td>49.6%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>50.0%</td>
<td>34.5%</td>
<td>34.1%</td>
<td>35.8%</td>
<td>35.3%</td>
<td>12.9%</td>
<td>33.6%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>64.7%</td>
<td>83.6%</td>
<td>80.2%</td>
<td>77.2%</td>
<td>77.2%</td>
<td>78.9%</td>
<td>72.0%</td>
</tr>
</tbody>
</table>
Table 11. Measured classifier accuracy for combinations with "Down" exercise

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Down/ Balloon</th>
<th>Down/ Rect.</th>
<th>Down/ Eyes closed</th>
<th>Down/ Relax</th>
<th>Down/ Counting</th>
<th>Down/ Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td><strong>90.9%</strong></td>
<td><strong>92.2%</strong></td>
<td><strong>91.4%</strong></td>
<td><strong>85.8%</strong></td>
<td><strong>91.8%</strong></td>
<td><strong>89.2%</strong></td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>84.1%</td>
<td>84.1%</td>
<td>79.7%</td>
<td>78.4%</td>
<td>87.9%</td>
<td>67.2%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>72.0%</td>
<td>67.2%</td>
<td>71.6%</td>
<td>65.9%</td>
<td>62.9%</td>
<td>59.1%</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td><strong>59.9%</strong></td>
<td><strong>58.2%</strong></td>
<td><strong>51.3%</strong></td>
<td><strong>53.0%</strong></td>
<td><strong>43.1%</strong></td>
<td><strong>47.0%</strong></td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td><strong>59.1%</strong></td>
<td>56.5%</td>
<td>47.4%</td>
<td>55.2%</td>
<td><strong>42.2%</strong></td>
<td>47.0%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>57.8%</td>
<td><strong>59.1%</strong></td>
<td><strong>55.6%</strong></td>
<td><strong>54.3%</strong></td>
<td><strong>49.1%</strong></td>
<td><strong>52.6%</strong></td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>57.8%</td>
<td>56.5%</td>
<td>48.3%</td>
<td>55.2%</td>
<td>47.4%</td>
<td>44.8%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>42.7%</td>
<td>42.7%</td>
<td>50.9%</td>
<td>49.6%</td>
<td>47.8%</td>
<td>48.7%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>57.8%</td>
<td>42.2%</td>
<td>47.0%</td>
<td>51.3%</td>
<td>52.2%</td>
<td>47.4%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVN</td>
<td>65.1%</td>
<td>60.8%</td>
<td><strong>68.1%</strong></td>
<td><strong>67.7%</strong></td>
<td><strong>67.2%</strong></td>
<td><strong>50.0%</strong></td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td><strong>62.9%</strong></td>
<td><strong>63.4%</strong></td>
<td>58.6%</td>
<td>61.6%</td>
<td>62.9%</td>
<td>43.1%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>59.5%</td>
<td>55.6%</td>
<td>52.2%</td>
<td>51.7%</td>
<td>51.7%</td>
<td>43.5%</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td><strong>68.5%</strong></td>
<td><strong>66.8%</strong></td>
<td><strong>79.3%</strong></td>
<td><strong>81.5%</strong></td>
<td><strong>88.8%</strong></td>
<td><strong>77.2%</strong></td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>68.1%</td>
<td>64.7%</td>
<td>78.4%</td>
<td>73.7%</td>
<td>76.3%</td>
<td>68.1%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>61.2%</td>
<td>58.2%</td>
<td>52.2%</td>
<td>63.4%</td>
<td>56.0%</td>
<td>40.1%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>51.7%</td>
<td>53.9%</td>
<td>64.7%</td>
<td>66.4%</td>
<td>63.4%</td>
<td>57.8%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>68.1%</td>
<td>65.1%</td>
<td>78.4%</td>
<td>73.3%</td>
<td>75.0%</td>
<td>68.1%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>65.9%</td>
<td>62.1%</td>
<td>78.9%</td>
<td>81.4%</td>
<td>87.5%</td>
<td><strong>75.9%</strong></td>
</tr>
<tr>
<td>Ensemble</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>91.4%</td>
<td>88.8%</td>
<td>84.1%</td>
<td>84.9%</td>
<td><strong>91.8%</strong></td>
<td>75.0%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>94.0%</strong></td>
<td><strong>93.5%</strong></td>
<td><strong>90.1%</strong></td>
<td><strong>92.7%</strong></td>
<td><strong>90.9%</strong></td>
<td><strong>88.4%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>60.8%</td>
<td>60.8%</td>
<td>51.7%</td>
<td>47.8%</td>
<td>49.1%</td>
<td>48.3%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>51.7%</td>
<td>61.2%</td>
<td>9.1%</td>
<td>59.1%</td>
<td>33.6%</td>
<td>53.0%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>84.1%</td>
<td>84.1%</td>
<td>79.7%</td>
<td>78.4%</td>
<td>87.1%</td>
<td>64.7%</td>
</tr>
</tbody>
</table>
Table 11.4 Measured classifier accuracy for combinations with "Balloon" exercise

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Balloon/Rect</th>
<th>Balloon/Eyes closed</th>
<th>Balloon/Relax</th>
<th>Balloon/Counting</th>
<th>Balloon/Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>88.8%</td>
<td>89.2%</td>
<td>88.8%</td>
<td>89.7%</td>
<td>90.5%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>64.7%</td>
<td>85.3%</td>
<td>83.2%</td>
<td>78.0%</td>
<td>88.8%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>59.9%</td>
<td>67.2%</td>
<td>67.7%</td>
<td>67.2%</td>
<td>76.3%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49.6%</td>
<td>60.3%</td>
<td>62.1%</td>
<td>61.2%</td>
<td>61.6%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>50.0%</td>
<td>59.5%</td>
<td>61.6%</td>
<td>63.8%</td>
<td>59.9%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>52.2%</td>
<td>62.1%</td>
<td>61.5%</td>
<td>58.6%</td>
<td>58.6%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>52.6%</td>
<td>59.9%</td>
<td>66.4%</td>
<td>59.1%</td>
<td>62.1%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>50.9%</td>
<td>61.2%</td>
<td>50.0%</td>
<td>57.8%</td>
<td>54.3%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>45.7%</td>
<td>50.4%</td>
<td>46.6%</td>
<td>54.3%</td>
<td>59.5%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVM</td>
<td>52.2%</td>
<td>62.9%</td>
<td><strong>66.8%</strong></td>
<td>58.6%</td>
<td><strong>62.2%</strong></td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>53.9%</td>
<td>63.4%</td>
<td>62.1%</td>
<td>56.5%</td>
<td>59.5%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>51.7%</td>
<td>60.8%</td>
<td>60.8%</td>
<td><strong>60.3%</strong></td>
<td>57.3%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>62.1%</td>
<td><strong>81.5%</strong></td>
<td><strong>72.0%</strong></td>
<td><strong>75.0%</strong></td>
<td>67.2%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td><strong>66.4%</strong></td>
<td>70.3%</td>
<td>69.8%</td>
<td><strong>71.1%</strong></td>
<td><strong>68.1%</strong></td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>47.8%</td>
<td>54.7%</td>
<td>58.2%</td>
<td>56.0%</td>
<td>59.5%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>48.3%</td>
<td>57.3%</td>
<td>54.3%</td>
<td>55.6%</td>
<td>53.9%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>65.1%</td>
<td>68.5%</td>
<td>68.5%</td>
<td>71.1%</td>
<td>66.8%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>60.8%</td>
<td>79.3%</td>
<td>69.8%</td>
<td>72.8%</td>
<td>67.2%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>69.0%</td>
<td>88.8%</td>
<td>83.6%</td>
<td>82.8%</td>
<td><strong>90.5%</strong></td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>92.2%</strong></td>
<td><strong>91.8%</strong></td>
<td><strong>90.5%</strong></td>
<td><strong>93.5%</strong></td>
<td><strong>93.5%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>50.4%</td>
<td>63.8%</td>
<td>62.1%</td>
<td>59.1%</td>
<td>56.0%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>29.3%</td>
<td>33.6%</td>
<td>36.2%</td>
<td>30.6%</td>
<td>11.2%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>63.8%</td>
<td>84.5%</td>
<td>80.6%</td>
<td>79.7%</td>
<td>86.6%</td>
</tr>
</tbody>
</table>
Table 11.5 Measured classifier accuracy for combinations with “Rectangle” exercise

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Rect/ Eyes closed</th>
<th>Rect/ Relax</th>
<th>Rect/ Counting</th>
<th>Rect/ Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>86.2%</td>
<td>90.9%</td>
<td>85.5%</td>
<td>80.2%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>79.7%</td>
<td>79.7%</td>
<td>69.4%</td>
<td>62.5%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>67.7%</td>
<td>74.6%</td>
<td>62.1%</td>
<td>53.0%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td>58.6%</td>
<td>70.3%</td>
<td>58.6%</td>
<td>47.4%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>57.8%</td>
<td>58.2%</td>
<td>57.3%</td>
<td>47.3%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>59.9%</td>
<td>68.1%</td>
<td>58.6%</td>
<td>47.4%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>56.0%</td>
<td>67.7%</td>
<td>58.3%</td>
<td>47.8%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>45.3%</td>
<td>50.0%</td>
<td>44.8%</td>
<td>53.4%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>49.1%</td>
<td>52.6%</td>
<td>48.3%</td>
<td>50.0%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVM</td>
<td>63.4%</td>
<td>65.1%</td>
<td>56.0%</td>
<td>53.9%</td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>59.1%</td>
<td>59.9%</td>
<td>58.2%</td>
<td>53.4%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>57.8%</td>
<td>56.5%</td>
<td>58.2%</td>
<td>52.6%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>71.1%</td>
<td>68.5%</td>
<td>68.5%</td>
<td>74.1%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>67.2%</td>
<td>68.4%</td>
<td>65.1%</td>
<td>68.5%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>59.9%</td>
<td>59.9%</td>
<td>52.2%</td>
<td>48.7%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>59.1%</td>
<td>55.6%</td>
<td>56.5%</td>
<td>51.7%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>67.2%</td>
<td>67.7%</td>
<td>65.1%</td>
<td>68.5%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>68.5%</td>
<td>66.4%</td>
<td>63.8%</td>
<td>71.6%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>84.1%</td>
<td>87.5%</td>
<td>82.8%</td>
<td>64.7%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>90.9%</strong></td>
<td><strong>94.0%</strong></td>
<td><strong>90.1%</strong></td>
<td><strong>90.1%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>61.6%</td>
<td>61.6%</td>
<td>59.9%</td>
<td>46.1%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>61.6%</td>
<td>8.6%</td>
<td>30.6%</td>
<td>34.9%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>78.0%</td>
<td>80.6%</td>
<td>69.8%</td>
<td>60.3%</td>
</tr>
</tbody>
</table>
Table 11. 6 Measured classifier accuracy for combinations with "Eyes closed" exercise

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Eyes closed/ Relax</th>
<th>Eyes closed/ Counting</th>
<th>Eyes closed/ Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>80.6%</td>
<td>87.1%</td>
<td>93.5%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>64.7%</td>
<td>79.7%</td>
<td>90.1%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>59.1%</td>
<td>66.8%</td>
<td>64.2%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td>48.7%</td>
<td>49.1%</td>
<td>53.9%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>48.3%</td>
<td>49.1%</td>
<td>53.9%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>51.7%</td>
<td>50.0%</td>
<td>53.0%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>54.3%</td>
<td>44.0%</td>
<td>48.3%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>50.9%</td>
<td>45.7%</td>
<td>51.7%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>46.6%</td>
<td>48.3%</td>
<td>57.8%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVN</td>
<td>68.1%</td>
<td>73.3%</td>
<td>82.8%</td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>53.4%</td>
<td>66.4%</td>
<td>68.1%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>53.9%</td>
<td>54.7%</td>
<td>55.2%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>81.0%</td>
<td>80.2%</td>
<td>95.3%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>76.7%</td>
<td>79.3%</td>
<td>89.2%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>55.2%</td>
<td>60.8%</td>
<td>63.4%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>69.0%</td>
<td>72.4%</td>
<td>68.5%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>77.6%</td>
<td>78.9%</td>
<td>87.9%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>78.4%</td>
<td>80.6%</td>
<td>94.0%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>68.1%</td>
<td>79.3%</td>
<td>92.7%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>83.6%</strong></td>
<td><strong>87.1%</strong></td>
<td><strong>94.8%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>48.3%</td>
<td>49.1%</td>
<td>58.6%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>32.8%</td>
<td>32.8%</td>
<td>32.8%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>65.1%</td>
<td>79.7%</td>
<td>90.1%</td>
</tr>
</tbody>
</table>
Table 11. 7 Measured classifier accuracy for combinations with "Relax" and “Counting back” exercise

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Relax/Counting</th>
<th>Relaxe/Point</th>
<th>Counting/Point</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>87.1%</td>
<td>90.9%</td>
<td>91.8%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>67.2%</td>
<td>85.8%</td>
<td>79.3%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>61.2%</td>
<td>72.4%</td>
<td>70.3%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td>52.6%</td>
<td>43.1%</td>
<td>40.9%</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear</td>
<td>52.2%</td>
<td>43.5%</td>
<td>40.9%</td>
</tr>
<tr>
<td>2. Quadratic</td>
<td>48.7%</td>
<td>48.3%</td>
<td>55.2%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>52.2%</td>
<td>54.3%</td>
<td>51.3%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>40.9%</td>
<td>67.2%</td>
<td>42.7%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>44.4%</td>
<td>46.1%</td>
<td>50.9%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVN</td>
<td>74.1%</td>
<td>84.1%</td>
<td>78.0%</td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>62.5%</td>
<td>69.0%</td>
<td>70.3%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>51.7%</td>
<td>65.1%</td>
<td>52.6%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>90.1%</td>
<td>92.7%</td>
<td>97.0%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>82.3%</td>
<td>87.1%</td>
<td>86.2%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>58.6%</td>
<td>71.6%</td>
<td>57.8%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>71.1%</td>
<td>78.9%</td>
<td>67.2%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>81.9%</td>
<td>87.1%</td>
<td>86.6%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td>89.2%</td>
<td>91.8%</td>
<td>94.0%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>76.7%</td>
<td>89.7%</td>
<td>90.1%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td>88.8%</td>
<td><strong>94.8%</strong></td>
<td><strong>92.7%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>44.0%</td>
<td>38.4%</td>
<td>42.2%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>8.6%</td>
<td>43.1%</td>
<td>34.1%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>68.1%</td>
<td>85.8%</td>
<td>78.9%</td>
</tr>
</tbody>
</table>
Table 11. 8. Measured accuracy with four and three exercises

<table>
<thead>
<tr>
<th>Classifiers/Exercises</th>
<th>Point/Close eyes/Relax/Count back</th>
<th>Point/Close eyes/Relax</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Tree</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine Tree</td>
<td>62.5%</td>
<td>75.9%</td>
</tr>
<tr>
<td>2. Medium Tree</td>
<td>42.7%</td>
<td>62.6%</td>
</tr>
<tr>
<td>3. Coarse Tree</td>
<td>36.9%</td>
<td>50.3%</td>
</tr>
<tr>
<td><strong>Logistic Regression</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Discriminant Analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear Discriminant</td>
<td>22.8%</td>
<td>33.9%</td>
</tr>
<tr>
<td>2. Quadratic Discriminant</td>
<td>25.9%</td>
<td>36.8%</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Linear SVM</td>
<td>26.1%</td>
<td>36.8%</td>
</tr>
<tr>
<td>2. Quadratic SVM</td>
<td>24.4%</td>
<td>40.2%</td>
</tr>
<tr>
<td>3. Cubic SVM</td>
<td>23.7%</td>
<td>33.6%</td>
</tr>
<tr>
<td>4. Fine Gaussian SVM</td>
<td>54.7%</td>
<td>60.1%</td>
</tr>
<tr>
<td>5. Medium Gaussian SVM</td>
<td>42.5%</td>
<td>48.0%</td>
</tr>
<tr>
<td>6. Coarse Gaussian SVM</td>
<td>29.5%</td>
<td>42.2%</td>
</tr>
<tr>
<td><strong>Nearest Neighbor</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Fine KNN</td>
<td>72.4%</td>
<td>78.2%</td>
</tr>
<tr>
<td>2. Medium KNN</td>
<td>63.8%</td>
<td>76.4%</td>
</tr>
<tr>
<td>3. Coarse KNN</td>
<td>41.8%</td>
<td>44.3%</td>
</tr>
<tr>
<td>4. Cosine KNN</td>
<td>45.0%</td>
<td>52.3%</td>
</tr>
<tr>
<td>5. Cubic KNN</td>
<td>64.0%</td>
<td>75.6%</td>
</tr>
<tr>
<td>6. Weighted KNN</td>
<td><strong>70.0%</strong></td>
<td>76.1%</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Boosted Trees</td>
<td>48.1%</td>
<td>69.0%</td>
</tr>
<tr>
<td>2. Bagged Trees</td>
<td><strong>76.3%</strong></td>
<td><strong>92.0%</strong></td>
</tr>
<tr>
<td>3. Subspace Discriminant</td>
<td>23.1%</td>
<td>32.5%</td>
</tr>
<tr>
<td>4. Subspace KNN</td>
<td>22.2%</td>
<td>26.1%</td>
</tr>
<tr>
<td>5. RUSBoosted Trees</td>
<td>42.7%</td>
<td>62.6%</td>
</tr>
</tbody>
</table>
Appendix E Analysis and Evaluation

12.1 Experimental Design and Analysis

The experimental design that will be used to analyze the performance on the model is two-factor full factorial design. For this design two factors, features and classifier will be used. We have earlier in the report already investigated the combinations of the two factors, and we will use same set here to build full factorial design with all possible combinations.

The factor classifier has 5 levels, which are Decision Tree, Discriminant Analysis, Support Vector Machine, Nearest Neighbor and Ensemble. The factor feature has 3 levels which are different number of commands as presented in the Table 12.1. For the sake of this project, we will compute the experiment only once.

Table 12.1. Measured classifier accuracy for the two features

<table>
<thead>
<tr>
<th>Classifier</th>
<th>2 commands</th>
<th>3 commands</th>
<th>4 commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>0.9090</td>
<td>0.7590</td>
<td>0.6250</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>0.5520</td>
<td>0.3680</td>
<td>0.2500</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.7800</td>
<td>0.6010</td>
<td>0.5470</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>0.9700</td>
<td>0.7820</td>
<td>0.7000</td>
</tr>
<tr>
<td>Ensemble</td>
<td>0.9270</td>
<td>0.8020</td>
<td>0.7600</td>
</tr>
</tbody>
</table>

12.2 Computation of Effects

To do an analysis of the effects we need to arrange factors in a matrix similarly like we did in Table 12.1. First we calculated the sum of rows and columns, which gave an overall result for sum of rows and sum of columns.

After that, the mean value of each row and column is calculated and the overall mean value as well. The overall mean value is then used to compute the effect, by taking the difference between column/row mean and the overall mean values.

Table 12.2. presents the result of the analysis, where we can see that average of the classifier versus average of the features with 68.88% accuracy impact on the system.

Table 12.2 Result of effect analysis

<table>
<thead>
<tr>
<th>Classifier</th>
<th>2 commands</th>
<th>3 commands</th>
<th>4 commands</th>
<th>Row Sum</th>
<th>Row Mean</th>
<th>Row Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>0.9090</td>
<td>0.7590</td>
<td>0.6250</td>
<td>2.2930</td>
<td>0.7643</td>
<td>0.0755</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>0.5520</td>
<td>0.3680</td>
<td>0.2500</td>
<td>1.1700</td>
<td>0.3900</td>
<td>-0.2988</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.7800</td>
<td>0.6010</td>
<td>0.5470</td>
<td>1.9280</td>
<td>0.6427</td>
<td>-0.0461</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>0.9700</td>
<td>0.7820</td>
<td>0.7000</td>
<td>2.4520</td>
<td>0.8173</td>
<td>0.1285</td>
</tr>
<tr>
<td>Ensemble</td>
<td>0.9270</td>
<td>0.8020</td>
<td>0.7600</td>
<td>2.4890</td>
<td>0.8297</td>
<td>0.1629</td>
</tr>
<tr>
<td>Column Sum</td>
<td>4.1380</td>
<td>3.3120</td>
<td>2.8820</td>
<td>2.4890</td>
<td>0.8297</td>
<td>0.1629</td>
</tr>
<tr>
<td>Column Mean</td>
<td>0.8276</td>
<td>0.6624</td>
<td>0.5764</td>
<td>0.6888</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Effect</td>
<td>0.1388</td>
<td>-0.0264</td>
<td>-0.1124</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12.3 Allocation of Variation

Experimental errors are giving us information about how big the difference is between the estimated accuracy and measured accuracy. We have in previous section computed measured accuracy which is 0.6888.

To calculate estimated accuracy for the Decision Tree classifier on 1 segment feature, we get the following:
Estimated accuracy = measured accuracy + column effect + row effect
Estimated accuracy = 0.6888 + 0.1388 + 0.0755
Estimated accuracy = 0.9031

Similarly, we calculate the estimated accuracy for the rest of the combinations. The result is presented in Table 12.3.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Classifier</th>
<th>2 commands</th>
<th>3 commands</th>
<th>4 commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>0.9031</td>
<td>0.7379</td>
<td>0.6519</td>
<td></td>
</tr>
<tr>
<td>Discriminant</td>
<td>0.5288</td>
<td>0.3636</td>
<td>0.2776</td>
<td></td>
</tr>
<tr>
<td>Support Vector</td>
<td>0.7815</td>
<td>0.6163</td>
<td>0.5303</td>
<td></td>
</tr>
<tr>
<td>Machine</td>
<td>0.9561</td>
<td>0.7909</td>
<td>0.7049</td>
<td></td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>0.9905</td>
<td>0.8253</td>
<td>0.7385</td>
<td></td>
</tr>
</tbody>
</table>

The error computations, the differences between the measured and estimated accuracy are presented in Table 12.4.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Classifier</th>
<th>2 commands</th>
<th>3 commands</th>
<th>4 commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>0.0059</td>
<td>0.0211</td>
<td>-0.0269</td>
<td></td>
</tr>
<tr>
<td>Discriminant</td>
<td>0.0232</td>
<td>0.0044</td>
<td>-0.0276</td>
<td></td>
</tr>
<tr>
<td>Support Vector</td>
<td>-0.0015</td>
<td>-0.0153</td>
<td>0.0167</td>
<td></td>
</tr>
<tr>
<td>Machine</td>
<td>0.0139</td>
<td>-0.0089</td>
<td>-0.0049</td>
<td></td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>-0.0635</td>
<td>-0.0233</td>
<td>0.0215</td>
<td></td>
</tr>
</tbody>
</table>

Now we have calculated the difference between measured an estimated accuracy, and we can use the information to find variation of errors. We do that by using sum of squared errors equation, SSE:
SSE = each result in the table is squared and summed
SSE = (0.0059)^2 + (0.0232)^2 + (-0.0015)^2 + (0.0139)^2(0.0633)^2 + (0.0211)^2 + (0.0044)^2 + (-0.0153)^2 + (-0.0089)^2 + (-0.0233)^2(-0.0269)^2 + (-0.0276)^2 + (0.0167)^2 + (-0.0049)^2 + (0.0215)^2
SSE = 0.0015

To find SST, total variation, we first need to find the total variation of response SSY, and we do that by using the following equation: SSY = SS0 + SSA + SSB + SSE,

SS0 is calculated by multiplying factors level numbers with squared overall mean accuracy.
SS0 = 5 rows * 3 columns * (0.6888^2) = 7.1167

We calculated SSA and SSB by taking square of each factors mean and multiplying it with other factor’s level number.
SSA = ((0.8276^2) + (0.6624^2)) + (0.5764^2)*5 rows = 7.2797
SSB = ((0.7643^2) + (0.3900^2) + (0.6427^2) + (0.8173^2) + (0.8297^2))*3 columns = 7.5171

So, we calculate SSY:
SSY = SS0 + SSA + SSB + SSE
SSY = 7.1167 + 7.2797 + 7.5171 + 0.0015
SSY = 21.915

Now we use the following equation to calculate the total variation SST:
SST = SSY – SS0 = SSA + SSB + SSE
SST = 21.915 – 7.1167
SST = 14.7983

Now we know the total variation and we want to see how much each factor is important for the system. We do that by dividing each factor with the total variation and multiplying it with 100%.

The importance of factor features variation on the system:
100 * (SSA/SST) = 100 * (7.2797/14.7983) = 100 * (0.4919) = 49.19%

The importance of factor classifiers variation on the system:
100 * (SSB/SST) = 100 * (7.5171/14.7983) = 100 * (0.5080) = 50.80%

The unexplained variation based on experimental errors:
100 * (SSE/SST) = 100 * (0.0015/14.7983) = 100 * (0.0001) = 0.01%