Sudan University of Science & Technology

College of Computer Science &

Information Technology

Post Graduate Studies

EEG-Based Detection of Human Emotions

التعرف على مشاعر الإنسان بناء على الرسم الكهربائي للدماغ

Dissertation

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy in Computer Science and Information Technology

By: Mohamed Ahmed Abdulla

Supervisor: Prof. Lars Rune Christensen

September / 2020

ACKNOWLEDGEMENT

I am grateful to my mother and father Sumaya and Ahmed, who have provided me with moral and emotional support in my life. I am also grateful to my other family members and friends who have supported me along the way.

Special thanks to Professor Lars Rune for his wisdom, knowledge and support in every step of the way.

I would like to extend my thanks to my friend Mutasim for his help in delivering the payments for the conferences and for his major support for publishing the papers.

And last but by no means least, I am grateful to everyone in Sudan University of Science and Technology who had helped me with my PhD program, directly or indirectly.

ABSTRACT

EEG (Electroencephalography) allows eliciting the mental state of the user, which in turn reveals the user emotion, which is an important factor in HMI (Human Machine Interaction). Researchers across the globe are developing new techniques to increase the EEG accuracy by using different signal processing, statistics, and machine learning techniques. In this work we discuss the most common techniques that can yield better results, along with discussing the common experiment steps to classify the emotion, starting from collecting the signal, and extracting the features and selecting the best features to classify the emotions, along with highlighting some standing problems in the field and potential growth areas.

In this work, we have identifed 10 different emotions based on Valance, Arousal and Dominance using five different models. EEG signals are collected and passed to the proposed models, the accuracy of the detection was ranging from 50% to 70%. Two sessions have been conducted per subject to collect the data for training and for testing the models. Evaluation has been conducted to assess the new model performance, the evaluation is measuring the performance of the model on external data that looks similar in shape to the data conducted in this study experiment. The self-assessment manikin (SAM) assessment technique has been used to tag the training data, but the SAM model comes with its own challenges, therefore, theoretical model has been proposed to accommodate for the SAM challenges as well as making the experiment easier to conduct, but making the model architecture more complex.

مستخلص البحث

تخطيط أمواج الدماغ او ما يعرف بـ(EEG (Electroencephalography وهي إشارات كهربيه يبثها الدماغ تساعدنا في إستنباط الحاله الذهنيه للمستخدم والتي تتيح لنا التعرف على الحالة المزاجيه والذهنيه للمستخدم. تعرف على هذه الحاله الذهنيه من الأهميه بمكان, حيث أنها تتيح للآله التعرف على مشاعر الإنسان مما يسهل عملية التخاطب بين الآله والإنسان أو ما يعرف بـ(HMI (Human Machine Interaction. تقوم الأبحاث حول العالم بمحاولة تطوير تقنيات جديدة تحسن من جودة المعلومات المستنبطه من هذه الإشارات بإستخدام مفاهيم معالجه الإشارة والإحصاء والذكاء الإصطناعي. في هذا البحث سوف نناقش أشهر هذه المفاهيم, وأفضلها في إستخراج معلومات مفيده من الإشارات. وسنناقش أيضا انواع التجارب المستخدمه واكثرها إنتشارا والكيفية المستخدمة لإستخراج مشاعر الإنسان, إبتداء من جمع الإشارات وإستخلاص بعض الخصائص منها وإختيار أفضلها وتمثيلها في مشاعر في مفهومه قادرين على إستيعابها لإنها تندرج تحت نظام قياس محدد, وسنتطرق أيضا الى بعض المشاكل والصعوبات وسنذكر أيضا المناطق التي يمكننا تطويرها مستقبلا.

في هذا البحث قمنا بتصنيف 10 مشاعر مختلفه, هذا التصنيف بناء على النوعية, أو ما يعرف بـ(valence), والكمية أو الحدة, أو ما يعرف بـ(Arousal) وأخيرا التحكم او ما يعرف بـ(Dominance), في هذا البحث قمنا بتطوير 5 أنواع مختلفه من خوارزميات الذكاء الإصطناعي (machine learning models). تجمع الإشارات وتمرر إلى الخوارزميات الخمس, دقة المشاعر المخرجه كانت بين 50% و 70%. قمنا بعمل جلستين لكل متطوع مخرجات الجلسه الاولى إستخدمت لتدريب الخوارزميه مخرجات الجلسة الثانيه إستخدمت لإختبار الخوارزميه. قمنا بعمل إختبار لقياس الأداء, هذا الإختبار قائم على معالجة البيانات القادمه من بحوث أُخرى, هذه البيانات تشبه نوعا ما البيانات المستخدمه في هذا البحث.

قمنا بإستخدام إختبار المجسم الذاتي أو ما يعرف في الأوساط العلميه بـ(self–assessment manikin (SAM) لتحديد المشاعر ووضع علامات على الإشارات لمعرفة مشاعر المستخدم في لحظه قراءة الاشاره. ولكن هذا الإختبار لديه مشاكله الخاصه, لهذا قمنا بإقتراح نموذج علمي نظري يقوم بحل بعض المشاكل التي يتسبب بها نموذج SAM, و بعض الحلول الأخرى لتسهيل التجارب وجعلها أسرع وأسهل ولكن الخوارزميه ستتعقد أكثر.

TABLE OF CONTENT

List of Tables

LIST OF FIGURES

LIST OF APPENDICES

LIST OF PUBLICATIONS

Conferences

Paper 1: EEG Emotion Detection Review

Published in: March-2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)

Authors: Mohamed Ahmed, Lars Rune Christensen

Paper 2: EEG Emotion Detection Using Multi-Model Classification

Published in: Feb-2018 - International Conference on Bioinformatics and Systems Biology (BSB)

Authors: Mohamed Ahmed, Lars Rune Christensen

Journals

Paper 3: A Review of EEG Emotion Recognition

Published in: June-2019 – International Journal of Computer Trends and Technology (IJCTT)

Authors: Mohamed Ahmed, Lars Rune Christensen

Paper 4: Novel Model Architecture for EEG Emotion Classification

Published in: July-2019 – International Journal of Biotech Trends and Technology (IJBTT)

Authors: Mohamed Ahmed, Lars Rune Christensen

Paper 5: A Novel Theoretical EEG Machine Learning Model for Emotion Recognition

Published in: September-2019 International Journal of Computer Trends and Technology (IJCTT)

Authors: Mohamed Ahmed, Lars Rune Christensen

CHAPTER ONE

1. GENERAL INTRODUCTION

1.1. Introduction

In this chapter, we will discuss an introduction to the field and important topics we will be using in the studies, to get the reader at the same level of awareness. We will start by defining the EEG and the brain-computer interface (BCI) field, then we will show how we can measure emotion and how we will convert it to numbers to make it easy for the machines to digest, and finally we will run through some of the well-chosen researches in the field and acknowledge their work, at the same time differentiate between different types of brain wave signals.

1.2. Motivation

In the field of HMI (Human Machine Interaction), there is a desire to enhance the interaction with the machine if the machine is aware of the user emotion this knowledge will drive the field forward because it will enhance the communication between the user and the machine.

Understanding human emotion is a problem that can be approached from different perspectives. Early researches tried to detect the emotion from text and speech and facial images or facial depth images more recent researches tried to detect the emotion from ECG (electrocardiogram), temperature, or skin conductivity, EOG (Electrooculogram), heart rate, Eye blinking, and Heart Rate Variability (HRV). The biometrics shows high correlation with the human emotion especially brain signals but there are many ways to measure the brain activity. (M. Rajya Lakshmi, et al., 2014) There are several invasive and non-invasive techniques for understanding the brain signals such as EEG (Electroencephalogram), fMRI (Functional Magnetic Resonance Imaging), MEG (Magneto Encephalography), NIRS (Near-infrared Spectroscopy), PET (Positron Emission Tomography), EROS (Event-related optical signal). The different measurement technique has its advantages and disadvantages, in this work, we will focus on EEG Emotion detection. One of the main goals for automation is to enhance the interaction between the human and the machine, making it as fast as possible as intuitive as possible to the humans, and we can't do that without anticipating the human emotions and act upon it, knowing the human emotions will result in a more informed decisions when the machine is trying to react to the users commands, this will make the process much more convenient for humans (L. R. C. Mohamed Ahmed Abdullah, 2019). Machines cannot process human mood without identifying emotions. Scientists have been trying to do this from a very long time, they used all the possible means to elicit the human emotions from text or facial expression or other bio-signals. but the lack of training data what's always an impediment achieve high accuracy and provide a reliable solution to this problem (Qibin Zhao, et al., 2009). So, the purpose of this work is to enrich the field with some advances in the experimental setup and new Machine Learning ML techniques.

1.3. Research problem

The state of the art emotion detection is poor, the performance of the state of the art is not great in terms of accuracy and range of emotions. Knowing the emotion of the user will indicate to the machine an important information to tweak the behavior to get a better result, the current state of the art is not performing well to achieve this. Inspired by human to human communication, humans can communicate verbally and non-verbally, non-verbal involves many different cues, facial expression, distance, body posture, all of this is communicating different types of emotions, this is drastically improving the humans ability to communicate, in terms of verbal communication, for example, when you are telling a story to someone, you will adjust the length of the story based on the other human emotion feedback, if he is excited you will add more detailed, if he looked bored, you will try to finish faster, the ambition is to be better than humans, hence, there is a long way to go (Saeid Sanei, J.A. Chambers, 2007). As an analogy for the previous example, in the BCI (Brain Computer Interface) field, a frustration emotion is signalling that the output is not satisfying to the user, so the machine needs to take a different path to give proper feedback, a relaxing emotion is signaling the satisfaction of the current trajectory.

In order to get the human emotion we need an ML model that can accurately predict the human emotion and give reliable feedback, to be used in all the other types of system, and enhance the overall Machine to Human Interaction, whether it's in the manufacturing or the day to day work.

1.4. Research Questions

To design an emotion recognition system that can be used to detect emotions using the EEG signals and improves the human-machine interaction. This thesis concentrates on these questions:

- What's the current state of the art techniques to measure the emotions?
- How can we improve the range of emotions that can be detected and the accuracy of detection in respect of the range.
- What can we do better to enhance the field even more?

In these research questions, the researcher is seeking to address how to elicit the human emotions and how to improve the field, though the design of emotion recognition system, that could lead to improve the human-machine interaction and drive the field forward.

1.5. The potential benefit

Emotionally aware systems have the potential to enhance human-computer interaction hence improving how systems are used and the deliverables from those systems. Professionals in many domains like e-learning, e-commerce, or even gaming, can harvest the power of detecting the user emotional state and empower the domain with a new use case, and involve it in the design of the product. Emotion recognition systems have the potential of identifying human emotions better than other humans, as it has access to the bio-signals of the user. Many businesses can also use this technology to report on the quality of their product, by measuring how a subset of the users feel about the product, this product can be anything that is targeting the entertainment of the users, like video games or movies, or play, or even a material product (Picard & R. W., 1995). It can also be used as a public measure to reflect customer satisfaction, and help to sell the p roducts, like the International Standards Organization (ISO) standards as an analogy. All of the previously mentioned domains are big markets, but it can also be used for less significant areas like IoT and home appliances, by adjusting the environment to reduce stress. Emotion recognition systems might also have adverse implementations like creating systems capable of emotional manipulation of users, to adjust the mood or control it. Advances in emotion recognition systems will increase the awareness not only in computer science but in all areas of science, like psychology or clinical trials, for instance studying the emotions effect on memory, as well as help answering many still standing questions like how the shape of the bio-signal is connected to the emotion, which can enhance the clinical trials for instance.

1.6. Aims and Objectives

In a conjunction with the research problem sections and research question section, this research is aiming to investigate the current ways of measuring the emotions using the EEG signals and improve the emotion recognition and motivate other researchers in the Human-Machine Interaction (HMI) field and other fields as well. The main objectives are:

- Investigate the current approach for detecting the emotion, the experiment setup and reading the EEG signals, and the range of emotion being detected.
- Set up an experiment environment to collect accurate EEG data.
- Enhance the emotion detection methodology and the range of emotion being detected and the accuracy of the detection.
- Providing some future and theoretical enhancements in the experimental setup and the model architecture.

1.7. Methods

Before starting with emotion classifications, we have to discuss the definition of emotions and the wheel of emotions described in a psychological studies, this will allow us to be able to quantify and measure emotions. SAM (self-assesmennt Maniken) will be used to elicit the true user emotions, also measure the strength of the emotions, as well as measuring the accuracy of the survey before submitting it to the ML classifier. An experiment setup will be conducted to collect the EEG data readings. And finally, a novel ML classifiers and filtering that will run on the data to predict the result.

The apparatus being used is EPOC Plus with 14 channels: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4.

In the experiment setup 191 images have been selected to be used to trigger the stimuli in the tested subjects. The goal here is not to detect images, the images are only for stimuli purposes and we get rid of it after that, the goal here is to detect emotion which means that we need to link every brain pattern (brain waves) with the emotion that it represents.

The experiment starts with collecting the EEG data based on the photos that are automatically moving infront of the test subjects. The second step is to ask them how they felt while that image was presented, which is the self-assessment manikin SAM. The data has been collected during two sessions, the first for trainning the second for testing then the best channel "Electrode Selection" is selected in a step called "feature selectioin" followed by a setp called feature extraction, where the main methods in this step are: Riemannian Geometry extraction, xDAWN, Hankel, CSSP, CSP, and more but those are the main ones. Five models have been created to classify the emotions, all of them are based on the Logistic Regression. later in this wor we will discuss the evaluation and testing techniques.

1.8. Contribution

The main contribution for this thesis is to provide a way to detect the human emotions and to illustrate the challenges and the complexity of detecting the emotions and elicit it from the subjects, and the challenges of working in the domain of emotion recognition in general. As well as the attempt to increase the range of emotion being detected and increasing the accuracy of the detection.

This work adds to the methodologies of detecting the emotions and identifies issues and road map for further investigations and future development in the field. The contribution is important because enhancing the Human-Machine Interaction (HMI) and provide effective computing has a huge impact on people lives, in both commercial and day to day lives, with shading some lights on the benefits and the limitations.

1.9. The thesis orgainzation

In this chapter we discussed the general idea behind this this research as well as describing the aim and objectives. In chapter 2 we will list some of the other research areas in the field and their achievements to draw a picture of the current state of the art. Chapter 3 we will go throw the most useful techniques and methods has been used in the field to be able to distill them and choose the best approach, to be able to discover a better approach in chaprt 4, which is discussing the experiment setup and the novel technique that this work used to predict the emotions. Then we will go through a thorough inestigations and discusstions about how to test and evaluate this novel methodology in chapter 5. Chapter 6 will discuss how the model and the experiment setup can be inhanced in a detailed fashon. Chapter 7 will summarize the final discussions and conclusions and list the contribution of this study as well as providing some insights about the future inhancements. at last a list of references followed by the appendixes containing screenshots highlighting important softwares as well as the source code.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. BCI and EEG

2.1.1. Introduction

The neurons of the human brain process information by changing the flow of electrical currents across their membranes. These changing currents generate electric and magnetic fields. By putting electrodes to the scalp, the change of the electric field can be measured and recorded. Electroencephalogram (EEG) is a record of the electrical potentials between those different electrodes. EEG means the writing out of the electrical activity of the brain.

2.1.2. Evoked Potential

The evoked potential is the change in the brain's electrical activity that follows a stimulus. This potential is recorded by using EEG. Visually Evoked Potential (VEP) is recorded from the scalp over the virtual cortex. The visual cortex is under the occipital scalp. Light stimuli, e.g. flickering light or image, are used to stimulate the visual systems. These stimuli elicit a response in the visual cortex. This response is called the VEP. VEP reflects the attention (to the stimuli) and can determine the direction of the eye gaze. There is transient VEP and steady-state VEP. Transient VEP arises if the electrical excitation of the visual system is able to abate before new stimuli are presented. P300 is an example of a transient VEP. There is a resonance phenomenon arising in the visual attention to a flickering light source and this resonance happens in the Steady-State VEP (M. Congedo, et al., 2013). P300 means that the response in the brain waves is occurring after 300 milliseconds from the flickering light (R. Jenke & A. Peer and M. Buss, 2014). Steady-State Visually Evoked Potential (SSVEP) is the electrophysiological response of the visual cortex to a rapidly repeating (flickering) stimulus. This response generally has a sinusoidal waveform with the same temporal frequency as the driving stimulus. This means that SSVEP is periodic. The periodic pattern of SSVEP can be detected and correlated to the periodic pattern of the stimuli. SSVEP reflects the attention to a rapidly oscillating stimulus (Minho Kim, Byung Hyung Kim, and Sungho Jo, 2015).

2.1.3. BCI, EEG, HMI Field

Recording the EEG signals is done by placing electrodes on the scalp of the subject. This will measure the voltage by microvolt (mV) with a certain frequency Ex. 100 Hz. The recordings may be mono-polar or bipolar. Mono polar is simply measuring the voltage, bipolar is the voltage difference between two electrodes. The mono-polar recording is more popular. The placement of the electrodes is called 10-20 position system, which is proposed by the International Federation of Societies for Electroencephalography and Clinical Neurophysiology (Saeid Sanei, J.A. Chambers, 2007).

EEG is one of the simplest techniques for recording the brain's electrical activity, compared to the other none invasive techniques like fMRI functional Magnetic Resonance Imaging and NIRS near-infrared spectroscopy and Magnetoencephalography MEG etc. it is there for widely acceptable and well known as a rising tool in the development of HMI Human-Machine Interaction field and BCI Brain-Computer Interface field humans (R. Jenke & A. Peer and M. Buss, 2014) the success on the BCI relies on the accuracy of feature extraction from the EEG signals, this signals contains distinguishable features collected from the person's brain and it represents various kind of movement or eye blinking or emotion swings. The higher the accuracy of predicting the classes and the higher the number of classes that can be predicted are the better the BCI system is, in other words, if the BCI system can predict a wide variety of classes with high accuracy the better.

In recent years the field of BCI is getting bigger and the researchers started to adopt this field more for both healthy and clinical populations but still, there is no commercial adoption of the BCI technologies due to the lack of robustness when translating these technologies to the real world (uncontrolled environments) beyond the research setup and laboratory. The method and techniques and processing and classification algorithm that has been invented since the inception of the field about 20 years ago of EEG inverse solutions and diagonalization methods such as the common spatial pattern CSP (Minho Kim, Byung Hyung Kim, and Sungho Jo, 2015), canonical correlation analysis CCA, independent component analysis ICA (B. H. Kim and S. Jo, 2018), with the multiple variations of each other's and possible combinations, one can see that there is no further major innovation in the field. Can also say a new method based on these methods and techniques deliver only moderate improvement and do not increase reliability in a significant way. There are multiple subfields in the BCI causing the research to be dispersed and not focused on a single problem, the fact that there are three subfields of BCI modalities, namely, motor-imagery (MI), steady-state evoked potentials (SSEP) and P300 (R. Jenke & A. Peer and M. Buss, 2014), is currently treated with dedicated pre-processing, signal processing and classification tools they all interchanged and faces of the same issue. The classification techniques are also dispersed and fragmented. We can divide the existing techniques of classifications in two categories, first classification techniques that follow a “hard machine learning” approach, second classification techniques that use “spatial filtering” to increase the signal-to-noise ratio SNR followed by a simple classification algorithm. The “hard machine-learning” approach generalizes well across sessions and across persons, but it requires a large number of training data. Also, it is always computationally expensive. But the opposite happens for the “spatial filtering” approach, in fact, it’s bad in the generalization capabilities but fast for training the models due to the simplicity of the model and the lower volume of training data and lower computational cost. In light of this situation, it has been stated that “the field would benefit from a new approach in research development that focuses on robust algorithm development”. It has also been recommended to start regarding the pre-processing, feature extraction and classification not as isolated processes, but as a whole (M. Congedo, et al., 2013).

2.1.4. Motion Imagery (MI)

One of the simplest techniques for recording brain’s electrical activity Electroencephalogram is and it’s considered as a non-invasive tool for measuring the brain activity, compared to the other non-invasive neuroimaging procedures of functional Magnetic Resonance Imaging (fMRI), Near-infrared spectroscopy (NIRS), Magnetoencephalography (MEG) etc. It is therefore widely accepted as a promising tool in the development of man-machine interaction systems of Brain-Computer Interface (BCI) technology. The success of BCI based communication heavily depends on the efficacy and robustness of the distinct features extracted from EEG (Thomas & P., 2017). EEG is capable of capturing distinguishable features from the brain associated with a person’s various kinds of movements, actions and thoughts. The higher the accuracy in the decoding of subject intentions is, the higher the performance of any BCI system. One of the most commonly used brain patterns in EEG-based BCI is motor imagery, which is the mental rehearsal of motor movements without execution. For example, if the person moved his arm there is a particular area in the brain light up. This paradigm is popular because of its simplicity and easiness to be performed even by a locked-in patient suffering or surviving from neurological diseases such as stroke, Parkinson's disease etc.

There is different types of motor imagery (MI) (imagination of movement of the left hand, right hand, foot, tongue etc.) generate distinct signatures in EEG and can be identified by trained medical personnel but most importantly it can be identified by efficient signal processing and robust machine learning algorithms. The identified EEG features can be mapped to command signals to control an external device or computer applications, for example, if the person moved his right arm that can be identified and the conversion of this knowledge to a command is called mental command, which will fulfill the ultimate goal of BCI technology, (Minho Kim, Byung Hyung Kim, and Sungho Jo, 2015)

'Transforming human thoughts into actions'. Motor imagery (MI) training has an additional therapeutic benefit too because it can activate neuronal populations associated with motor control, which in turn can help to improve the motor skills of even a stroke survivor. Therefore, the study on brain patterns related to motor imagery is important in BCI research progress.

Many studies in the literature discuss motor-imagery related Event-related desynchronization (ERD) and imagery related Event-related synchronization (ERS) patterns in EEG, their subject-specificity, spectral, spatial and temporal variabilities, the influence of neurological pathology etc., that have to be carefully considered while developing motor imagery based BCI. The ERD/ERS refers to the power decrease and increases in EEG signals accompanied by motor imagery tasks, of which the spectral, spatial and temporal components vary over subjects. Though the discriminative frequency components mainly appear in the mu (8-12 Hz) and beta (12- 20 Hz), significant inter-subject variability has been reported in the literature, beta waves have to be generated using the frequency band algorithm first there is another name for it it's also called the wavelet transformation (A. Samara & M. L. R. Menezes and L. Galway, 2016). On account of this fact, techniques to tackle these variabilities can improve the classification performance of BCI systems. A fisher ratio based discriminative subject-specific frequency band selection algorithm and reports good classification performance by addressing the inter-subject variability of discriminative frequency bands during motor imagery. Discriminative frequency bands are selected only from single motor cortex channel 'C4' which have then been applied on all motor cortex channels. However, the discriminative weight of frequency bands can vary across channels for the specific motor imagery tasks. A hybrid channel selection method that selects highly discriminative channels in the motor cortex region based on Fisher ratio analysis and investigates the impact of employing channel-specific bands in motor imagery classification can give a better result. After selecting the informative channels and frequency components, then calculating the bandpass filtering and machine learning techniques to classify left and right-hand motor imagery. So the steps will look like figure 1 (Thomas & P., 2017).

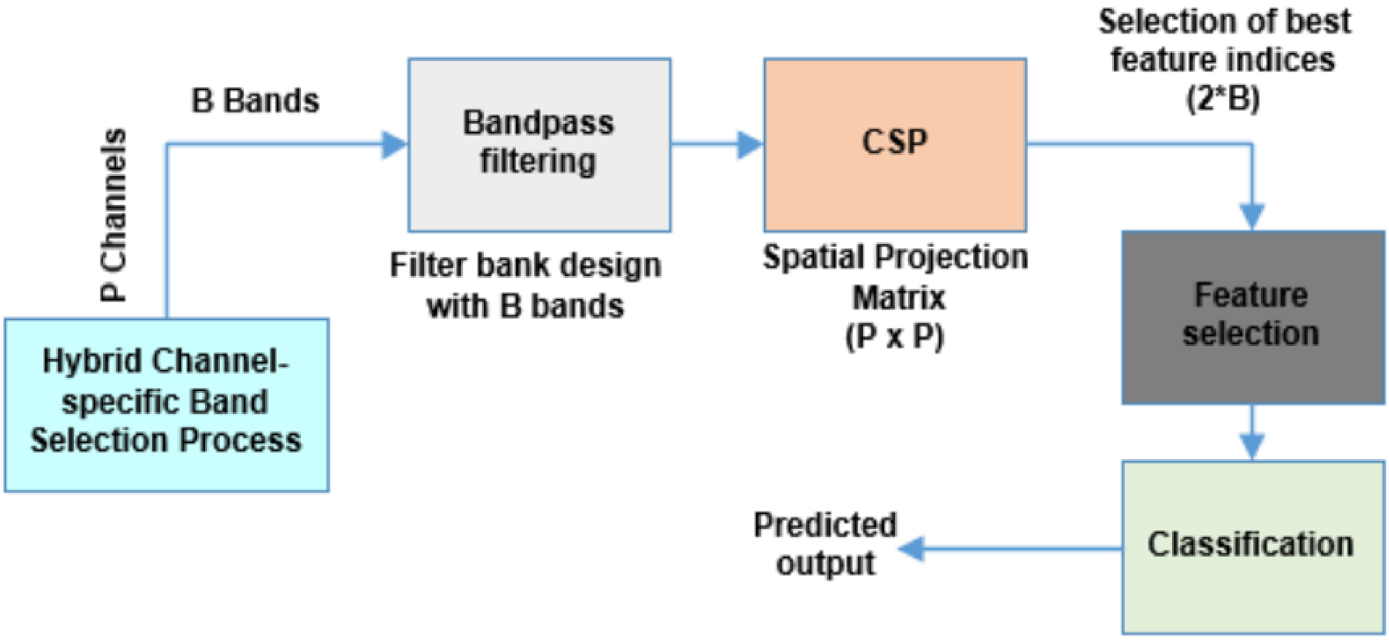


Figure 1 Signal processing stage after channel/band selection (Thomas & P., 2017)

2.1.5. EEG Data Format

There is a wide variety of data formats to store the Bio-signals, BioSig is an open-source software library, supports approximately 50 different data formats. Many data formats have been developed by commercial companies, research groups and standardization organizations. For EEG recording and the Brain-Computer Interface research, some data formats can be used flexible data exchange between organizations or institutes, e.g. (GDF) General Data format, (EDF) European Data Format, (BDF) Biosemi Data Format, BKR and so on. Different electrodes in the scalp are sending EEG signal data so it is mostly multichannel data. The EEG signals are always stored based on channel and time. There is a sampling frequency for each EEG data signals, those signals are stored later on in a timely fashion. EEG data can be stored as channel-based order or time-based order or combination of both. Channel-based order means that the data is stored channel by channel and each channel contains data from sample 1 until sample Ns, with Ns = the number of samples. Most EEG amplifiers basically provide channel-based order recording but due to memory limitation, the signals cannot be all buffered so a combination with time-based order is needed. Time-based order means that the data is stored sample by sample timely and each sample contains data from channel 1 until channel Nch, with Nch = the number of channels. Some examples of time-based order recording are BKR format and Implicit Association Test (IAT) BrainRobot EEG data format.

EEG Data is recorded first then buffered by the EEG amplifier before it is sent to a computer to be stored. The buffer size is always a constraint. Although most EEG amplifier record data in channel-based order, the limited buffer size make the data storing always in a timely fashion.

In BKR format the record is not in time-based order, a record is based on experiment or trial. One BKR file may contain data from different kinds of experiments, for example in spelling experiments, the first trial is spelling “BCI” then the second trial is spelling “CHUG” and so on. Therefore trail is the set of signals or data that represent a mental state or a response from a stimulus. Each experiment is stored as one record and all records are arranged in only one file. There is also a different perception of the number of channels and the number of electrodes. EEG signals come from different electrodes. Due to many kinds of signal processing, e.g. spatial filtering, the number of channels is not always the same as the number of electrodes. For example, EEG signals from 6 electrodes can be only 1 channel if the averaging filter is used. Other examples of spatial filtering can be seen.

2.2. REVIEW Experiments

Emotion detection is a standing problem for a long time most of the researches in this area using the same approach for tackling this problem with a small variation. The steps look like this:

1- Emotion stimuli materials (images, videos, audio, game) there is a number of emotion stimuli databases can be used to trigger the emotion into the subject

2- The emotion may differ from subject to another that’s why the studies relying on the feedback from the subject, the intensity of the emotion may differ as well that’s why most of the studies are using variations of SAM (Self-Assessment Manikin) to rate pleasure, arousal, and dominance

3- Filtering and selection of the EEG data

4- Classification: the most common classifier is the SVM but depending on the study researchers tend to use the most relevant classifier for their data.

In emotion recognition from EEG, it is not generally agreed upon which emotion to detect. Deferent studies trying to detect deferent types of emotion:

1- Discrete emotion: (happy, sad …) in this set of studies researches take a single emotion to detect or opposite set of emotions

2- Positive Negative

3- Arousal Natural: low arousal high valence (LAHV), high arousal high valance (HAHV), and high arousal low valence (HALV)

2.3. Related Work

This study (G. Cheng, 2017) tried to predict the human emotions in order to create a closed-loop of playing music stimuli and measure the emotion to increase awareness learning recovery or enhance the brain functionality or be used as a treatment, the accuracy of the model has been collected by a user’s survey by asking them to develop a mental strategy then measuring the success of achieving the targeted strategy. The emotion in this study is measured by the valance and arousal.

In this study (S. Jirayucharoensak, et al., 2014) a new (DLN) Deep Learning Network has been proposed to predict the colouration between the input signals three auto-encoders has been used along with two softmax for valance and arousal classification and by using PCA on the initial input this resulted in a model that is 49.5% accurate of identifying 3 levels of valance and arousal, also the DLN model scored better than SVN and Bayes classifiers.

This Study (X. Li, 2016) a hyper deep learning model has been proposed to link CNN and RNN (Convolutional Neural Network and Recurrent Neural Network) the CNN has the ability to find the correlation between input signals while the RNN has the ability to learn long term dependencies, it also has a potential in giving predictions not only for an entire trial but also for each time step, which is very important in real-time emotion monitoring scenarios.

The DLN models are easier to set up and to construct not like the traditional models by shifting the task of finding the best signal correlation load to the layer of the neural network.

(H. Zhang, 2017) Major Depressive Disorder is a mental disorder can be detected from the amygdala region, In this study, an emotional up-regulating method has been applied to healthy individuals by performing set of tasks then self-report the affective state the tasks e.g. recall a happy memory and the accuracy will be measured by t-test the results from using the CSP was 72% as mean from an 11 session.

(S. I. Alzahrani, 2016) The goal in this study is to spell out the words the results can reach 70% after a certain training session per user. The modes being used is linear discriminant analysis LDA and SVM support vector machines.

(W. L. Zheng, et al., 2018) In this study an eye movement has been collaborated with the brain waves with 6 electrodes best mean accuracy of 85.11% has been achieved for four emotions (happy, sad, fear, and neutral). The study used film clips as the stimuli and conducted on 15 subjects 75 films has been selected each film duration is 2 minutes more than one session has been conducted each 24 films is considered a session SAM has been conducted, the EEG electrodes are FT7, FT8, T7, T8, TP7, and TP8 placed using the 10–20 system. SVM has been used for classification and bimodal deep auto-encoder (BDAE)

Table 1: Studies on emotion detection

#	Study	Range of emotions	Electrodes	Feature Extraction	Classification	Experiment	Result
1	2017	Excited, Relax, Sad, Average (neutral)	AF3, T7, T8, AF4 with 128 Hz	Wavelet	Support Vector Machine (SVM) and Learning Vector Quantization (LVQ)	10 subjects, emotion database, morning, noon, and night,	Every 10 second, the detection accuracy Excited 88%, Relax 90%, Sad 84%, Average 87%
2	2016	Anger, Surprise, Other	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 and AF4 with 128Hz	Short-Time Fourier Transform (STFT) window 1sec, (mRMR)	Random Forests (RF) and (SVM)	DEAP	RF better than SVM
3	2016	Valence, Arousal	Fp1, Fp2, F3 and F4	Wavelet + Basic Statistics Higher Order Crossing (HOC)	SVM	DEAP dataset	Valence 82%, Arousal 76%
4	2016	Happy, angry, sad, neutral	10-20 system	Principal Component Analysis (PCA) Fourier Transform (FT), Short-time Fourier Transform (STFT) slow cortical potential (SCP) and Wavelet Packet Transform (WPT) Discrete Wavelet Transform (DWT)	radial basis function neural network (RBFNN) and SVM	5 subjects	HOC is the best
5	2017	- high arousal, high valence x - low arousal, high arousal, high valence x - low arousal, low valence x - high arousal, low valence	32	Statistical Feature, Hjorth Features, Non-Stationary Index, Higher-Order Crossings,	Relief algorithm, Bhattacharyya distance	DEAP, 32 subjects, 40 Music videos	the statistical feature is more powerful

#	Study	Range of emotions	Electrodes	Feature Extraction	Classification	Experiment	Result
6	2016	positive or negative	32 128Hz	spatial filter of common spatial pattern (CSP)	ElasticNet, LDA, QDA, SVM	set of movie clips, 10 subjects	spatial filters better than conventional methods (CSP)
7	2012	low arousal low valence (LALV), low arousal high valence (LAHV), high arousal high valence (HAHV), and high arousal low valence (HALV)	**3 channels**	adaptive way (ASI-based algorithms)	quadratic discriminant analysis (QDA), Mahalanobis distance (MD), -nearest neighbour (-NN), support vector machine (SVM),	16 subject, IAPS	AsI-based algorithms introduced performin better than simple SVM
8	2015	happy, calm, sad, and scared	Fpl, Fp2, C3, C4, F3, and F4	No Extraction	Three layers of restricted Boltzmann machines (RBMs)	21 subjects	better recognition accuracy than KNN, SVM ANN
9	2014	(LALV), low arousal high valence (LAHV), high arousal high valence (HAHV), and high arousal low valence (HALV)	64	Wavelet	kernel Fisher's discriminant analysis (KFDA) kernel eigen-emotion pattern (KEEP	10 volunteers	Novel feature extraction method KFEP bette than normal classification
10	2012	Arousal detection (strong vs. calm), and valence detection (positive vs. negative).	FP1/FP2, F7/F8, F3/F4, FT7/FT8, and FC3/FC4.	asymmetric features and filter bank common spatial pattern (FBCSP) as a benchmark and proposing Recursive Fisher linear discriminant (RFLD)	K-nearest neighbour (KNN), Naive Bayes (NB), and support vector machine (SVM)	video clips last for less than 20 minutes, 4 subjects	Novel feature extraction method
11	2014	happy, sad and neutral)	Twelve channels (AF3, F7, F3,	short-time Fourier transform (STFT) 1sec window. differential laterality (DLAT)	Gaussian Naïve Bayes (GNB)	Music listening (24 trials per day).	Comparing feature selection techniques

#	Study	Range of emotions	Electrodes	Feature Extraction	Classification	Experiment	Result
			FC5, P7, O1, O2, P8, FC6, F4, F8, and AF4)	and differential causality (DCAU)			
12	2015	Regret, rejoice, other emotion	64	Approximate entropy (ApEn)	Fisher Linear Discriminant (FLD)	25 subject using a gambling paradigm	Extracting the regret emotion from the sign
13	2107	‘anger’, ‘contempt’, ‘disgust’, ‘fear’, ‘sad’, ‘surprise’, ‘happy’.	14 channels	Wavelet	Mel-frequency cepstral coefficient (MFCC) multilayer perceptron (MLP)	IAPS	A human can have more than one emotion a time
14	2018	positive/negative and the approach/withdra wal	F8, FC2, FC6, C4, T8, CP2, CP6, and PO4, F7, FC1, FC5, C3, T7, CP1, CP5, PO3	DEEP PHYSIOLOGICAL AFFECT NETWORK (Deep Learning model)	multi-layer convolutional neural network (CNN)	1,280 videos, along with the 64 combinations of physiological signals per video.	Deep neural network for features extractior
15	2016	Normal, Abnormal	64	Wavelet, Discrete wavelet transform (DWT)	feed forward back propagation	10 subjects	75% for normal and 65% for abnormal
16	2014	Valence, Arousal, Liking with Positive Negative for each	32 and 10	Band power and PSD by Wavelet Transform	Support Vector Machine (SVM)	32 participants DEAP	The best combination is one-minute EEG da using band power filter from 10-channel probes
17	2005	arousal, valence, dominance and liking	32-channel	Gaussian Mixture Model and wavelet	linear ridge regression and support vector regression (SVR)	40 one-minute long music videos And let then score dominance (on a scale from 1 to 9) and familiarity (on scale 1 to 5). From DEAP dataset	unsupervised training have better results than traditional classification

2.4. Comments on the Studies of emotion detection

1. In this work (Esmeralda C. Djamal & Poppi Lodaya, 2017) as you can see in the table it's trying to detect 3 emotions, the experiment has been done on 10 subjects but only 10 trial per subject, it's using music to evoke the emotion into the subjects, they used Wavelet as filtration for the signals and Learning Vector Quantization LVQ as classification, LVQ is using neural net, but it's considered as a precursor to self-organized maps SOM and also precursor to K-nearest neighbour K-NN. They reach a high accuracy as they are detecting only 3 emotions, as well as wavelet is known for its simplicity, but it will hide some information from the classifier, which means it's not good if you are trying to detect wider range of emotions.

2. In this work (P. Ackermann, et al., 2016) the special thing about this paper is that it's comparing two filtering techniques and two classification algorithms, the filtering techniques are Min-Redundancy-Max-Relevance (mRMR) and Short Time Fourier Transform (STFT) and the classification algorithms are SVM and Random Forests (RF), the classification seems to decline when using a large number of features, but overall it's more robust. This mixed model is important to understand the contrast between using different filtering and classification algorithms.

3. This work (A. Samara & M. L. R. Menezes and L. Galway, 2016) main contribution is rehearsing the emotion detection in respect of the Circumplex Model of emotion. The avert from detecting pinned emotion and using the valance, arousal, and dominance is considered to be a step forward.

4. This work (A. Patil & C. Deshmukh and A. R. Panat, 2016) is considered to be one of the early studies that are using the Higher-Order Crossings (HOC) and Hjorth features. They also incorporated the Principal Component Analysis (PCA) as well as Fourier Transform (FT), and Short-time Fourier Transform (STFT), then it has been proven that using HOC is better than the statistical features.

5. This paper (Byun, et al., 2017) is not introducing any technically novel way of classifying the emotions but it's highlighting the usage of a long trial, by employing a long music session, a large set of data has been collected from 32 subjects, and 1 minute of trial. as well as collecting new shape of data, this is considered as a stand-alone dataset that can be referred by other researchers in the field.

6. The main contribution for this work (K. Yano and T. Suyama, 2016) is the usage of a down streamed data. Signal has been collected from 10 subjects with 32 channel EEG apparatus that collect the signals by 512Hz and the humming window is 80 second of watching movie clips, but the data was lowered to 128Hz and then filtered to 0.5Hz and they removed 6 channels that are insignificant for identifying any results. The only takedown for this study that it used a 5-second sliding window to measure the accuracy of the results, this model might suffer from overfitting and this fact will go unnoticed, as this way of measuring the accuracy is not fully guaranteed.

7. This work (P. C. Petrantonakis and L. J. Hadjileontiadis, 2012) is considered solid investigation on time-frequency domain but also with a small relation the frequency domain, this data segmentation will isolate the data and change its shape to make it more recognizable by the classification algorithm, in other words, it segregates the data to help the discrimination. This work is collecting its own data.

8. This work (Y. Gao & H. J. Lee and R. M. Mehmood, 2015) is presenting a novel idea of employing the deep learning, or the multilayer network in favour of the emotion recognition, this work is different as it delegated the responsibility of the filters to the lower layers of the network, strictly speaking, the lower layers of the network will figure out the small relations of the data and help segregate the data. So they apply 3 layers of Boltzmann machines (RBMs) and back propagation to fine tune the network. The researchers in this study claim to achieve more accuracy than traditional ways of recognizing the emotion, the main benefit here is not that it achieves

the same or better result, but the fact that it's much easier to apply this deep network than the traditional models, as it requires less setup and it's autonomy make it a more convenient option, but it requires relatively more data than traditional models. Another important aspect of this work is that it's less prone to overfitting, and this feature is coming from the fact that it's more autonomous than traditional models, and the filters are not handcrafted in the model.

9. This work (Y. H. Liu, et al., 2014) is introducing a new feature extraction technique called kernel Fisher's emotion pattern (KFEP), which is considered as the main contribution of this work but the takedown is that it's measuring the accuracy on the IAPS observation which is harmful to the credibility of the method, but it's a good start to prove that KFEP is effective.

10. The main contribution of this work (D. Huang, et al., 2012) is that it harvests the knowledge from the neuroscience and apply them in developing a new ways of filtering the signals, by applying the insights and the breakthroughs of the neuroscience field on the BCI field, it drives the whole field forward, and we need more of this kind of studies. To be more specific, this study took the fact that the left hemisphere is more alerted in positive emotions and the right hemisphere is more alerted in negative emotions, and the study builds an algorithm on top of that fact called Asymmetric Spatial Pattern (ASP) which emphasizing the whole hemisphere and the electrodes on that hemisphere. Two other filtering methods have been applied for evaluation, one of the best two methods as well, filter bank common spatial pattern (FBCSP), which is an enhanced variant of (CSP) and asymmetric features.

11. The importance of this work (Y. P. Lin and T. P. Jung, 2014) that it highlights a different aspect of the emotion recognition which is exploring the day to day measurement of emotion, and measuring the emotion 4 times per day during the period of two weeks, in an average of 30 seconds per measurement, the intention is to monitor and help controlling the emotion over a long time range which is a deferent aspect compared to the other studies that is focus on

the technicality of detecting the emotions, instead of the usages and applications of the emotion recognition.

12. This work (Ou Lin, et al., 2015) is introducing a new philosophical discussion, regarding whether we are measuring the emotion or the mood, what's make it philosophical is the lack of data and evidence in this area, so we tend to use common sense while arguing about it, but this study provided an evidence of the fact that mood lives longer and it affects the measured emotions, so it's an important aspect when considering measuring the emotion. As well as introducing the concept of the human may have more than one emotion at a time, so we need to address that in the measurements as well, so it shows the evidence of the fact that your initial mood is affecting the preceding emotions.

13. In this work (D. Handayani, et al., 2015) it shares the same idea with this work (D. Huang, et al., 2012) of respecting the brain hemispheres while classification not only that but it's also corporate enhancements on the long short-term memory (LSTM) approach so the model is getting more confident as it observes more of a specific feeling.

14. The special about this work (B. H. Kim and S. Jo, 2018) is that it collects its own data and recognize 4 emotions from the data, happiness, fear, anger and confusion. The lack of data in this domain forces us to highlight a relatively small study like this as it's contributing to the field. And it's doing a decent job of detecting the emotions. There is another contribution as well which highlights the relationship between the emotion and the electrodes that contributed more in the result.

15. This work (S. G. Mangalagowri and P. C. P. Raj, 2016) is an investigation on the most effective filtration techniques, and it's using Support Vector Machine (SVM) for all the options. It discusses different options in filtering and feature selection.

16. The new thing about this work (I. Wichakam and P. Vateekul, 2014) is that it's using music as emotional stimuli.

17. This study (JOSEPH A. MIKELS, et al., 2005) is using deep learning and utilizing the convolutional neural network (CNN) in the network layers.

2.5. Summary

In this chapter, we discussed the evoked potential and the importance of this concept in the BCI field. We also mentioned the definition of the MI and the deference between the researches that tries to use the MI as the main drive or the main goal for the study and emotion recognition. We also discussed the SAM Self-Assessment manikin and the importance of this survey on the field, and finally we run through some of the well-chosen researches in the field and we extracted the range of emotions that this researches tried to cover, the electrodes that they were using, the feature extraction techniques, the machine learning models that they used, the experiment setup that they conduct, and finally the result that they landed on by using the previous path of execution.

CHAPTER THREE

3. ANALYSIS AND METHODOLOGY APPROACH

3.1. Introduction

In this chapter, we will scan the field of BCI and EEG from a high view angel and we will discuss the implications and the complications of trying to classify EEG signals in general and especially trying to detect emotions and recognize them. We will also discuss how to filter the signals before running the Machine Learning classifier on them and the most effective techniques and the least effective techniques as well, then we will discuss the most effective ML techniques and the least effective ML techniques, but without mentioning wich ones that we used this will be discussed in later chapters.

3.2. REVIEW OF DATABASES

There is a number of free databases that provide emotion stimulus and metadata around each item in the dataset:

3.2.1. International Affective Picture System (IAPS):

IAPS International Affective Picture System is a set of static images used to cause emotional arousal in the subject, every image will contain the dimensions valence, arousal, and dominance (JOSEPH A. MIKELS, et al., 2005).

The image set contains various pictures depicting mutilations, snakes, insects, attack scenes, accidents, contamination, illness, loss, pollution, puppies, babies, and landscape scenes, among others. This dataset will provide insight into the dimensional aspects of emotion. For instance, heart rate and facial electromyographic activity differentiate negative from positive valence, whereas skin conductance. Also, IAPS shed light on discrete emotions (disgust, sadness, fear, nurturance, and erotic happiness) have different valence and arousal ratings, and that they can be distinguished by facial electromyographic, heart rate, and electrodermal. *There is also International Affective Digital Sounds (IADS) which contain sounds.*

3.2.2. Geneva affective picture database (GAPED):

Geneva affective picture database (GAPED) the extensive use of these stimuli lowers the impact of the images by increasing the knowledge this why new images dataset is being introduced. An array of measurement has been used to give a comprehensive view of the dataset like facial expressions, as well as physiological reactions that are both centrally and peripherally driven. The limitations of this dataset are the lake of Positive emotions compared to the negative ones (Elise S. Dan-Glauser & Klaus R. Scherer, 2011)

3.2.3. Nencki Affective Picture System (NAPS):

Nencki Affective Picture System (NAPS) the motive for this dataset is the limitations for numbers of stimuli in specific categories or poor picture quality of the visual stimuli thus 1,356 realistic, high-quality photographs has been introduced. These images are divided into five categories (people, faces, animals, objects, and landscapes). The dataset rated according to the valence, arousal, and approach-avoidance dimensions using bipolar semantic slider scales using Self-Assessment Manikin (SAM). All the images in the dataset are high quality 1,600 by 1,200 (Artur Marchewka, et al., 2014).

3.2.4. Open Affective Standardized Image Set (OASIS):

Open Affective Standardized Image Set (OASIS) it’s an open-access image set not copyright restricted like IAPS with 900 images and four categories animals, objects, people, and scenes. Also, the dataset has extremely positive and extremely negative images (Benedek Kurdi, et al., 2017).

Image sets are not the only way to trigger emotions, this study (Ou Lin, et al., 2015) conducted a gambling task to trigger to types of emotions Regret and Rejoice to be able to identify regret and find the best features set that describe the regret emotion.

Other studies (Byun, et al., 2017) (K. Yano and T. Suyama, 2016) (D. Huang, et al., 2012) used music listening or movie clips to trigger the emotion and later on the map it to the collected data by using SAM.

3.3. Features extraction

In emotion detection using EEG, there are deferent features describing a range of emotion but it's not agreed which feature set are most describing which emotion specifically (R. Jenke & A. Peer and M. Buss, 2014).

Before starting with feature extraction techniques we can list common ways to preprocess the signal and remove the artefacts like eyes blinking.

Removing artefacts:

3.3.1. Principal Component Analysis (PCA):

Principal Component Analysis It is a powerful tool for dimension reduction of data without loss of information (J. Kaur and A. Kaur, 2015). The data is linearly transformed in such a way that only orthogonal components are retained.

3.3.2. Independent Component Analysis (ICA):

Independent Component Analysis PCA will reduce the data to components then ICA will separate the components. So they will be separated into EEG data and Artifacts. Later on, we can remove the artifacts. That's why we identified them. But the limitation is the number of factors affecting the signal is not identified also the assumption that EEG data and artefacts are especially fixed is not always right.

3.3.3. Fractional Dimension:

This method is helpful to reduce the complexity of a signal. It's a measure of the space-filling so it will try to fit a minimum number of circles in a given original value which will represent our EEG signal.

3.3.4. Other techniques

like CAR (M. Rajya Lakshmi, et al., 2014) Common Average Referencing this will remove the noise by subtracting the common activity from the position of interest (SL) Surface Laplacian or (CSP) Common Spatial Patterns

After cleaning the data from artefacts we can start with feature extraction:

3.3.5. Statistical

3.3.5.1. Autoregressive (AR):

The AR modelling represents the EEG signal and it's common. There are other methods like a weighted moving average filter to calculate the randomness of a signal.

3.3.5.2. ARMA and MVAM:

Autoregressive Moving Average and Multi-Variate AR can be used to analyze the signals in the time domain.

3.3.5.3. GARCH:

Generalized Autoregressive conditional heteroskedasticity, used for time-varying volatility, the volatility is the standard deviation.

3.3.5.4. Others

Like the Burg Method and Durbin Recursion and Yule-Walker.

3.3.6. Time domain

3.3.6.1. ERP:

Event-related Potential it's difficult to detect ERP linked to emotion

3.3.6.2. Hjorth features:

This method is providing three parameters (features) Activity and Mobility and Complexity, the Activity is just the squared standard deviation to get the signal power.

3.3.6.3. Non-Stationary Index (NSI):

Non-Stationary Index analyzing the variation of local average over time this will represent the measure of the complexity. The signal is divided into smaller parts and the average of each part is computed the NSI is the standard deviation for those averages

3.3.6.4. Fractal Dimension (FD):

Fractal Dimension it's a measure for complexity and there are many ways to compute it.

3.3.6.5. Higher Order Crossings (HOS):

Higher Order Crossings this is the most solid method (A. Patil & C. Deshmukh and A. R. Panat, 2016) this method has been used in preprocessing as a noise reduction technique.

3.3.7. Frequency domain

3.3.7.1. Band power:

The very common method the frequency bands can be delta (0-4 Hz), theta (4-8 Hz), alpha (8-16 Hz), beta (16-32 Hz), and gamma (32-64 Hz) used in this study (JOSEPH A. MIKELS, et al., 2005) and the humming window is usually 1 second. Commonly used with Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) or Short Time Fourier Transform (STFT) or Power Spectral Density (PSD). STFT is the most solid approach.

3.3.7.2. Higher Order Spectra (HOS):

Higher Order Spectra it's an extension of second-order measures. Second-order measures work fine if the signal has a Gaussian form (Normal distribution), Any Gaussian signal is completely characterized by its mean and variance. But the HOS of Gaussian signals are either zero or contain redundant information.

3.3.8. Time-frequency domain

3.3.8.1. Hilbert–Huang Transform (HHT):

Hilbert–Huang Transform it's a way to break down the signal to intrinsic mode functions (IMF) with the trend, IMF is a function representing the signal part. HHT work well with data that is nonstationary and nonlinear. And it's more like an algorithm.

3.3.8.2. Short Time Fourier Transform (STFT):

Short Time Fourier Transform: this method is introduced to provide the bridge between the Fourier Transform and the Wavelet Transform the FT does not provide time-frequency analysis so the signal is broken down into parts and the part signal is assumed to be stationary.

3.3.8.3. Wavelet Transform:

We can use DWT or CWT Discrete or Continuous. We can perform multiresolution analysis (MRA) also known as a multiscale approximation (MSA) to balance time resolution and frequency resolution

3.3.9. Multi-layered neural network (deep learning):

(Y. Gao & H. J. Lee and R. M. Mehmood, 2015) In this study they apply deep learning on the raw signal without hand- crafted feature extraction they really on the deep learning layer to provide the abstraction layers. Three layers of Restricted Boltzmann Machines (RBM) are introduced

3.4. Features Selection

Researchers perform features selection to identify which subset of the features actually matter to get the best classification result. There is no general agreement on which features are better to identify which emotion. But there are deferent methods than we can use to spot the best features. Feature selection considered part of the machine learning field:

3.4.1. Min-Redundancy-Max-Relevance (mRMR):

It’s an algorithm used to accurately identify the features that correlate to the result.

3.4.2. Relief:

The purpose of this algorithm is to draw instances at random, then compute the nearest neighbours, and then adjust the feature weighting vector to give more weight to features that discriminate the instance from neighbours of different classes

3.4.3. Bhattacharyya distance:

It measures the similarity of two discrete or continuous probability distributions. It is closely related to the Bhattacharyya coefficient which is a measure of the amount of overlap between two statistical samples (Byun, et al., 2017)

3.4.4. *Comparison studies:*

For example (Byun, et al., 2017) is trying to detect emotion from music listening and the most effective channels are 3 channels and the most effective wavebands are beta and alpha. By reducing the channels and the frequency bands they select the most relevant features to extract the emotions that they are targeting to classify.

3.5. Classification

3.5.1. Support Vector Machine (SVM):

Support Vector Machine most common classifier due to the high-quality results (Sunil Kalagi, et al., 2017)

3.5.2. Learning vector quantization (LVQ):

Learning vector quantization competitive network which uses supervised learning (Esmeralda C. Djamal & Poppi Lodaya, 2017).

3.5.3. k-nearest neighbours (k-NN):

k-nearest neighbours algorithm Very simple to understand and easy to be implemented but poor runtime performance (Sunil Kalagi, et al., 2017).

3.5.4. Artificial Neural Networks (ANN):

Artificial Neural Networks nonlinear classifier the most used version of ANN is Multi-Layer Perceptron Neural Network (MLPNN) or (MLP).

3.5.5. Restricted Boltzmann machines (RBMs):

Restricted Boltzmann machines (Y. Gao & H. J. Lee and R. M. Mehmood, 2015) uses three layers of RBM to identify 4 different distinct emotions.

3.5.6. Others:

LDA Linear Discriminant Analysis fails if the discriminatory function is not in mean but in the variance of the data (M. Rajya Lakshmi, et al., 2014), also NBC Naive Bayes classifier it’s not widespread in the BCI Brain-Computer Interface applications. Hidden Markov Model (HMM), Gaussian Mixture Models (GMM).

3.6. Summary

We discussed the first level of emotion recognition which is how we are going to trigger this emotion on the subject in an ethical manner, of course, we can put the subject in a very stressful situation to see the fight and flight responses but that requires a certain level of discussions in the ethical board, hence the well-known databases that we can use without this kind of issues. Also we discussed how we can extract the features from the signals and we categorized which techniques can be used online vs offline, we also run through different machine learning techniques and identify the efficiency of each one.

CHAPTER FOUR

4. METHODOLOGY

4.1. Introduction

In this chapter we will describe the methodology that has been used to classify different emotions and also the definition of the emotion and how we convert it into a number to be easily identified by the machine learning classifiers, we also picked the best techniques and combined them into one and we created a way to pick the best result from the best five techniques that we chose.

4.2. Methodology

EEG signals are often analyzed on short-time segments called Trails (A. Barachant, 2013) in other words if we represent the EEG data in a table format where the columns are the channels and the rows are the actual values (signals) the trail will be the group of rows when the subject was looking at a particular image.

4.2.1. Describing what emotion means

In our daily life we describe emotion as distinctive words, the word happy is well known because we all have an agreement on how happiness looks like, but if you are very happy people tend to say I'm glad which is the same as happiness just a little stronger feeling of happiness, we can't use this foggy definitions in our study we have to change it to some kind of numbers to be more convenient to measure the strength of a particular emotion. As well as to cover a wide range of emotions.

In this study (Scherer & Klaus R. (2005)., 2005) Emotion is *defined as an episode of interrelated, synchronized changes in the states of all or most of the five organismic subsystems in response to the evaluation of an external or internal stimulus event as relevant to major concerns of the organism.*

The *components* of an emotion episode are the respective states of the five subsystems and the *process* consists of the coordinated changes over time.

Table 2 shows the relationship between components and subsystems as well as presumed substrate and functions. Three of the components have long-standing status as modalities of emotion or shapes of emotion expression, body expressions and arousal, and subjective experience. The clarification of action tendencies and the preparation of action have also been implicitly associated with emotional arousal (e.g. fight-flight responses) but it is only after the clarification of these motivational consequences in a sequence of theories some studies claim for the emotion-differentiating function of action tendencies that these important features of emotion episodes have acquired the status of a major component in their own right. Many theorists still prefer to see emotion and cognition as two independents but interacting systems. However, one can argue that all subsystems underlying emotion components function independently much of the time and that the special nature of emotion as a hypothetical construct consists of the coordination and synchronization of all of these systems during an emotion episode, driven by appraisal.

Table 2: Relationships between organismic subsystems and the functions and components of emotion

Emotion function	**Organismic subsystem and major substrate**	**Emotion component**
Evaluation of objects and events	Information processing (CNS)	Cognitive component (appraisal)
System regulation	Support (CNS, NES, ANS)	Neurophysiological component (bodily symptoms)
Preparation and direction of action	Executive (CNS)	Motivational component (action tendencies)

Communication of reaction and behavioural intention	Action (SNS)	Motor expression component (facial and vocal expression)
Monitoring of internal state and organism-environment interaction	Monitor (CNS)	Subjective feeling component (emotional experience)

CNS = central nervous system; NES = neuro-endocrine system; ANS = autonomic nervous system; SNS = somatic nervous system.

How can emotions, as defined above, be distinguished from other affective phenomena such as *feelings*, *moods*, or *attitudes*?

Let us take the term *feeling* first. As shown in Table 2, the *component process model* only uses this term for the *subjective emotional experience component* of emotion, presumed to have an important monitoring and regulation function. In fact, it is suggested that ‘‘feelings integrate the central representation of appraisal-driven response organization in emotion’’, thus reflecting the total pattern of cognitive appraisal as well as motivational and somatic response patterning that underlies the subjective experience of an emotional episode.

Using the term feeling, a single component denoting the subjective experience process, as a synonym for emotion, the total multi-modal component process, produces serious confusions and preventing us from understanding the phenomenon. In fact, it can be argued that the long-standing debate generated by William James’s peripheral theory of emotion is essentially due to James’s failure to make this important distinction: when in 1884 he asked ‘‘What is an emotion?’’, he really meant ‘‘What is a feeling?’’.

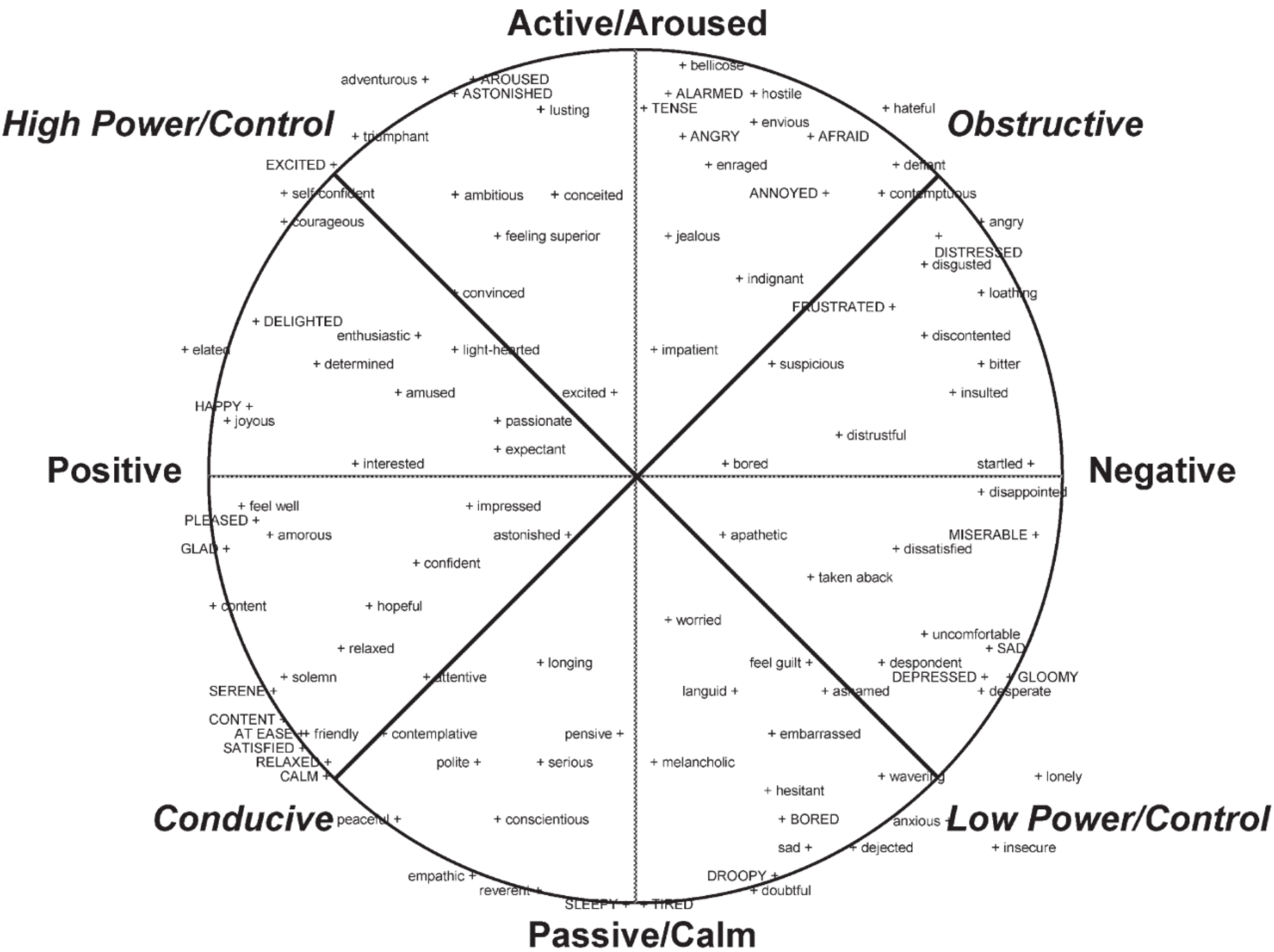


Figure 2: Dimensional structure of the semantic space for emotion (Lesley Ann Axelrod, 2009)

SAM model (Margaret M.Bradley & Peter J.Lang, 2002) is ranging from smiling to a frowning figure, in other words, the happy and unhappy figures in this order, when representing the pleasure dimension, and ranges from an excited to a relaxed figure, wide-eyed to sleepy figure for the arousal dimension. The dominance dimension represents changes in control with changes in the size of the figure inside the box: a large figure indicates maximum control in the situation small figure indicates a minimum control in the situation. In this version of SAM, the subject can place an ‘x’ over any of the five figures in each scale, or between any two figures, which results in a 9- point rating scale for each dimension. In this study we are using an enhanced version of SAM to suite the study dynamics, we also make it computerized rather than the paper-based SAM model that has been proposed in this study (Margaret M.Bradley & Peter J.Lang, 2002).

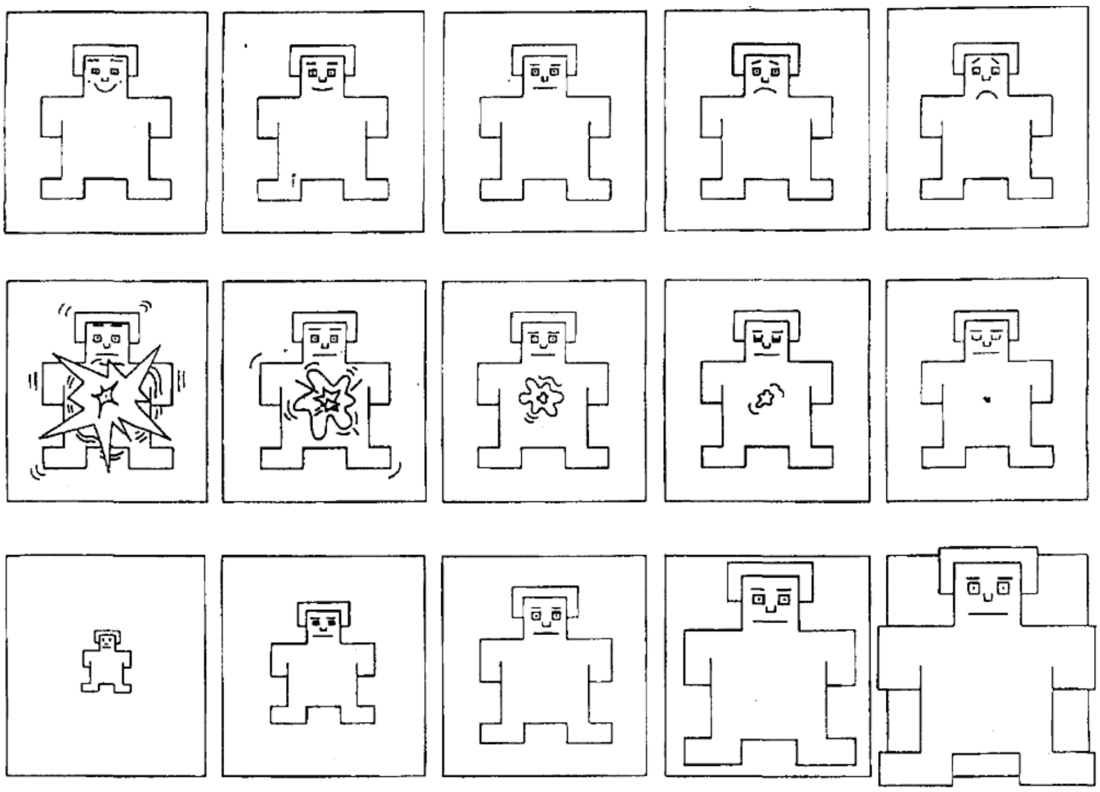


Figure 3: The Self-Assessment Manikin (SAM) used to rate the dimensions of valence (top panel), arousal (middle panel), and dominance (bottom panel) (Anon., 2002)

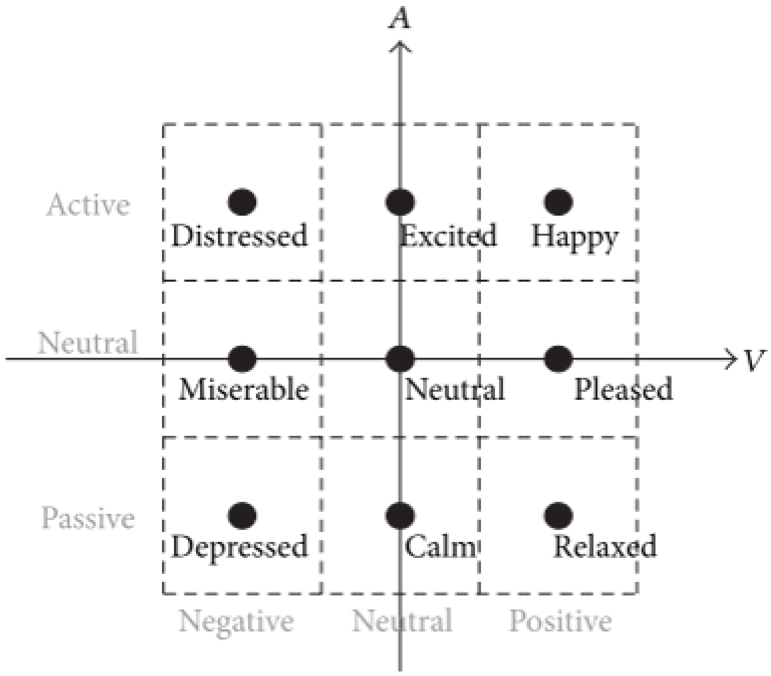


Figure 4: Emotion State Classes (S. Jirayucharoensak, et al., 2014)

4.2.2. Apparatus

EPOC Plus with 14 channels: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. With LSB = 0.51μV and 128 Hz with a resolution of 16 bits. The price at the time of this study was 750 USD plus 100 USD for the license to use the SDK, the SDK is giving a raw EEG Data.

4.2.3. Setup

191 images have been selected from GAPED (Elise S. Dan-Glauser & Klaus R. Scherer, 2011) to trigger the stimuli in the tested subjects, the images have been selected from all the rages of emotions provided by GAPED to get the maximum possible range of emotion. Contents included pictures of a snake, spider, gun, mutilated face, rolling pin, soldier, flowers, mountains, cake, baby, and others.

It’s very important to note out that the goal here is not to detect images, in other words, the images are only for stimuli purposes and we get rid of it after that, the goal here is to detect emotion which means that we need to link every brain pattern with the emotion that it represents. If you are happy the signals should look in a certain way and if you are angry the signals should look in another way and we can classify and distinguish between these signals.

Experimental Procedure

There were two volunteers in this study we asked them to wear the headset and sit comfortably, we also asked them not to move at all during the recording so the data don’t get mixed up, anybody movement will result in a bad data and we will require to start from the beginning, there was 4 session each one lasted around 2 min. there were 2 steps first we collect the EEG data based on the photos that are automatically moving and the second step is to ask them how did they felt during that image which is the self-assessment manikin SAM.

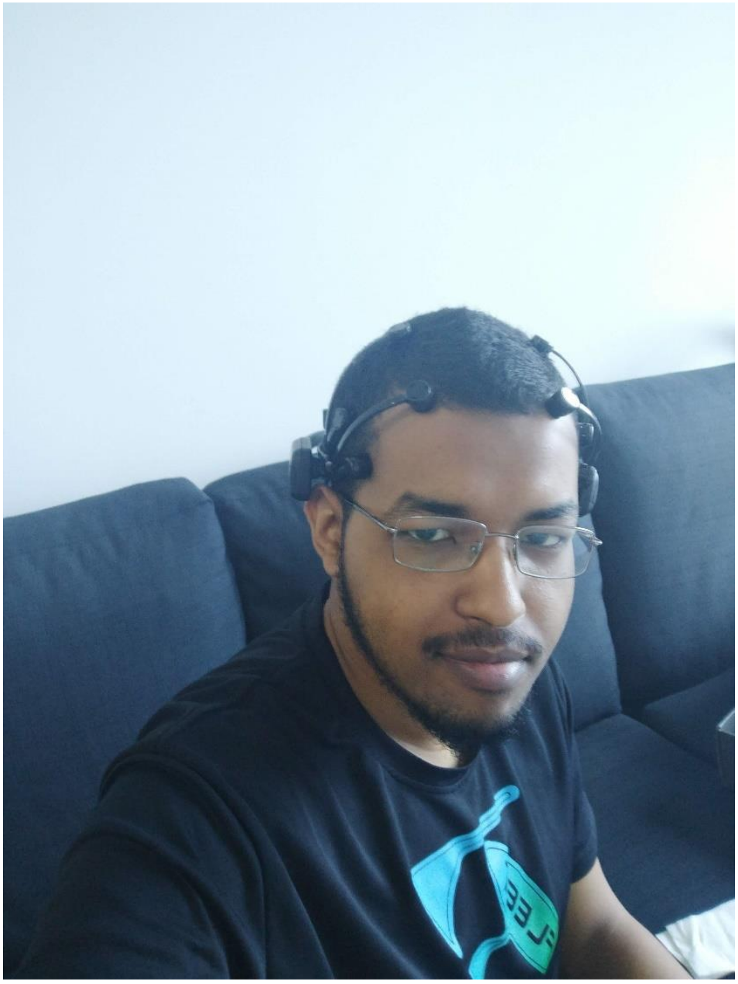


Figure 5: One of the volunteers wearing the headset

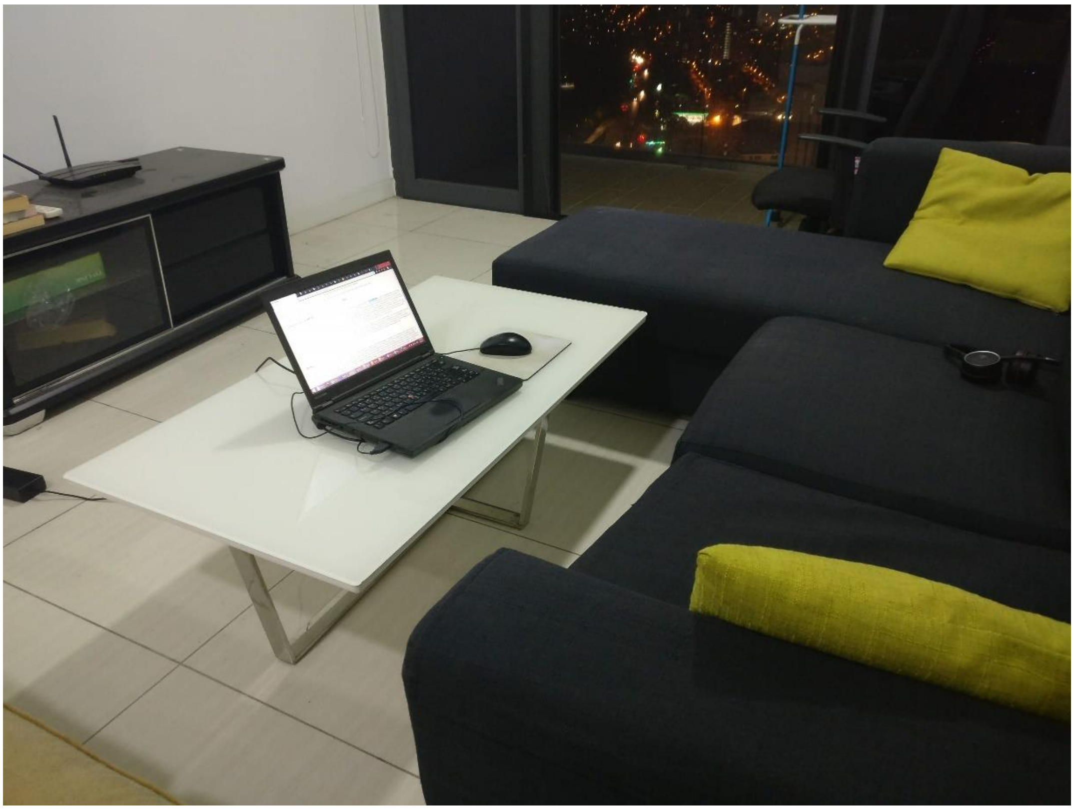


Figure 6: The experiment setup

Tow healthy subjects participated in this study. The data will be collected during 4 sessions.

Session 1:

a five second of countdown then the participants will look at computer screen and watch images that changes every 2.5 seconds, as per this study (H. Xu and K. N. (Kostas) Plataniotis, 2012) the P300 ERP (Event Related Potential) will happen after 300 milliseconds so 2.5 second is a good number that will cover the whole emotion at the same time will not allow the subject to be distracted. Multiple images will be shown to the subjects per session. The data from this session will be tagged by the image and the trails will be identified.

Session 2:

The same as session 1 but this data will be used to test the model. Keeping in mind that this data will be distorted because the subject will be familiar with the images and the emotions will be less this time due to the emotion memory. Also, the order of the images will be shuffled.

Session 3:

now the same images will be presented and the subject has to select from 3 buttons describing the valance as (happy, natural, sad) and the arousal as 3 level and the dominance as 3 levels as well (M. M. Bradley and P. J. Lang, 1994). The maximum number of emotions from this setup will be 20 distinct emotions.

Session 4:

The same as session 3; the data from this session will be compared with session 3 to eliminated the conflicts e.g. if the subject select happy in session 3 and sad in session 4 this emotion will be discarded.

Figure 7: the emotion collecting tool buttons are valance, arousal and dominance

The first subject watched 131 images then tag it with the appropriate emotions, he has 5 distinct emotions from the range of possibility which is 20 possibilities. The second subject watched 60 images then tag it as well he has 10 distinct emotions tagged.

4.2.4. Model Generation

Emotion recognition is a classification problem and due to the SAM model of detecting the emotion now, we can group them by the three main dimensions Valance, Arousal, Dominance. To detect this emotion, we have multiple models we can use. Rather than using only one model we are using the best model per subject, which means we will train all the models on the subject then we pick the best one. The first step is the channel selection.

4.2.4.1. Riemannian Geometry

This approach has been applied in the past on radar signal processing and image processing. If there is an interest for using the covariance matrix as a feature in a classifier the natural choice will be to vectorize this covariance matrix in order to process the quantity as a vector and then we can use any vector-based classification algorithm so this approach will be the same as CSP like special filtering first then we can use any linear classification for example. The direct vectorization of special covariance matrix will not take into account the correlation between the vectors the coefficients of the symmetric and the Symantec positive definite SPD, along with the vectorized covariance metrics do not follow the normal distribution so a large number of classification algorithm are less effective like e.g. linear discernment analysis LDA which is optimum for Gaussian distribution. The Riemannian method is talking into account the Riemannian Geometry at each point a scalar product can be defined in the associated tangent space (A. Barachant, 2013).

.

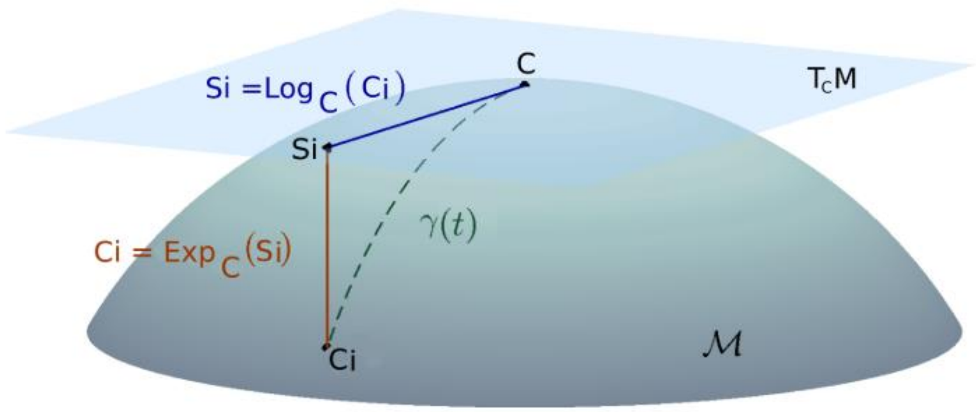


Figure 8: Manifold M and the corresponding local tangent space TCM at point C (A. Barachant, 2013).

4.2.4.2. Electrode Selection

There is a tow type of channels selection the First subject based which choose the channels that guarantee the best results per subject. The second type is Application-Based which select the best channel for a particular action or stimuli.

Backward selection and forward selection is including or excluding one channel at a time for best fit but the computational power of this solution is exponential and it’s time-consuming.

In this model, we choose to use the method proposed by (A. Barachant, et al., 2011) as it’s using the Riemannian distance between two covariance matrices to choose the best channels.

The range of model can be described as follow:

4.2.4.3. xDAWN

The proposed filtering in (B. Rivet, et al., 2009) (B. Rivet, et al., 2011) is used to provide the best SNR Signal to noise ratio, which takin into account the signal and the ration as opposed to PCA (Principal component analysis) which only taking into account the signal. Then the Logistic Regression will run on the filtered data from the previous step.

4.2.4.4. Hankel

The idea is to build covariance matrices by concatenating time delayed trial along the channel axis (Hankel matrices). It allows us to take into account the temporal as well as spatial information. It’s beneficial to reduce the size of the states. Again, we use the Logistic Regression model as a posterior step (S. Lemm, et al., 2005).

4.2.4.5. CSSP

Stands for Common Spatio-Special Pattern the idea is to feed the well-known CSP to the Hankel covariance then to the logistic regression (Z. J. Koles, et al., 1990).

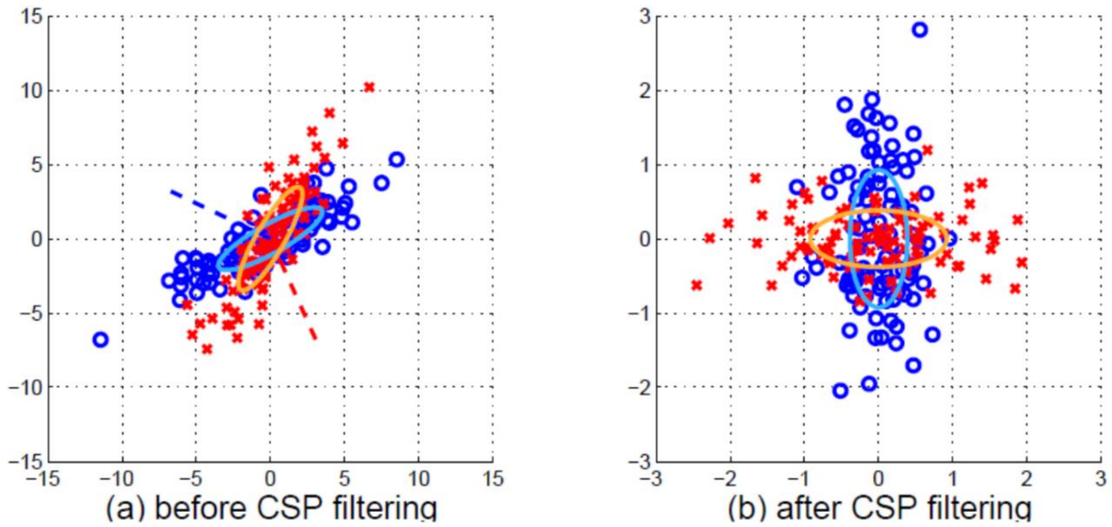


Figure 9: Describing the feature before and after applying the CSP filter

This setup is considered controlled environment due to multiple factors, first, we didn’t have the urge to clear the artefact e.g. eye blinking (movement) because we asked the subjects to try to reduce the eye blinking as much as possible but in real life scenarios this artefact has to be removed also

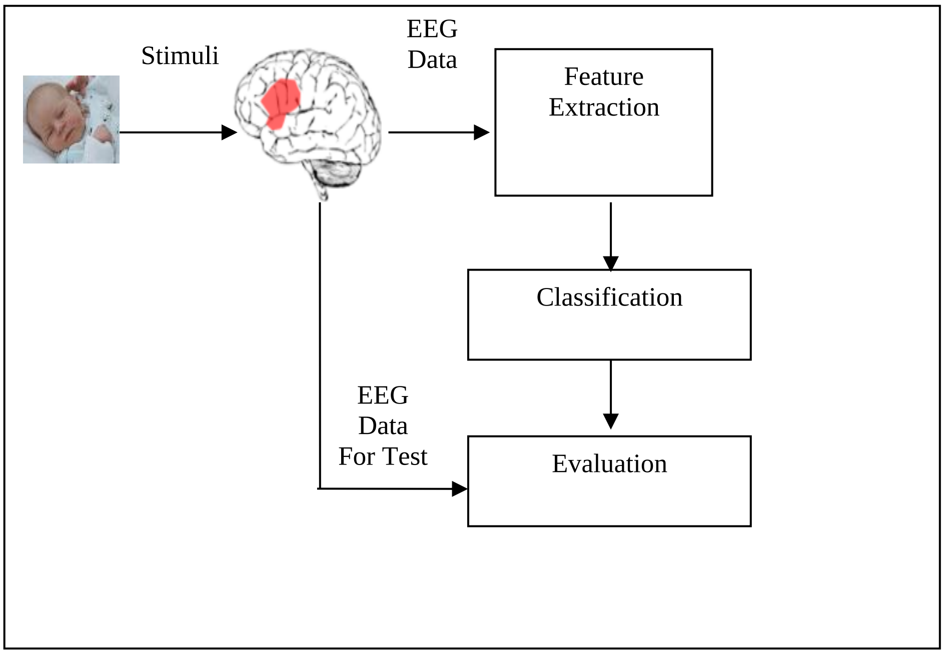


Figure 10: Describe the steps of EEG data collection and model generation

The trails are well known due to the experimental setup where we calculate when to show the images and then tag it with appropriate emotion this will create very clear lanes for the trails and will make the whole process easier. Another thing is image set GAPED is well known and the images tagged with arousal and valance from the previous studies, so it's easy to collect the appropriate images that guaranteed to trigger the targeted emotions also the fact that it's open-sourced and free to use makes it simpler and more convenient. Also, the range of emotion that we choose can produce 20 different emotions maximum but the standard SAM survey contains 5 items in each dimension which will result in 35 different emotions.

Along with the fact that all the sessions have been conducted in the same time frame which the mood of the subject is consistent and the subject is also isolated from any distractions like an external sudden sound or nearby movement. All this factor will not be present in a real-life scenario and will affect the quality of the signals and in turn the classification accuracy.

4.3. Summary

In this chapter, we discussed the definition of emotions and the wheel of emotions described in a psychological studies. We also discussed how we are collecting the true emotions by using SAM, and also measure the strength of emotions, as well as measuring the accuracy of the survey before submitting it to the ML classifier. We also described the experiment setup and the way that we conduct the EEG data reading. And finally describing the ML classifiers and the filtering that will occur on the data before pushing it to the classifiers.

CHAPTER FIVE

5. TESTING, VERIFICATION AND EVALUATION

5.1. Introduction

In the previous chapter, we defined our novel methodology of classifying the emotions, in this chapter we will discuss the accuracy of that model. To do that we will be running the model on data that has been collected by different parties. As well as comparing bit and pieces with other people work. As well as we are going to test the model accuracy within itself with running the proper measurements, in addition to evaluating the model on other similar data to compare the accuracy of the proposed model on different type and shape of data.

5.2. Verification

5.2.1. Testing

There is a difference between predicting the image and predicting the emotion, in other words, subject a could see an image and the evoked emotional response will be happy but the same image could be seen by subject b and the evoked emotion is sad hence the SAM method, also the images are infinite and building our model on a particular set of images we will need retraining step to change those images.

After collecting the EEG data and tagged it with the user responses (emotions), then we fitted all the proposed models by the data collected from session 1, then we choose the best model per person depending on an evaluation conducted from the data collected in the second session where the subject being introduced to the same images but in a deferent order to help with the emotional memory and to increase the accuracy of our evaluation, the premise here is if we introduced the same image to the subject the same emotional response will be triggered.

The accuracy is calculated based on the classes that have been predicted successfully, the classes have been produced by feeding the data from session two to the best model selected per subject, the first subject is 42.8% and the second subject is 72% this is because the second subject has half less distinct emotions. Another factor is the signal accuracy could be deferent between the subjects resulting in a difference in the accuracy.

5.2.2. Evaluation from a Machine Learning Perspective

Evaluation metrics are tightly coupled with machine learning concepts, Different machine learning tasks have different performance metrics. For example, if we build a classifier to detect spam emails, we would like to use average accuracy as a measurement for that classifier, or maybe we can use log loss, or also we can use AUC (Area Under the curve).

On the other hand, if we are trying to predict a number like a stock market value for a certain company, then we can use RMSE (root mean square error) as an evaluation measurement. If we are sorting items based on the relevance of a query submitted to the website, then we should consider something to evaluate the ranking such as precision-recall. Or we can use something like NDCG (normalized discounted cumulative gain)

Precision-recall is also famous for being used to measure the classification models.

These are examples of models and evaluation measurements for them.

5.2.2.1. Hyperparameter Search

Model evaluation is used when we are trying to make sure the model is performing correctly and we want to measure how correct this model is performing for the sake of comparing it to a different model and decide who is the best or for the sake of defining if this model is performing as per the safety error margin for the specific domain, to determine if this model is usable for this application or the accuracy is low enough resulting in making the model is unusable for this particular application.

But what if we want to enhance the model overall accuracy. Then we need to tweak the model parameter manually and train it again and watch the change in the result, of course, this has to be done by giving some thoughts before changing some parameter and using common sense of the domain (the data) and the understanding of how the model algorithm behaves.

This can be a very mandarin task and become complicated very quickly, and it’s kind of a tedious task that can be automated, sure indeed there is something called grid search for example to do the exact same thing, or we can use random search as well, or even smart search to find to optimum hyperparameter for the model.

5.2.3. Classification Evaluation Metrics

Classification is all about predicting the class label for a given input data.

Binary Classification: There are two possible outcomes

Multi-class classification: There are multiple classes, basically more than two classes.

We will give an example and we will discuss the evaluation measures for that example.

A Spam detection model where the features are the email data, basically the sender, the receiver, the subject and the email body and all the words inside it. In this example, we want to detect whether this email is spam or not.

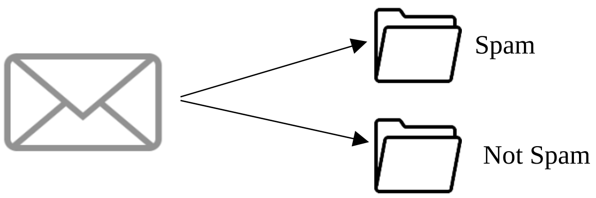


Figure 11: Email spam detection

5.2.3.1. Accuracy Evaluation Measure

Accuracy is the simplest form of measurement, the accuracy is measuring how often the model will guess the correct class, so it’s a ratio between the correct guesses and the total number of guesses.

5.2.3.2. Confusion matrices Evaluation Measure

Accuracy is easy and simple but the problem with accuracy is that there is no distinction between the classes, correct answers for class “spam” and class “not spam” are treated equally, and this is not enough.

If we look at how many examples failed for class “spam” versus class “not spam”, we might find new information, which is the accuracy might differ between classes.

For example, if the model detected the newly received email as spam but the fact was the email is not a spam, in this case, it’s called false positive, and the consequences of tagging the email as spam are bad compared to the other way around.

A Confusion Matrices show us a more detailed breakdown of correct and incorrect classifications for each class. The columns in the table are representing the prediction, and the rows of the table correspond to ground truth labels.

Table 3: Confusion matrices evaluation model predictions

	Predicted as positive	Predicted as negative
Labelled as positive	80	20
Labelled as negative	5	195

The overall accuracy of this model is calculated as follow:

(80 (predicted positive) + 195 (predicted negative)) / (100 (positive sum) + 200 (negative num))

= 91.7%

But we will lose the details of the accuracy of each class if we relied on this number only.

From the provided example the positive class accuracy is 80% calculated as follow:

80 (predicted positive) / (20 (predicted negative) + 80 (predicted negative)) = 80%

The negative accuracy is 195/ (5 + 195) = 97.5%

5.2.3.3. Per-Class Accuracy Evaluation measure

If we calculated the average accuracy per class that’s what it’s called A variation of accuracy. We calculate the average accuracy for each class. A micro average is an example of accuracy, and average per-class accuracy is a macro-average.

In the above example, if we calculated the average per-class accuracy it will be

(80% + 97.5%) /2 = 88.75%. Notice, in this case, the average per-class accuracy is totally different from the accuracy.

As we can see from the definition of the average per-class accuracy and the overall accuracy, when there are different numbers of examples (runs or predictions) per class, the average per-class accuracy will defiantly be different from the overall accuracy. This is very important because when the classes are imbalanced, i.e., there are a lot more runs of one class over the other, then the usage of accuracy will give us a very clear picture because the class that has more runs will dominate the result.

In that case, we should consider looking into the per-class accuracy, as it gives us a more detailed picture.

Per-class accuracy has its own down faults as well. For example, if there are very few runs of one class, then test result for that class will have a large variance, and that means that its accuracy per class of that specific class is not as reliable as other classes. Taking the average of all the classes to hide the confidence measurement of the individual classes

5.2.3.4. Log-Loss

Log-loss is also called logarithmic loss, the power of this measure is that it gets into the smallest details of the classifier model results. Specifically, log loss can be used when the model that we are developing is giving numeric output or a numeric prediction rather than classes. If we have only two classes and our model outputting 0.51 for the class “not spam” then we will understand that it’s may or may not be a spam and the classifier can’t really predict the right class because the prediction which is 0.51 is near the boundaries of the decision which is 0.5, in other words, we are dealing with the output prediction as a gauge.

The problem with accuracy is it’s all or nothing, no partial credit, but with log loss, we can have partial credit.

Let's say the model predicted that the newly arrived email was a spam but the model was sure by 90% probability, which means I was partially correct it wasn't 100% correct, in fact, it was 90% correct, if we really want to measure the accuracy and penalize the model for not being so sure we use log loss.

Log-loss will consider this idea of a probabilistic guess and will create a measurement of accuracy.

The log-loss equation looks like this:

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + \left(1 - y_i\right)\log\left(1 - p_i\right)$$

Equation 1: Log-loss function

Equation 1 doesn't look like an obvious thing for the first look at it, but when we mention the details it will look clear.

N is the data point count, P of I (Pi) represent the probability of the I (the) data point, and y of I (Yi) is the true class. So, we have P as the prediction and Y as the true value that we are measuring the accuracy against. Giving that we have two classes only that's mean the equation is looking into the summand of "spam" or "not spam".

Will give us a number we want it to be as minimum as possible, it will be used to penalize the model for overconfidence in a wrong answer and under-confidence in a correct one (Zheng, 2015).

5.2.3.5. Area Under The Curve (AUC)

the area under the curve or as an abbreviation (AUC). Or also can be called Receiver Operating Characteristic Curve or for short ROC curve. This an exotic name has been used since the 1950s from the analysis of the radio signal, The ROC curve plotting the rate of true positives to the rate of false positives, by doing this shows the sensitivity of the classifier shows. Or we can say, it shows you how many correct positive classifications can be gained as you allow for more and more false positives. If the classifier was perfect and making no mistakes then it will hit a true positive rate of 100% immediately, without obtaining any false positives, this almost never happens in real life.

The AUC is one way to summarize the ROC curve into a single number so that it can be compared easily and automatically, by calculating the area under the ROC curve we are changing the graph the is requiring a visual inspection to a quantifiable number that is easy to compare and measure. But at the same time the ROC is not a simple number is a very detailed graph, an exact two numbers can be representing totally different graphs. But for the sake of simplifying the graph and use this information for making more automatic task implementations like the hyperparameters tuning, for example, we need a simple number (M. Congedo, et al., 2013).

If we want to compare two graphs then the graph that has more space under it is the better graph, because the true positive rate rocketing to a 100% very quickly, in the other hand the bad ROC will cover small or a little area. That's mean as higher the AUC as better the model

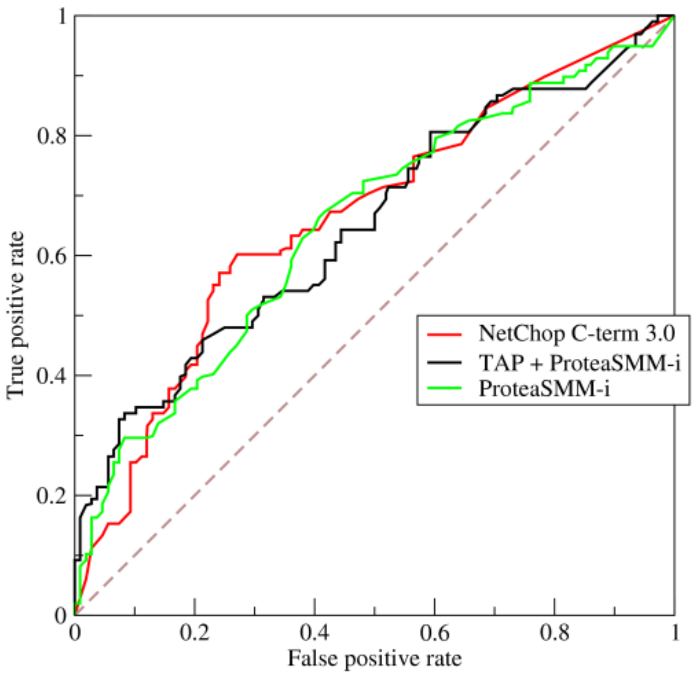


Figure 12: Sample of a ROC curve (example)

5.2.3.6. Ranking Metrics

By first look, we might think that these metrics are related to the continues data, but in fact, this measurement can be used for classification tasks, but more specifically for binary classification, which means that we have only two classes to predict from, just like our example "spam" and "not spam".

But this example doesn't really show the value of this metric, So let's say we are building a website and there is a query box when the user searches for something the result set will show items related to the query, here we can think of the two classes as a "relevant for the query" or "not relevant for the query" then we can order the result based on the probability of the prediction, also the model or the classifier can assign a value to the item in the result set, the more relevant the item is to the query the higher the value or the score is, instead of just one static label.

Another good example is the recommendation systems like the market basket analysis the recommendation system will act in one of two ways, either as a ranker or a score predictor. In the ranker case, the output is a list of items that are ranked, this list will be for each user. In the case of the score predictor, the recommendation system will return a predicted score for each user-item pair.

5.2.3.7. Precision-Recall

Precision is a metric and recall is another metric altogether, but we put them together here because they are related, and most of the time they are being used together.

Using the previous example of the market basket analysis or the recommendation system, if we are trying to answer the question how many of the items are relevant, or in other words, we got a list of items that are predicted from the recommendation system, so how many of those items are truly relevant? This question is revealing a metric called Precision. As the name implies the precision of the returned result.

On the other hand, recall is looking into the ration from all the items including the ones predicted, in other words from all the truly relevant items how many of them are predicted by the classifier (Zheng, 2015).

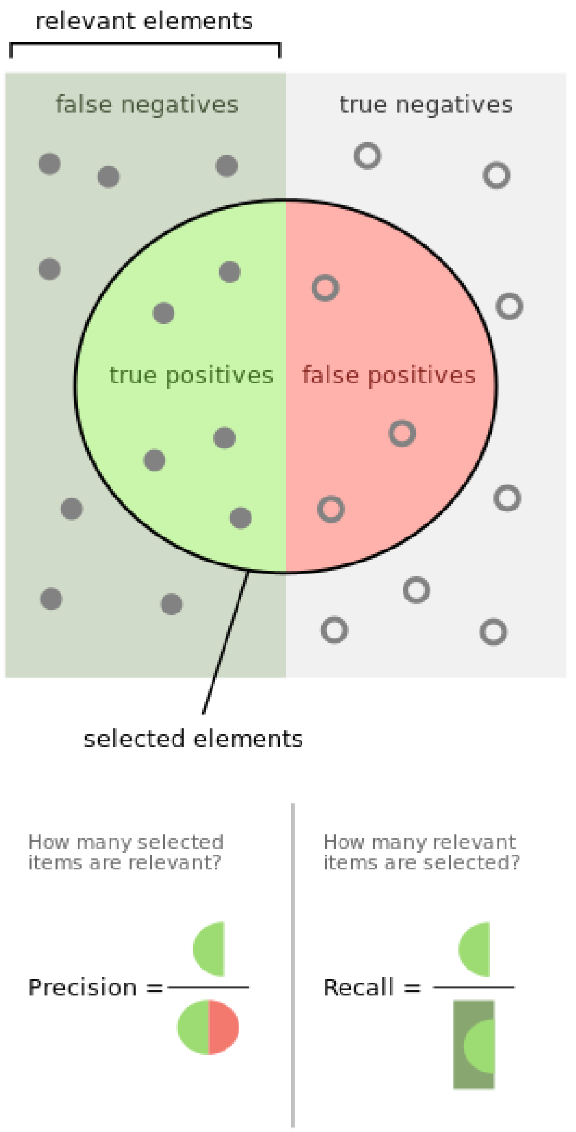


Figure 13: Precision and recall (Anon., n.d.)

$$\text{precision} = \frac{\#\ \text{happy correct answers}}{\#\ \text{total items returned by ranker}}$$

$$\text{recall} = \frac{\#\ \text{happy correct answers}}{\#\ \text{total relevant items}}$$

Equation 2: precision and recall

So, we can say, the precision is the ratio tp / (tp + FP) where tp is the number of true positives and FP the number of false positives. And the recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives.

5.2.3.8. Normalized Discounted Cumulative Gain (NDCG)

Precision and recall will deal with all the returned items equally, if the relevant item was in the first position or the relevant item was in the last position it will be treated the same, but this is not true usually people have items that are more relevant than others, when we look the result we should see the most relevant items on the top of the page and the less important or less relevant items in the bottom of the page, NDCG is trying to be one step closer to this behaviour, NDCG stands for (Normalized Discounted Cumulative Gain).

There are three related metrics CG (Cumulative Gain) and DCG (Discounted Cumulative Gain) and NDCG.

Cumulative gain is summing up the relevance of the top k items. Discounted cumulative gain (DCG) subtract the items count that are further down the list. Normalized discounted cumulative gain, NDCG as the name implies, is a normalized version of discounted cumulative gain DCG. It divides the DCG by the perfect DCG score so that the normalized score always between 0.0 and 1.0. in other words, DCG measures the usefulness, or gain, of a document based on its position in the result list.

5.2.3.9. F1 Score

F1 score is defined as a harmonic means between precision and recall. F1 is good when we have imbalanced data. To illustrate this let’s have the example.

Table 4: Example of a model run

		Prediction →			
		A	B	C	D
Actual ↓	A	10 0	8 0	1 0	1 0
	B	0	9	0	1
	C	0	1	8	1
	D	0	1	0	9

As we can see class A has more runs, also if we calculated the accuracy per class it might be a huge difference between classes this is what we mean by imbalanced data (Koelstra, et al., 2012).

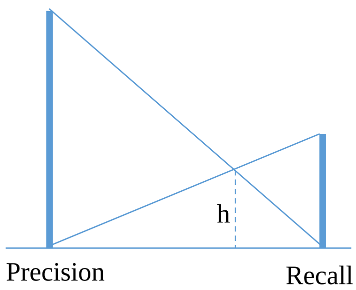


Figure 14: Harmonic mean (Anon., n.d.)

h: is half of the harmonic means

$$F_1 = 2\,\frac{\text{precision}*\text{recall}}{\text{precision} + \text{recall}}$$

Equation 3: F1 Score

The F1 score will be small if either precision or recall is small, the harmonic mean tends toward the smaller of the two elements.

5.2.4. Regression Metrics

5.2.4.1. Root Mean Square Error (RMSE)

RMSE (root-mean-square error), is the most used metric for regression tasks is also known as RMSD (root-mean-square deviation). We calculate the distance between the actual score and the predicted score then we take the square root of that.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{T}}.$$

Equation 4: Root Mean Square Error

Here y hat is the true score of the t(the) data point, and y of t is representing the predicted value, T is the count of the data points.

The RMSD can be described as the square root of the differences between true scores and predicted scores or the quadratic mean of these differences. In other words, we can consider it as the Euclidean distance between the vector of the observed values and the vector of the predicted values, averaged by the square root of t where t is the number of data points.

5.2.4.2. Quantiles of Errors

RMSE is the most common metric, but it's not perfect, there are some problems with it. Firstly, and most obviously, it’s an average, which means an outlier in the data may swing the accuracy of the measure, and result in it inaccurately. If the model performs very badly on one data point, it might be unnoticed or shifting the results. In other words, it's not robust against the outliers.

let’s take a look at the median the 50th percentile, which means the element of a set that is larger than half of the set, and smaller than the other half. If the largest element of a set changes from 1 to 100, the mean should change, but the median will not be affected at all. As we know from the real data scenarios there is always an outlier, in almost every aspect, which results in a lake of performance of the model in the real-life data, so it’s very crucial to figure out a model that will not be affected by the outliers, in other words, robust against the outliers. Hence the median absolute percentage MAPE.

$$\text{MAPE} = \text{median}\left(\left|(y_i - \hat{y}_i)/y_i\right|\right)$$

Equation 5: mean absolute percentage error

It gives us a relative measure of the typical error. On the other hand, we could compute the 90th percentile of the absolute per cent error, which would give an indication of an “almost worst-case” behaviour.

Perhaps the easiest metric to conduct is the percentage of estimates that differ from the true value by no more than X%. The choice of X depends on the nature of the problem. For example, the percentage of estimates within 10% of the true values would be computed by percentage of |(yi – ŷi)/yi | < 0.1. This gives us a notion of the precision of the regression estimator.

5.2.5. Datasets for Evaluation

For evaluating the model, we need to understand and evaluate multiple datasets, so let’s discuss some of those datasets especially those ones who are available publicly and free to use. Most of those datasets will not contain the same data our model meant to classify, but we can seek the dataset that has a similar shape or being collected by the same approach as our approach in this study so we can get a relatively similar result.

5.2.5.1. Clinical Datasets

The Temple University Hospital EEG Corpus: In this dataset is being collected from 12,000 patients using an apparatus that has a 16 channel, the EEG data collected size is 2 Terabytes.

EEG signals were recorded using several generations of Natus Medical. Incorporated’s NicoletTM EEG recording technology.

Upon completion will comprise over 20,000 clinical EEG records made at TUH, the apparatus function at 250 Hz using a 16-bit A/D converter and between 24 and 36 channels, the data is stored at EDF+ format which is one of the standards formats for storing the EEG data.

The report will also contain a briefing of the patient's clinical history and also the medications that were given to him. It also contains two extra fields, one is the Impression and the second is Clinical Correlation, which contains the physician's findings.

The patient information is ranging from gender, Date of birth, EEG signals reading duration, number of channels used, the pre-filtering that is used on the signals, and the apparatus frequency.

Approximately 75% of the sessions are standard EEGs less than one hour in duration, while the remaining 25% is from long-term monitoring sessions.

Also, in the data there are the events, this event has been added to the EEG Signals as markers and those markers have types that are representing the time when the patient triggered them, for example, Eye Open, Eyes Closed, Movement, Swallow, Awake, Drowsy / Sleeping, Hyperventilation, Talking. The event location in the timeline is approximate but still informative (A. Harati, et al., 2013).

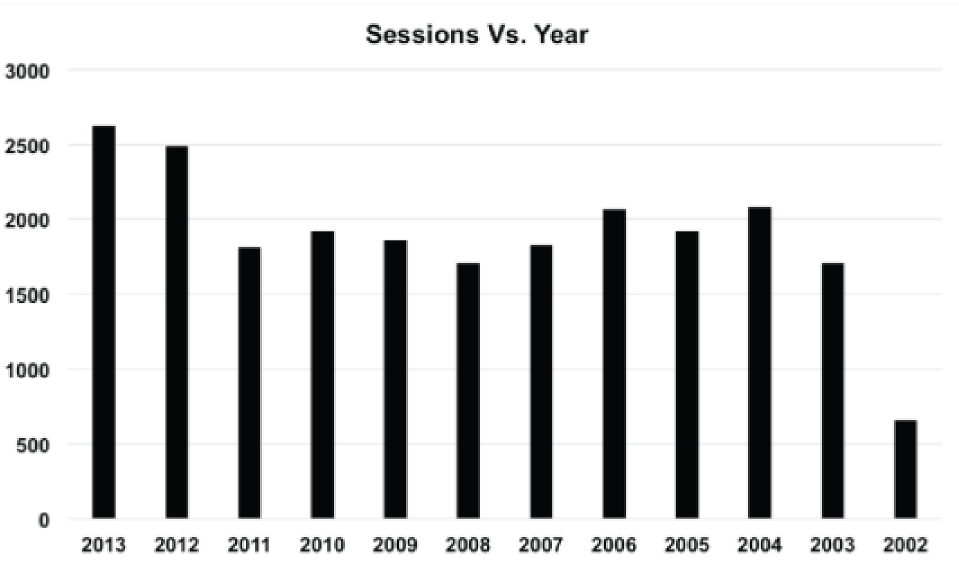


Figure 15: The number of EEG sessions per year (A. Harati, et al., 2013).

5.2.5.2. Motor Imagery

a. BCI2000

This data set has been obtained from 109 volunteers, it consists of over 1500 one minute and two-minute EEG recordings.

The data has been collected by making the volunteers perform different motor or imagery tasks, while they are wearing the headset. The headset has 64-channel EEG, and it was recorded using the BCI2000 system.

Each volunteer completed 14 experimental runs: two of the runs were one-minute task, the two runs act as a baseline runs, they consist of, one run with eyes open, the other run with a closed eye, and other three runs that are two minutes long, each run consists of the four following tasks:

1- The volunteer will be facing a screen and a target will appear on either the left or the right side of the screen. Then the volunteer opens his fist and close it, but open and closes the corresponding fist until the target disappears. Then the volunteer relaxes.

2- Some target will appear on the screen sides the left side or the right side of the screen. Then the volunteer has to imagine that he is opening or closing his fist, but based on the corresponding fist until the target disappears. Then the volunteer relaxes.

3- A target will come out on the top or the bottom of the screen. Then the volunteer will open and closes either both fists only if the target is on top. Or both feet only if the target is on the bottom until the target disappears. Then the volunteer relaxes.

4- The final task is, some target will appear on either the top or the bottom of the screen. Then the volunteer will imagine opening and closing either both fists only if the target is on top. Or both feet only if the target is on the bottom of the screen until the target disappears. Then the volunteer relaxes (M. Tangermann, 2012).

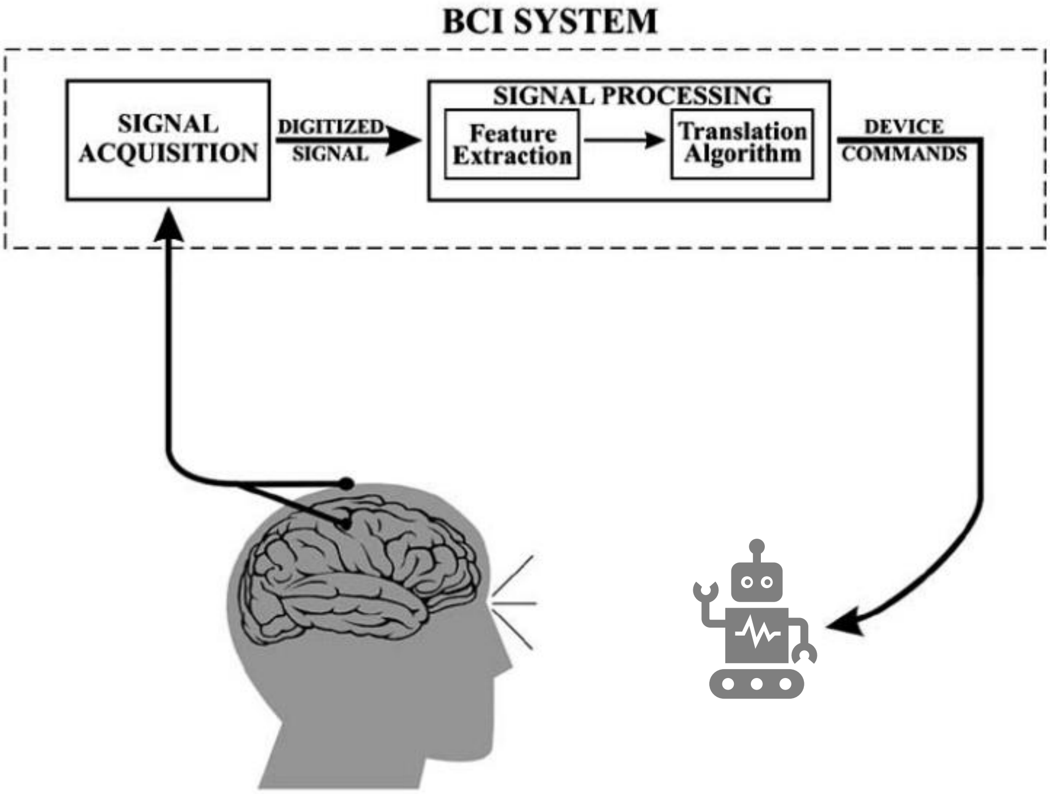


Figure 16: Basic design and operation of any BCI system.

a. Four class Motion Imagery

This data set consists of EEG data from 9 subjects, This EEG data consisted of four different motor imagery tasks, namely the imagination of movement of the left hand (class 1), right hand (class 2), both feet (cla ss 3), and tongue (class 4).

Two sessions were conducted on different days, the session has been recorded for each subject. Each session is consisting of 6 runs, each run is separated by short breaks. One run consists of 48 trials, 12 trial for each of the four possible classes, yielding a total of 288 trials per session. The recording of the EEG signals was divided into 3 parts:

- Two minutes with an open eye, while the subject is looking at a fixed cross on the screen.
- One minute with a closed eye.
- One minute with eye movements.

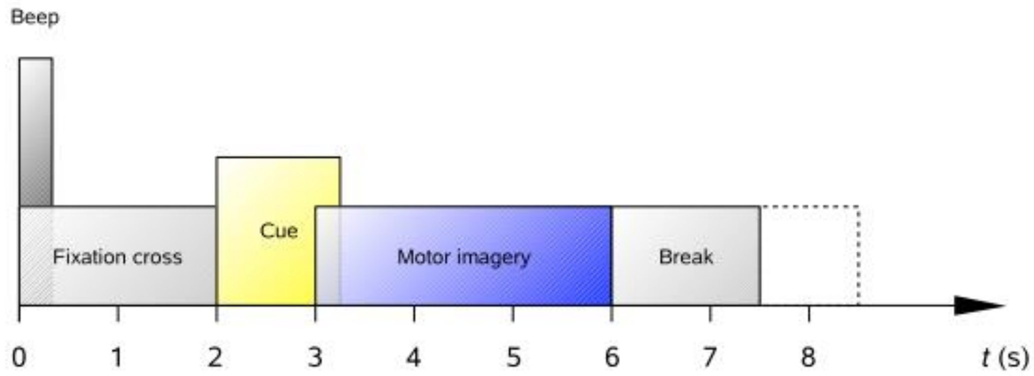


Figure 17: Timing scheme of the paradigm. (M. Tangermann, 2012)

The EEG data is collected by an apparatus that is capable of collecting 22 channels, the signals were sampled at 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz. All data sets are stored in the General Data Format for biomedical signals (GDF). one file per subject and session. However, only one session contains the class labels for all trials, whereas the other session will be used to test the classifier and hence to evaluate the performance. A GDF file can be loaded with the BioSig toolbox in Matlab or the EEGLab toolbox in Matlab as well (M. Tangermann, 2012)

b. BNCI Horizon 2020 by Graz University of Technology

This open access datasets are comprised of more than 25 motion imagery datasets. (Anon., 2014)

5.2.5.3. Emotion Analysis

DEAP dataset: This dataset is comprised of 32 participants were recorded as each watched 40 one-minute long excerpts of music video clips that are used as the visual stimuli to elicit different emotions.

A relatively large set of music video clips was gathered using a novel stimuli selection method.

A subjective test was then performed to select the most appropriate test material. EEG signals were recorded as they watched the 40 selected music videos.

Participants rated each video in terms of the levels of arousal, valence, like/dislike, dominance and familiarity. For 22 of the 32 participants, frontal face video was also recorded.

Due to licensing issues, we are not able to include the actual videos, but YouTube links are included.

This database has the highest number of participants in publicly available databases for analysis of spontaneous emotions from it is the only database that uses music videos as emotional stimuli.

A semi-automated method for stimulus selection was conducted, the main goal of this method is minimizing the bias arising from manual stimuli selection. 60 stimuli of the 120 initially selected stimuli were selected using the Last.fm3 music enthusiast website. Where Last.fm allows users to track their music and their listening habits and the user will receive recommendations for new music and events, based on his previous behaviour. Additionally, it allows the users to assign tags to individual songs, thus creating a set of classes of tags. Many of the tags are carrying out an emotional meaning, such as ’depressing’ or ’aggressive’. Also, the website is offering an API, allowing one to retrieve tags and tagged songs.

304 keywords as in tags were found in the website database. For each tag, the ten songs most often labelled with this tag were selected. This resulted in a total of 1084 songs. The valence-arousal space can be subdivided into 4 quadrants, namely low arousal/low valence (LALV), low arousal/high valence (LAHV), high arousal/low valence (HALV) and high arousal/high valence (HAHV). In order to ensure diversity of emotions, from the 1084 songs, 15 were selected manually for each quadrant, according to the following criteria: First, the tag has to reflect an emotional content accurately. Second, a music video has to be available for the song. Thirdly, the song has to be appropriate to use in the experiment. (Koelstra, et al., 2012)

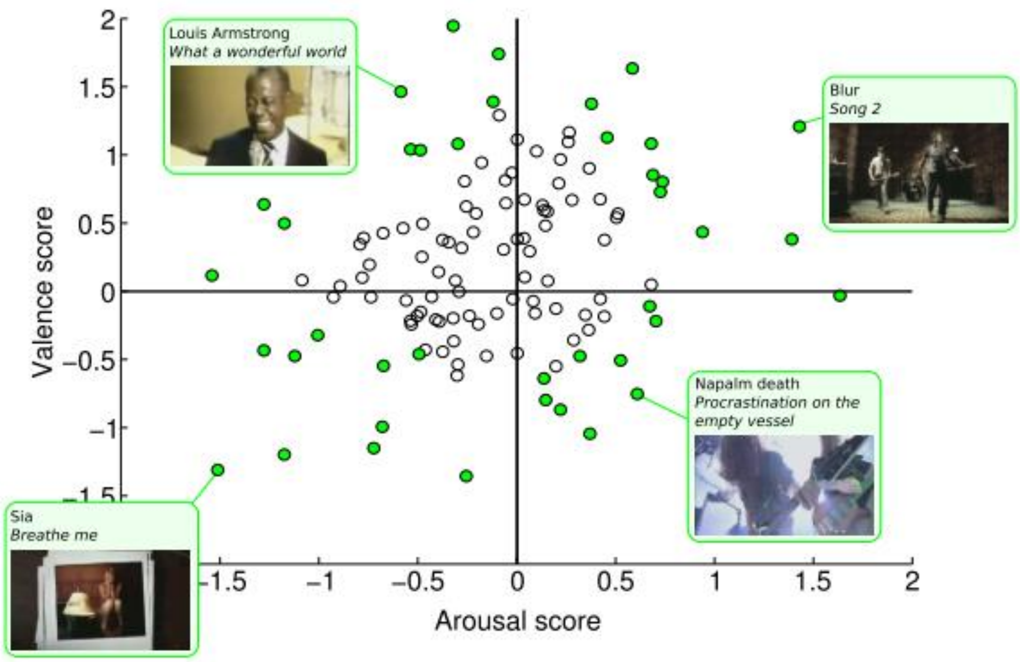


Figure 18: Value for the ratings of each video in the online assessment. (Koelstra, et al., 2012)

5.2.5.4. Showcases

a. Exercise in different environments

Heat stress and acute hypoxic exposure (lack of oxygen) are known to negatively influence endurance performance during exercise. Research suggests that hyperthermia may reduce the arousal, while hypoxia may alter neuronal transmissions. It is yet to be determined, how hyperthermia and hypoxia influence the cerebrocortical activity during self-paced exercise.

This study, therefore, aims to explore the exact nature of the cerebrocortical response under normal (cool), hot, and hypoxic environments during self-paced exercise.

Eleven men, well-trained cyclists, volunteered to take part in the experiment. They performed three 750 kJ self-paced run on the training bicycle, time trials were collected as quickly as possible under each environment. There was a 4 to 6 days interval between the time trials. The time trials were preceded by a warm-up in 20-22°C and 50% relative humidity. During the warm-up, the cyclists were equipped with sensors monitoring their heart rate and skin temperature.

The environments of the time trials were as follows:

1. Control/Cool: 18°C, 40% relative humidity, 12.5 km/h airflow.

2. Hot: 35°C, 60% relative humidity, 12.5 km/h airflow.

3. Normobaric hypoxic: 18°C, 40% relative humidity, FiO2 : 0.145, ~3000m, 12.5 km/h airflow.

The study found out that there are several changes in alpha and beta activity between the three environments during self-paced exercise. (Anon., 2014)

Figure 19: The Setup of the experiment (Anon., 2014)

b. Combining MEG and EEG recordings to optimize treatment for patients suffering from epilepsy:

Relevant physiological paradigms such as eye blinks were extracted and examined, along with checkerboard pattern reversal, as well as visual evoked potential (VEP) occurrence, moreover morphology and topography as well.

Twenty healthy adults were a volunteer for this study 10 of them were male and another 10-female volunteered as well. EEG signals were recorded. The population age ranged from 23 to 60 years old and they were free from any psychiatric or any neurological issues.

The experiment starting with five minutes resting then EEG Signals were collected while they are First, eyes closed then eyes opened, Then a 20 eye blinks for 40 seconds, and pattern reversal VEP according to ISCEV 2010 standard.

c. Rugby players cognitive performance

Volunteers in this study are professional rugby players.

For every volunteer or individual player, a total of 10-15 players are monitored each week. The EEG recordings are repeated between 2 to 4 times a week. To measure the speed and the accuracy of the player reaction and also to measure the decision making of every volunteer a two-stimulus oddball test has been used.

Behavioural measurements are combined with EVP event-related EEG metrics to provide a broad picture of the cognitive health of the player.

Also, an experiment of an Eyes-open and an eyes-closed and resting-state EEG are recorded and used as a psychological measure of mental alertness.

Six other systems and bio-data are also in use for this study. Six participants can be tested at the same time, while the full session’s recordings are collected within 20 minutes.

5.2.5.5. Sleep Dataset

Sleep Recordings: The recordings were obtained from 22 males and females (21–35 years old). Each subject took temazepam (a benzodiazepine) on one of the two nights. The developed estimators the two of them detected the effects of age, temazepam, and time of night on the quality of the sleep. Females were found to have twice the SWP (Slow-Wave Power) of males, but there is no gender effect on SW%

(slow-wave micro continuity) was found. These results confirm the earlier reports that are showing gender affects SWP but not sleep depth. Subjectively assessed differences SW% were correlated to sleep quality between the nights, not in SWP. These results demonstrate that slow-wave micro continuity, being based on a physiological model of sleep, reflects sleep depth more closely than SWP does. (B. B. Kemp, et al., 2000)

5.2.5.6. P300

P300 in disabled subjects: The data for each of the volunteers is contained in a zip-archive. Each file corresponds to only one run, this matrix contains the raw EEG, the dimension of the matrix is 34 × the number of samples.

Each of the 34 rows corresponds to one electrode. The ordering of electrodes is: Fp1, AF3, F7, F3, FC1, FC5, T7, C3, CP1, CP5, P7, P3, Pz, PO3, O1, Oz, O2, PO4, P4, P8, CP6, CP2, C4, T8, FC6, FC2, F4, F8, AF4, Fp2, Fz, Cz, MA1, MA2.

Each column corresponds to one temporal sample; the sampling rate is 2048 Hz.

The data is reference-free because it was recorded by a Biosemi Active Two system (Matthieu Duvinage, Thierry Castermans, Mathieu Petieau, Thomas Hoellinger, Guy Cheron, Thierry Dutoit, 2013).

The events matrix contains the time-points at which the flashes (events) occurred. In each of the datasets, the first event comes after 400 ms of the beginning of the EEG recording.

The variable target contains the index of the image the user was focusing on.

The system is based on the P300 evoked potential and is tested with five severely disabled and four able-bodied subjects.

P300 event-related potential. The P300 is a positive deflection in the human EEG, appearing approximately 300 ms after the presentation of rare or surprising, task-relevant stimuli.

Users were facing a laptop screen on which six images were displayed, the images showed a television, a telephone, a lamp, a door, a window, and a radio.

The images were selected according to an application scenario in which users can control electrical appliances via a BCI system.

The images were flashed in random sequences, one image at a time. Each flash of an image lasted for 100 ms and during the following 300 ms, none of the images was flashed.

The EEG was recorded at 2048 Hz sampling rate from 32 electrodes placed at the standard positions of the 10–20 international system.

Subjects 1 and 2 were able to perform simple, slow movements with their arms and hands but were unable to control other extremities. Spoken communication with subjects 1 and 2 was possible, although both subjects suffered from mild dysarthria. Subject 3 was able to perform restricted movements with his left hand but was unable to move his arms or other extremities. Spoken communication with subject 3 was impossible. However, the patient was able to answer yes/no questions with eye blinks. Subject 4 had very little control overarm and hand movements. Spoken communication was possible with subject 4, although mild dysarthria existed. Subject 5 was only able to perform extremely slow and relatively uncontrolled movements with hands and arms. Due to a severe hypophony and large fluctuations in the level of alertness, communication with subject 5 was very difficult (Ulrich Hoffmann, Jean-Marc Vesin, Touradj Ebrahimi, Karin Diserens, 2008) .

Figure 20: Images were flashed, one at a time (Ulrich Hoffmann, Jean-Marc Vesin, Touradj Ebrahimi, Karin Diserens, 2008)

5.2.5.7. Animals EEG

Rats EEG: This study focusses on the synchronization between the left and the right hemisphere in rat electroencephalographic (EEG) channels, by using various synchronization measurements, namely nonlinear interdependences, phase synchronizations, mutual information, cross-correlation, and the coherence function.

The measures that have been conducted in this study has a value for the study of synchronization in real data.

Although the data are EEG recordings from rats, their main features are common to human EEG too.

The EEG signals were obtained from electrodes placed on the left and right frontal cortex of male adult WAG/Rij rats. The collected EEG data was in 200 Hz, and the main objective of this setup was to study changes in synchronization between the two hemispheres, after unilateral lesions with ibotenic acid (the acid is making the rats uncomfortable and having mood swings).

For each rat, ten data segments pre and ten segments post-lesion were analyzed. The length of each data segment was 5 sec (R. Q. Quiroga, et al., 2002).

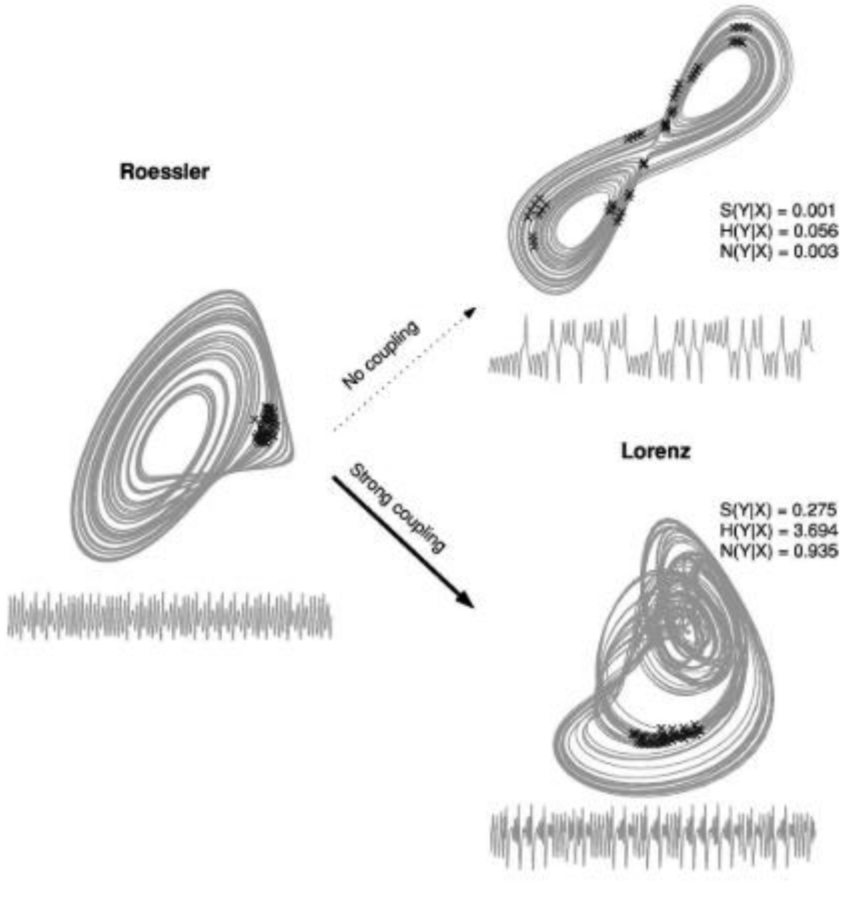


Figure 21: Basic idea of the nonlinear interdependence measures (R. Q. Quiroga, et al., 2002)

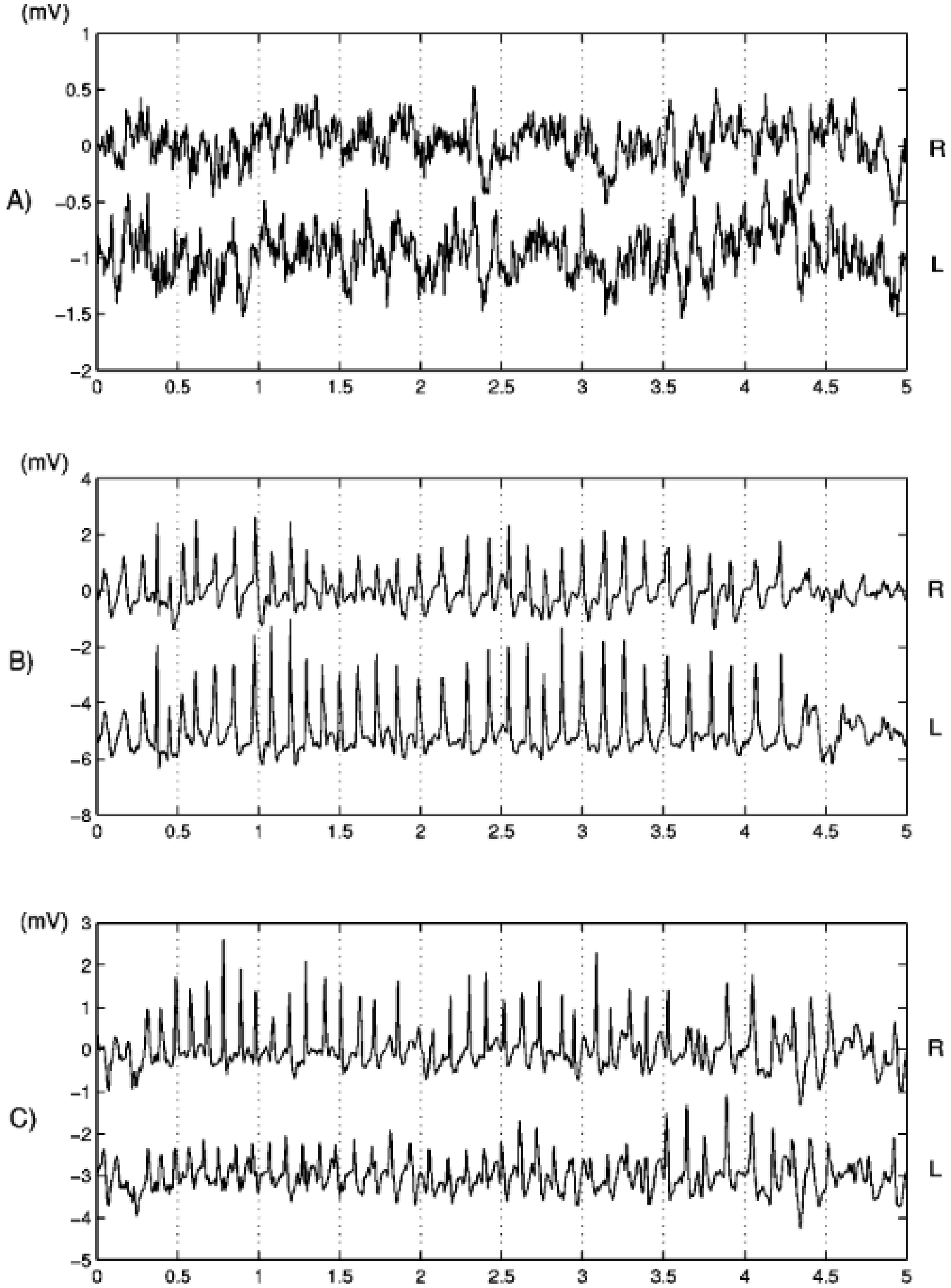


Figure 22: Three rat EEG signals from the right and left cortical intracranial electrodes. For better visualization, left signals are plotted with an offset (R. Q. Quiroga, et al., 2002).

5.2.5.8. Epilepsy data

The clinical purpose of these recordings was to delineate the brain areas to be surgically removed in each individual patient in order to achieve seizure control.

We included intracranial EEG recordings from five epilepsy patients. These recordings were performed prior to and independently from our study as part of the epilepsy diagnostics in these patients.

Multichannel EEG signals were recorded with the intracranial strip and depth electrodes all manufactured by AD-TECH (Racine, WI, USA). An extracranial reference electrode placed between 10/20 positions. EEG signals were either sampled at 512 or 1024 Hz, depending on whether they were recorded with more or less than 64 channels.

All patients gave written informed consent that their data from long-term EEG might be used for research purpose.

5.2.6. Evaluation

For evaluating the model, we needed to go throw multiple datasets, and look throw them, and we going to pick one of the datasets that are available publicly that contains the same data shape of our model or has the same experiment setup, so our classier can classify them and we get similar results, so we can compare the accuracy of the data that has been collected by us and the data that has been collected by other studies. So we can fairly compare the accuracy of our model. Making sure the experiment setup is the same is just a rational thing to do, because it will be entirely wrong to try to use a model that classifies an emotion on a data that has been collected for the goal of moving objects, the experiment setup is totally different, and the end goal of the model is totally different. In other words, it's irrational to use a screwdriver to hit a nail into the woods, it might work, but it's not efficient, different tools meant for different tasks.

That's why we picked this data set (Delorme, et al., 2004) University of California San Diego, this dataset contains a collection of 32-channel data from 14 volunteers (7 of them are males, the other 7 are females) the data has been collected using the Neuroscan software. Volunteers are performing a go-nogo categorization task and a go-no recognition task on an image presented for 20 ms only. Each volunteer responded to a total of 2500 trials. Data is containing the events and is sampled at 1000 Hz total data size is 4Gb. The images are representing natural scenes, there are multiple images categories, those categories have been chosen to be as diverse as possible. The animal category included images of mammals, birds, fishes, arthropods, and reptiles.

Volunteers seated in a room, the lights were dimly lit, at 110 cm from the volunteer there was a screen, piloted from a PC computer. To start a task, a volunteer had to press a touch-sensitive button. Then a small point was drawn in the middle of a black screen photograph that was flashed for 20 Ms Participants gave their responses following a go/no-go paradigm. For each response, they have to lift their finger from the button in a timely fashion and as accurately as possible. Volunteers were given 1 second to respond, after what any response is considered as a no-go response. More specially, in the animal categorization task, volunteers had to respond whenever there was an animal in the picture or not.

There was a wide range of non-target images, with outdoor and indoor scenes, natural landscapes or city scenes, pictures of food, fruits, vegetables, trees and flowers... In the categorization task, 500 distractors and 500 targets were seen by every subject but randomly distributed among all 10 series. In the recognition task, 750 distractors and 210 target photographs were used.

So, this data set seems quite similar to what we were doing so we will run the evaluation on this dataset and represent and discuss the results.

Running our model on this data set will take up to 66 hours per subject, giving that we have 14 volunteers so that will take 15.4 days of none stop running, the issue is not entirely the processing time but the machine specs as well.

So, we had to run our evaluation on 5 tasks only and on 5 subjects only, this will give us a good idea around the accuracy of our model. So, the confusion matrices table is looking like this:

Table 5: Confusion Matrices for model evaluation

	Class 1	Class 2	Class 3	Class 4	Class 5	Recall
Class 1	3	1	1			TP:3 FN:2
Class 2		4	1			TP:4 FN:1
Class 3		2	3			TP:3 FN:2
Class 4		2	2	1		TP:1 FN:4
Class 5					5	TP:5 FN:0
Precision	TP:3 FP:0	TP:4 FP:5	TP:3 FP:4	TP:1 FP:0	TP:5 FP:0	

First, we can calculate the accuracy which is a ratio between the correct guesses and the total number of guesses.

Accuracy = 3+4+3+1+5 / 25 => 16/25 = 0.64

Precision means the percentage of your results which are relevant. In a mathematical way, Precision = TP/(TP+FP).

Precision (Class1) = 1, Precision(Class 2) = 0.44, Precision (Class 3) = 0.42, Precision (Class 4)=1, Precision (Class 5) = 1

Average Precision = (1 + 0.44 + 0.42 + 1 + 1) / 5 = 0.77

Recall refers to the percentage of total relevant results correctly classified by your algorithm. In a mathematical way, Recall = TP/ (TP+FN).

Recall (Class 1) = 0.6, Recall (Class 2) = 0.8, Recall (Class 3) = 0.6, Recall (Class 4) = 0.2, Recall (Class 5) = 1.

Average Recall = 0.64

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution.

F1(Class 1) = 0.749, F1(Class 2) = 0.56, F1(Class 3) = 0.49, F1(Class 4) = 0.33, F1(Class 5) = 0

The thing with F score, as smaller the value as better that’s why we can see the best f score is for class 5 which is 0 zero and that’s because the class 5 is the most accurate class of them all.

5.2.1. Results Discussion

5.2.1.1. Training the models

The way that this study trained the model is straight forward which is passing all the training data to all the models, then each model applies the filters before applying the classification algorithm.

Table 6: Models predictions for the first volunteer, Pl is the Predicted Label, prob. Is the probability

#	True Label	Model 5		Model 4		Model 3		Model 2		Model 1	
		Prob.	PL	Prob.	PL	Prob.	PL	Prob.	PL	Prob.	PL
0	315	47.23	315	41.00	315	32.11	315	30.99	315	10.60	315
1	314	46.91	315	41.08	315	37.98	315	31.71	315	15.57	315
2	315	31.89	315	38.69	315	30.38	315	25.95	315	16.84	315
3	315	23.17	315	36.63	315	19.47	315	31.62	315	12.44	315
4	291	49.26	291	42.36	315	48.61	315	31.27	291	10.97	315
5	282	46.48	315	41.85	315	41.40	315	31.66	315	14.36	315
6	315	54.05	315	42.91	315	53.67	315	31.60	315	14.73	315
7	315	39.55	315	39.87	315	39.58	315	31.70	315	17.60	315
8	270	33.32	315	38.11	315	10.70	175	30.32	315	14.09	315
9	315	56.86	315	43.77	315	45.04	315	14.69	165	16.98	315
10	259	24.46	315	35.49	315	10.83	175	31.35	315	14.12	315
11	246	32.04	315	38.80	315	31.84	315	10.70	191	15.47	315
12	315	23.99	315	35.29	315	14.69	315	14.63	315	7.38	282
13	315	53.01	315	43.89	315	50.68	315	31.69	315	7.71	270
14	315	39.02	315	39.81	315	14.30	315	30.69	315	9.09	315
15	315	48.02	315	40.52	315	49.64	315	31.67	315	16.86	315
16	315	25.69	315	34.94	315	11.52	165	31.60	315	7.87	315
17	315	27.81	315	37.06	315	22.37	315	31.71	315	10.83	315
18	315	46.67	315	41.62	315	54.72	315	31.66	315	8.78	315
19	315	34.12	315	38.64	315	16.35	175	30.64	315	15.41	315
20	315	53.99	315	42.73	315	40.16	315	31.56	315	8.74	315
21	194	54.05	315	42.05	315	41.71	315	31.55	315	23.49	315
22	191	21.12	315	36.01	191	18.67	315	31.33	191	10.46	191
23	185	44.32	315	40.24	315	25.41	315	31.52	315	18.86	315
24	182	25.96	315	36.46	315	19.79	315	30.56	315	10.67	315
25	315	28.34	315	37.25	315	31.76	315	31.50	315	9.88	315
26	170	29.13	315	38.32	315	32.12	315	13.31	315	6.25	182
27	315	14.58	315	32.61	315	21.15	165	31.22	315	9.01	315
28	315	25.59	315	35.62	315	13.06	175	31.32	315	6.41	175
29	165	19.54	315	34.32	165	12.63	175	31.35	165	13.71	315
30	169	46.87	169	42.77	315	30.89	165	6.25	169	6.67	165
31	175	34.67	315	40.24	175	45.94	175	31.71	315	13.55	315
32	175	45.69	175	42.79	315	44.70	315	31.73	315	13.59	315
33	181	28.15	315	36.31	181	31.40	165	31.67	181	6.55	181
34	185	20.22	315	39.34	315	19.94	194	31.21	315	6.66	314
35	194	19.57	315	34.56	315	16.60	175	29.55	315	6.77	291
36	315	38.19	315	40.35	315	48.08	315	23.32	315	12.20	315
37	315	33.43	315	39.53	315	47.59	315	30.69	315	6.98	185
38	315	28.60	315	38.12	315	28.88	165	25.15	165	7.04	181
39	315	24.50	315	37.77	315	15.54	175	31.06	315	6.45	282
40	315	30.55	315	39.03	315	27.64	315	5.26	315	6.90	194
41	315	37.99	315	39.95	315	43.43	315	31.40	315	8.91	315
42	315	56.06	315	44.09	315	51.20	315	25.38	315	15.10	315
43	315	50.28	315	42.21	315	46.28	315	31.52	315	15.53	315
44	237	48.24	315	43.43	315	56.62	315	31.11	315	17.73	315
45	315	13.97	315	31.31	315	31.21	165	31.63	315	6.39	259
46	249	32.90	315	38.83	315	15.00	175	30.61	315	7.67	185
47	315	33.35	315	38.73	315	38.52	315	31.17	315	18.09	315
48	256	60.45	315	44.76	315	42.61	315	31.51	315	14.06	315

There are different ways we can get the final result from the table 6, a list of options will be presented:

5.2.1.2. Mode

We can rely on the mode, the mode is a statistical term that refers to the most frequently occurring number found in a set of numbers, in other words, we can count the most repeated class across all the models, as a majority voting and consider it as the final result. Relying on the count might be better as some models have a consistently high probability, which will control the final result all the time.

5.2.1.3. Probability

What should be the action if we have the same count, for example, model 1 and 2 predicted class A, model 3 and 4 predicted class B and model 5 predicted C, in this case, we have an equal count between class A and B, then we should also consider the probability as a parameter and get the average prediction per class.

5.2.1.4. Weighted Probability

Let’s say in our experiments model A shows better performance than model B, if we used the average probability per class then we are omitting the fact that we should consider the result from model A a little bit more than model B, so we should give model A a weight of 2 for example and model B a weight of 1, then multiply the average to the weight before considering it as a final result.

5.2.1.5. Stacking

Rather than putting hard criteria on how we should select the final result from all the previous models, we can delegate this to another new meta-model stacked on top of main 5 models to predict the most accurate result

5.2.1.6. How the accuracy is calculated

In table 6, you can see all the models predictions for the first volunteer, in this ensemble model architecture we elected the top prediction from each model, for example in the row number 0, model 5 prediction is 315 with a probability of 47.2% these probabilities are multiplied by a hundred to make it easier for the human eye to read, but the actual prediction from the model is following the proper statistical terms, which is between 0 and 1. After each model has populated the predicted class and the prediction probability, then the algorithm will select the highest probability from across all the models, this will be the final result for the trail.

There are many ways to train the models like:

5.2.1.7. Bagging (Bootstrap AGGregatING)

We have 5 different models, we can split the training data to 5 smaller subsets randomly, then pass the smaller training subset to each model, at the end we use their collective output to determine the final result, by doing this we make sure none of the models is overfitted.

5.2.1.8. Boosting

Rather than splitting the training data, we can pass the whole training data to the first model, this model will not have a 100% accuracy on the training data, it will still make mistakes, the next step is to take the poorly predicted data and pass it to the next model, and so on till we have a good accuracy. This way is showing better performance than bagging, but it's suffering from a high potential of overfitting the models. As the previous way (bagging) we use the collective output of the models to determine the final result.

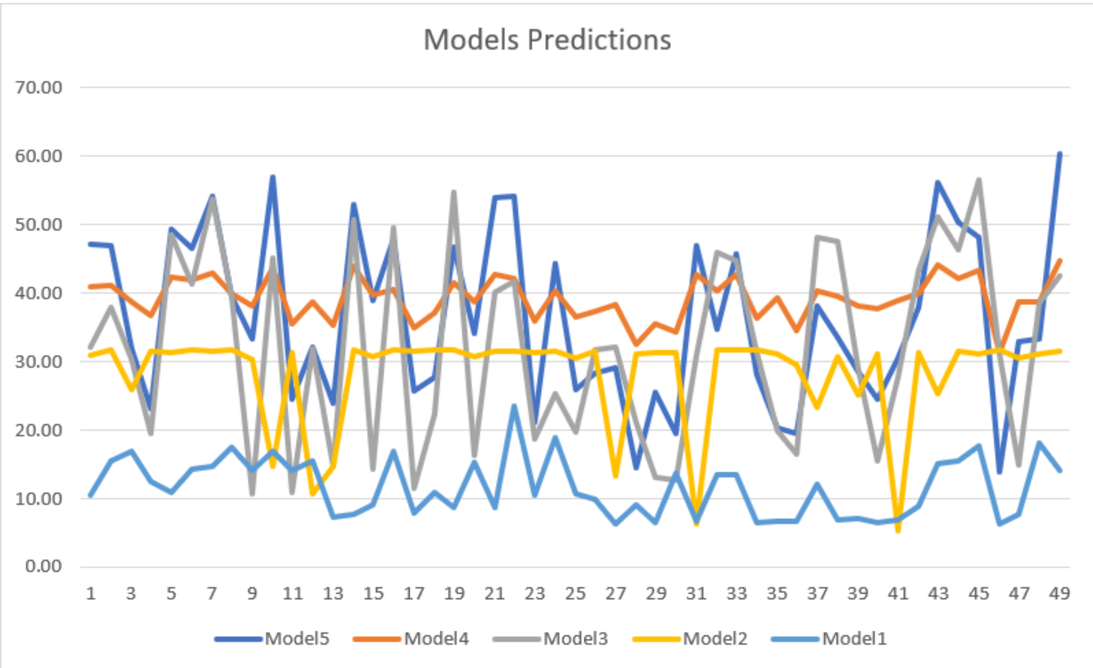


Figure 23 : Models predictions probability for the first volunteer.

As you can see from figure 23, model 1 seems to have a low accuracy across all the predictions, it depends on the examination if the model 1 usually have better predictions than the others, we can use the weighted prediction approach and give model 1 a higher weight.

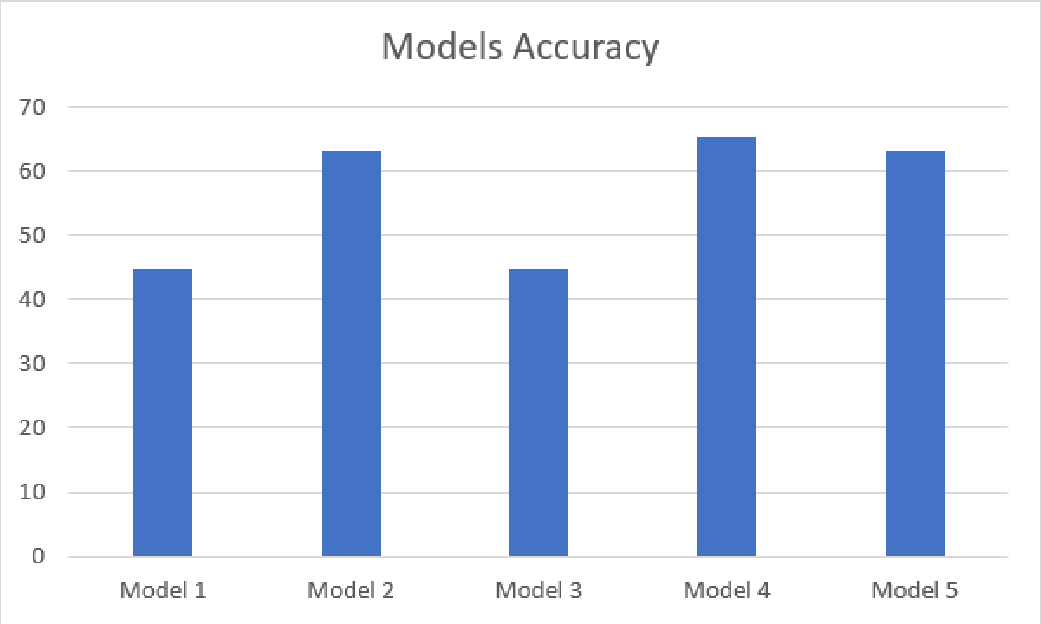


Figure 24: Models accuracy for volunteer 1.

5.2.1.9. Some Observations

Figure 24 is derived from table x the accuracy is based on how many class was predicted right, as you can see, 3 models have an accuracy of ~60% plus but the ensemble model accuracy is ~71% which indicates that the ensemble model is increasing the accuracy by ~10% and making the model more robust and making sure the models are not overfitted.

One might argue that in the current case model 1 have a low prediction percentages and low accuracy, so in this case, it seems useless and not contributing to increasing the accuracy, so it's better to remove it from the ensemble, to reply to this argument we have to look at why we add this model in the first place, model 1 is using the Hankel filter which skews the data using the Hankel transformation which is presenting a new shape of data that no other model can see, keeping model 1 in the ensemble will result in a more robust model and reliable results.

5.2.1.10. Future tuning

In this work, we are trying to create a robust solution for emotion recognition that can predict the emotion regardless of the application, but if there is a specific application in mind this model can be tuned more to fit that specific application, for example, if the application is producing long trails or short trails, there are many things that can be tuned here, the models filters, the algorithms parameters (hyper parameters) the filter parameters, the number of the models, and the final result for the ensemble.

5.3. Possible applications

There is a lot of applications that can benefit from this work, we will try to imagine some of them here, but the range of possibility is endless.

- EEG can be used to predict are this customer is willing to buy the product or not. Since we have an access to the cognitive state or the emotion of the person we can detect the excitement to determine whether this person I willing to buy the subject or not, in other words, we can link the excitement emotion with the buying activity or we can list other emotions as well, if one of them has been detected we can say the person is a buyer.
- EEG can be used to detect shapes, a user can imagine the shape and the model will detect it, simple shapes like a circle or a rectangle or square, this can be useful for many applications that are requiring users to draw, simplifying this process will make it very fast and continent.
- We can use it to measure IQ by the effectiveness of performing certain cognitive tasks.
- Detecting the awe moments, which is when someone understands something or realize something new, which is a different kind of emotion, it might seem the useless thing to try to detect but it will help us build a better model that can understand human emotion better.
- EEG can be used to determine if this image is appropriate for kids or not, a volunteer can just look at the picture and it will directly be classified as appropriate or not, which will reduce the time of the classification drastically.

- Can be used to predict which song to play at the current moment based on the human current emotion.
- Getting emotional feedback whether the volunteer liked the song or not.
- Or even helping the volunteer reaching a certain emotion by picking songs to evoke that emotion in him.
- Emotion detection can be used as a tool to enhance other aspects of AI, for example, we can enhance the music that is generated by AI, by detecting the emotions of the humans who are listening to the generated music and adjust it accordingly. By creating this feedback loop, the AI-generated songs can be drastically better.
- Mimic the game's joystick, so users can play games with their minds.
- Predicting the number that the volunteer is thinking about.
- Can be used to classify emotions as in emojis
- Classify whole words rather than characters, it’s not realistic as per the current level of technology, but it’s faster than spelling the word.
- Moving an object in a 3D world by detecting 12 directions.
- Another application might try to study how many emotion humans are capable of detecting in other humans and set this as an upper standard, or a goal for the models to reach, that can also be used as a test for the accuracy of the new models.
- We can exaggerate certain peeks or valleys in the EEG signals to get a more obvious signal for better classification.

5.4. Summary

In this chapter we explored different possibilities of testing and evaluating the machine learning models as well as running the proper ones on our model, we also explored different range of EEG datasets and we explained briefly the purpose of each one and how they collect the data and whether it will help in our evaluation or not, then we choose one data set and we ran our model into one of the datasets and explore the results. We finally listed some of the applications of detecting the emotions, and how it can contribute to making the world a little bit more continent place.

CHAPTER SIX

6. THEORETICAL MODEL AND FUTURE ENHANCEMENTS

6.1. Introduction

6.2. Motion Imagery

Motion imagery is targeting to convert brain signals to motion, in other words allowing subjects to move the robotic arm for example by using their brain waves. There are many types of motor imagery (MI), sensory interface or cognitive interface or motor interface (V. S. Kota, 2017). The state-of-art research in this field can differentiate between movements with an accuracy of 90% (Qibin Zhao, et al., 2009) and once you differentiated between cognitive functions you can use it for many things in life, like moving a remote-controlled car (Pan Peining, et al., 2018) or controlling a quadcopter or a drone (Picard & R. W., 1995) or even use it instead of the joystick in computer games, for a high accuracy and clinical trials it can be used to control a robotic arm, but this kind of accuracy requires invasive EEG that is installed by a medical professionals called Electrocorticography (ECoG).

6.3. Emotion Recognition

In emotion recognition side scientist has done fair amount of research as well despite the fact that emotion recognition is inherently not scientific, in other words, it's subjective and full of biases, there is no clear way to tell how the subject is feeling, not even the human himself, as his answer is considered biased as well (Byun, et al., 2017), compared to the motor imagery MI it's very clear to tell where is the subject limb (the limb that under the test for example, his arm location in the 3D space), this limitation in emotion recognition makes it harder to check the accuracy of the model, correctly most of the studies that are doing this kind of research has the same structure, same structure in terms of the experimental setup, and for creating the model and classifying the emotions.

6.4. Emotion Recognition Studies

Most of the EEG experiments have the same shape in terms of the experimental setup and the signal classification models (L. R. C. Mohamed Ahmed Abdullah, 2019), which means that the field desperately needs a novel way of either doing the experiment or classifying the signals. The first step is to elicit emotions by using external stimulus by getting this in motions into the subject by any ethical means, some studies using images some of them using videos (Y. P. Lin and T. P. Jung, 2014) audios (Ou Lin, et al., 2015) or games (Schneider, et al., 1997), or any social situation to trigger these emotions into the subject. while the subject is experiencing this in emotion, brain signal must be collected, then we are asking the user how you felt when you were doing this task, and how are you felt doing the other task, and so on. then we tag the brain signals with the emotion that the user says that he is experiencing. then some feature extraction will be collected from the brain signals, to skew the data to make it easier for the classification algorithm to detect the difference, then we pass this data to the classification algorithm. the accuracy of the model will be calculated from what the user said he is feeling at that moment. There are many feature extraction techniques can be applied to the signals, for example, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Fractional Dimension, Common Average Referencing (CAR), Surface Laplacian (SL), Common Spatial Patterns (CSP), Autoregressive (AR), Autoregressive Moving Average (ARMA), Multi-Variate Autoregressive (MVAM) and many more (L. R. C. Mohamed Ahmed Abdullah, 2019).

6.5. Measuring the accuracy is tricky

Measuring the accuracy of any model depends on the true value or we also call it the observation, in motor imagery we can easily observe undertake the location of the moving limb, of course depending on this study this moving limp could artificial or it could be human, so this observation will act as the true value in our machine learning algorithm, comparing that to the emotion recognition field, this true value is subject to a biased from the human that is under test, as we ask him how do you feel, then we take this as the true value, and we directly feed it as an observation in our machine learning model. and also, if we compared those two in terms of effort, we will notice that it's much easier in motor imagery than emotion recognition, which means it's easier and more accurate to collect data for motor imagery than emotion recognition.

6.6. **Challenges**

There are multiple challenges for collecting and classifying EEG signals in general and some challenges specifically for emotion recognition.

6.6.1. EEG Challenges

6.6.1.1. Nature of EEG data

There are so many reasons resulting in a distorted EEG data signals, the first reason is the skull and the scalp reducing the signal clarity, as well as the nature of the EEG is a data collected from the surface of the brain, while the part of the brain that is responding to the emotions is at the center of the brain, which is the amygdala (Y. Gao & H. J. Lee and R. M. Mehmood, 2015). In addition to that collecting the data from the surface is hiding the true source of the signal. as well as the EEG signals are different from person to another, in other words, if we collected data from two subjects, subject A and subject B, the definition of happiness in subject A is different from the definition of happiness in subject B. the brain waves shape will be different. This fact raises a huge problem where an EEG must be trained on a specific subject to get high accuracy. Some studies are trying to do some intra-subject classification, but the accuracy is always lower, it is low to an unacceptable level.

6.6.1.2. Data collection

Getting the right apparatus and setting up the environment for collecting the EEG data expensive and time-consuming, and also cleaning this data and tagging it will take even more time.

6.6.1.3. Experimental Setup

Most of the studies are using the SAM self-assessment manikin technique for getting the true observations to train the model, and to check the accuracy, taking the survey and labelling the EEG data with the SAM results is not an easy task. So, one of the main purposes of this study is to reduce the dependency on the SAM technique.

6.6.2. Advantages of motor imagery (MI) over emotion recognition

why are we comparing motor imagery with emotion recognition anyway? they are both ways to classify EEG signals, and both considered as a part of BCI (Brain-Computer Interface), and both using the same technique for classification.

6.6.2.1. *Brain* Mapping

It's not fair to compare motor imagery (MI) and emotion recognition when it comes to classification accuracy, as there is a one-to-one mapping between a certain area in the brain and a hand movement for example (B. J. B. B. & . H. B. Edelman, 2016) the state of the art in this type of recognition is around 90% it's not fair to compare this with emotion recognition is it's easy to be classified.

6.6.2.2. *Closed Loop (Neuro feedback)*

Another advantage of motor Imagery which is the closed feedback loop, that allows the machine learning model to be trained as well as the human brain to adapt as well, in any motor imagery set up there is always a part of the machine or the human is moving and the human under the study is looking at it, and have information about the location of that part at all times, this is considered as feedback loop to the Human to correct his cognitive signal and adapt to the machine, and the model will always try to adapt as well. This closed-loop is very important for its motor imagery accuracy.

6.7. **Building a conceptual model**

EEG classification the suffering from lack off data, it is a lot of open-source datasets, but it requires a lot of data to train the models. You can't get one of the datasets to dump it in a neural network and you will get an accuracy off around 70% the state-of-the-art accuracy is around 80%, there's a huge room of improvement (L. R. C. Mohamed Ahmed Abdullah, 2019). So the idea here is to develop an EEG model and another bio-signal model and trained them both at the same time but with a loss function that will treat prediction from the first Model as an observation for the second model and vice versa, this will make sure that the promise of both models is consistent.

6.7.1. Experimental setup

There are many open access EEG datasets, but we can't use any of those datasets as they only provide EEG signals, without any bio-signals. The closest dataset is SEED (Anon., 2002) as they provide an eye movement signal along with the EEG signal but that's not enough for our experiment, it has to be a combination of both signals that can lead to high accuracy. so, we must do that data collection ourselves. It's important to keep the task that will trigger the emotion as small as possible as measuring the emotion of a long period is not recommended. That's why we should use here and image dataset that is designed to trigger a range of emotions into the subjects. and we recommend the humming window to be 2.5 seconds (Mohamed Ahmed Abdullah & Lars Rune Christensen, 2019). Showing the image and reading the signals should be done using the same application. the purpose emphasizing this point is to make sure that the trails are as clean as possible, and the data that is gathered from The Experiment is reliable. We recommend to use more than 6 healthy subjects in this experiment, and collect more than 120 trail from each subject as less than that will not give good accuracy in the EEG model, also because EEG signals are subject sensitive in other words they are different from subject to another. We can achieve higher accuracy in the bio-signal model by using less data than that but it's important to collect data to get a high accuracy from both models. To increase the signal accuracy and to reduce the signal distortion subject have to sit still without any movement they can't speak during The Experiment time there are also recommended to minimize the blinking time this artefact can be removed later from the signals but it's better to avoid it in the first place, and also the environment should be quiet and the temperature is controlled to minimize the effect of those external factors on the EEG signals.

6.7.2. Theoretical Novel Model Architecture

This architecture relies on two different models the first is an EEG model and the second is a bio-signal model. The EEG model consists of 5 different models as per (Mohamed Ahmed Abdullah & Lars Rune Christensen, 2019), they are trained together, and we use the collective result from all of them. The bio-Signal model is gathering the data from a combination of different bio-sensors eye movement signals, Electromyography (EMG), Skin Conductance (SC) and Skin Temperature (SKT). It consists of simple SVM model with a feature extraction technique, each of those bio-signal apparatus is considerate as a feature in the model (Choubeila Maaoui & Alain Pruski, 2010) (Anon., 2002).

6.7.2.1. EEG *model* and the Bio-Signals Model

1) ***EEG Model Description:***

Two types of models can be used, the first one we will call it classical model, which is an SVM or logistic regression with some feature extraction techniques this type of model is good when we have few data but the problem of this model that if it's complex to set up and to understand as well as there is no neural network here. the second type of models is the neural network models especially the deep learning models with a CNN and RNN layers (Y. Gao & H. J. Lee and R. M. Mehmood, 2015), this type of models is recommended to use here as it's easy to set up and has higher potential for benefiting from this architecture.

2) ***Bio-Signals Model Description:***

In this work (Andreas Haag, et al., 2004) they used a neural network with logistic regression and sigmoid activation function for the hidden and the output layer and resilient propagation as the learning function. This work trained two different models, one for valance and the other for arousal. This work has an accuracy between 89% and 96%, although it's using more apparatus than what we recommended in this work, it's using Electromyography (EMG), skin conductivity (SC), Skin temperature, Blood volume pulse (BVP), Electrocardiogram (ECG) and Respiration. This is a good fit for the bio-signal model as it's using neural network and it has high accuracy, although when we use recommended apparatus the accuracy will be reduced as the features array will be reduced as well, resulting to a less data hence less accuracy. But we don't anticipate a huge reduction in results.

6.7.2.2. Swapping observations

We have two sets of models that are predicting emotions, the EEG models and the bio-signal models, each one of those predicting results for a single trial, this prediction can be similar or it can be different, in case of similarity both models will be rewarded, but if they are different then both of them will get penalized, the way to do this is to treat the first model prediction as an observation for the second model, and the prediction of the second model as an observation for the first Model, this will result in a consistent results between the two models. Which means the truth is what both models agree on, and this will result in a stable model.

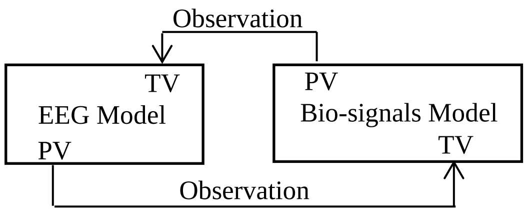


Figure 1: Observation swapping

Where PV is the predicted value and TV is the true value.

6.7.2.3. No need for SAM

Self-assessment manikin SAM is a survey to collect the human on emotion when a certain emotion has been invoked, it's defining how we should measure the emotion, by measuring pleasure arousal and dominance, the survey will be filled up by the subject under test and the result of the survey will be treated as an observation for the model. list method takes a lot of time and effort and the results are subject to the human biases, it's a big impediment in the advances of the field. by using this proposed architecture, we don't have to ask the subject how do you feel every time we show him a picture or make him listen to music, or any task we ask him to do, both models will try to predict the emotion and they will come up with a conclusion, this approach will save us last time and effort and it will be easier to gather more data, as the time of the experiment will be optimized.

6.7.2.4. A closed-loop is formed

The main intention of this model architecture is to form a closed loop between the emotions prediction and the subject and removing SAM from the picture, this closed-loop is inspired from the motion imagery Close loop, as in motion imagery you don't have to ask the user " where is your arm" in case of predicting the arm location for example. the closed-loop will result in the higher optimization of the subject time and he can wear the apparatus and totally forget that he's under experiment, which is making the whole process convenient.

6.7.2.5. Range of Emotions

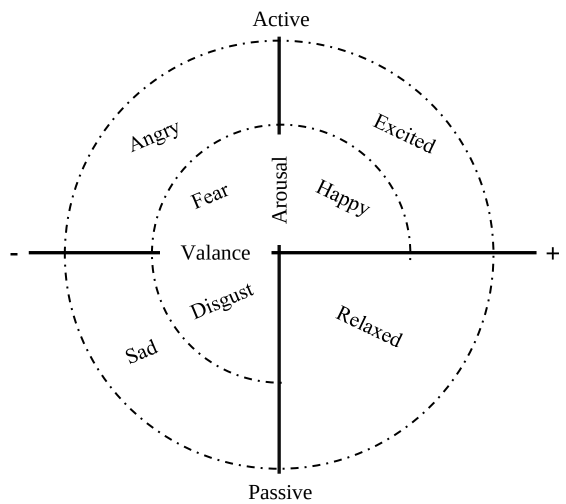


Figure 2: Optimum emotions range of emotions to be detected (S. G. Mangalagowri and P. C. P. Raj, 2016)

When we are saying were predicting emotion we're actually protecting a Vector of arousal and Valence for example now let's say they are a percentage, then we love those metrics to the actual emotion that we are trying to protect in our case it's a happy, excited, fear, anger, sad, disgusted, and relaxed. the dotted circle in figure 2 is identifying the emotions boundaries.

6.8. **Conclusion and discussion**

Conducting this experiment and running this model have a lot of challenges, one of the main challenges is the cost, The cost in terms of dollars can go up to USD 4k, and in terms of time it requires a setup for all the biosensors in each subject, and also cleansing and ordering this data and Tag it with the emotion is a major task and time-consuming. Some of the technical challenges is running those two models the EEG model and the bio-signal in sync and use their predictions as an observation for each other. another challenge here if the model has to be online, which means training the model has to be fast and the subject has to know the prediction of the model, as this would be an indicator to the subject that's whatever you felt just now will be classified as this emotion, this feedback loop is important to know the progress of the model through time, and to allow the subject to adapt and to form an understanding of how the classification model behave. SAM model is a hindrance as it's stopping this learning process to ask the subject how do you feel, and this simple question requires a lot of work to calibrate to that answer. as well as it cannot be online as we must stop the brain signals readings while the subject is giving his feedback. The study is intends to enhance the EEG emotion recognition from the bio-signals, but we can't rely on the bio-signal model prediction alone as it might be faulty and not accurate, it's not the source of truth, that's why the source of truth is the combination of the two models predictions. Also running this experiment will increase our collective understanding of the relationship between the EEG signals and the other bio-signals, this understanding will lead to a better prediction in the future as we can tweak our models to be more alerted to certain types of behaviors in the data. Running this experiment will also show as the inter-subject accuracy as it's always lower than the models that have been trained on a particular subject, those models are optimized for that particular subject hence they have higher accuracy.

CHAPTER SEVEN

7. CONCLUSION

7.1. Introduction

In this chapter, we will summarize the final discussions and conclusions and list the contribution of this study.

7.2. Conclusion and discussion

Mapping the SAM experiment results with the EEG data require a lot of work and it requires precision and carefulness, especially with tagging the EEG data with the SAM feedback many of studies dropped some of the collected data because it contains low-quality signals, also the EEG data has to be tagged with emotions of the subject himself because the emotion is different from subject to another. Cleaning the data and tag it for the classification is a mandarin task thus it's a progress hindrance. Also, there is no closed feedback loop to enhance the accuracy of emotion detection, by closed feedback loop we mean a method to speed up collecting the data and tag it for classification and a way to make the classifier learn in real-time. Another point is people are deferent from each other's (Y. H. Liu, et al., 2014) that's mean the EEG data is unique from person to another and the tanning of the classifier has to be person-specific which is a problem if there is an application require detection directly without the possibility of doing the training session first. Furthermore, there is no general agreement on which feature Ex. (channel or frequency band…) are best to measure which emotion (emotion to features mapping). Moreover, humans can have more than one emotion at the same time (B. H. Kim and S. Jo, 2018) so currently, there is no way to classify more than one emotion with probability Ex. (60% existed 40% scared).

Based on the study setup and the different models that have been adopted the accuracy is ranging from 50-70 depending on the distinct emotion that the subject selected as the range widens the accuracy shrinks. Considering the training data for the models which were one session the models were able to produce a very height accuracy, but in another hand, we noticed the model generation is a slow process due to the multiple models being trained per subject that can be enhanced in the retraining phase by focusing on the most accurate model. The relatively low number of volunteers in this study makes it harder to see the value of other models, not all the models are the best fit for each subject. This architecture needs to be challenged more by using more classes and more data for more accuracy. By changing the setup to be more than one session for training and only one for testing this will increase the accuracy. Finally, to collect this data it is better to consult an ethical board and collect a written approval rather than a verbal one. It’s possible to predict emotions from EEG signals but that doesn’t mean we understand emotions we just noticed a phenomenon then we take the best advantage of it by cleaning the signals and selecting the best signals and then detect the patterns but this field need more input from the neuroscience and the medical field to flourish the BCI and HMI field.

7.3. Future work

There are enhancements ideas that can take place, some if them might be easy to implement, others might need a lot of research before considering them, but we will list them here:

- This architecture is fit for window-based analysis (Trials) and the window for the current setup is 2.5 second for online analysis this need to be reconsidered.
- Another thing could be enhanced could be in the model selection, for the current setup we select the best model per subject a better approach could be picking the highest probability across the models but now we need to consider the retraining because if we relied on all the models we have to retrain all of them every time which computationally expensive.
- There is no understanding of how the emotion looks like, for example, what anger will look like, or what fight or flight response will look like, we need advances in neuroscience and phycology to identify what are the signals will be shaped like, then we can apply this knowledge in machine learning to fill in the gaps and get better emotions detection, but now what all the research are doing is just guessing. And relying on models that are optimized for a certain subject to solve a certain problem, without any real insight about human feelings.
- Model to choose the model: if we don’t know what shape of signal represent which emotion and we don’t know what filter is the best to highlight that emotion, then rather than doing a lot of research on what model can work can delegate this responsibility to another model, its another ML problem.
- Currently, we are measuring the model accuracy by scoring it based on whether it recognizes the right emotion or not, it’s a step hard function, the emotion is

not black or white, it's gradient, so we need to measure the accuracy with respect of the emotion distance, the distance between the true value and the predicted value, as closer this distance as higher the score, as further this distance as lower the score.

- The accuracy should also consider If more than one model predicted the right result.
- The lack of EEG data makes it hard to know if the any of the studies has an overfitted model or not, this is more like a problem than a future work, but it's something can be enhanced on.
- The EEG models are subject-specific, If model is trained on subject A, and we run it on subject B directly, without retraining it on subject B signals, the accuracy will definitely drop, but there is no research in the field discussing how much is the drop and how it will drop, the field needs this analysis, will it be able to classify certain emotions, is it a consistent drop over all the classes or in certain classes, all this is still standing questions.
- It's better to classify arousal, valance and dominance than classifying emotions like (happy or sad or surprised), because the core of emotion is the vector arousal, valance, and dominance, from those we can map them to an emotion, for example in the scale of valance, we can disagree where is the happy start from and where should it end, but the valance is just a number isolating the objectivity of a certain emotion. So, the model prediction should be the emotion vector, not the emotion itself.
- Grid Search needs to be applied here, on the filtering algorithms VS classification algorithm VS link between emotion and model VS link between emotion detection and electrode all this can be considered as a hyperparameters for the emotion model, and a grid search is needed to identify which combination is better for which emotions, but this can be very costly and time-consuming.
- This field needs a number that represents "what is the human to human accuracy of detecting the emotions?" this number is important so it can set the bar high, and considered as a benchmark for all the studies to try to reach and drive the field forward. This field has the potential to detect emotions better than humans, as it has an access to the bio-signals of the subject.

- Instead of doing a complex costly grid search, we can calculate the confusion matrix between the frequency band (wavelet) and the channels to select only the most effective electrodes.
- This field requires a study in the emotional memory, to answer some questions like how much time it takes from human to switch from happiness to sadness or another kind of emotions?, this kind of insight is important to optimize the experiment setup and top optimize the humming window, and to optimize the classification for example (if the classified emotion doesn't follow the general rule we can fall back to the normal feeling or the recommended feeling).
- We can apply the VR here as it's famous for emerging the user in rich emotional experience, definitely, it's a better tool than showing the users images or videos to stimuli emotions.
- Motion imagery or motor imagery MI has higher accuracy than general emotion recognition, so why not employing the power of MI in emotion recognition? well, one of the problems is that MI is detecting a limb movement, not really a deep emotion, a workaround for this, can detect the emotion from the facial expression detected from the EEG signals, but that will be prediction over prediction so it might suffer from low accuracy. But definitely will be consider as a step forward as currently the emotion recognitions has to be done in a closed controlled environment and can't be used outside the labs, this will create a leap forward as it will allow the subjects to smile or frown on an emotional stimuli, which is considered as a step outside the lab.
- It's too early for this suggestion but can we incorporate MI and emotion recognition to allow subjects to move, but filter out this movement signals and focus on the emotional signals? Just like what we are doing now with eye blinking artifacts? The importance of this incorporation is to allow the subjects to use it outside the labs and controlled environments, which is a step forward to the day to day usages of such technology, and it will be convenient for real-life applications.
- The field is lacking from a full, big, detailed dataset to be used as a benchmark for the emotion recognition studies, a reference to measure the accuracy of different work.

7.4. Contributions

- The first contribution of this work is publishing the paper EEG Emotion Detection Review on the conference IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)
- The second contribution is publishing the paper EEG Emotion Detection Using Multi-Model Classification on BSB International Conference on Bioinformatics and Systems Biology
- Publishing the paper "Novel Model Architecture for EEG Emotion Classification" in IJBTT 2019 – International Journal of Biotech Trends and Technology
- Publishing the paper "Novel Theoretical EEG Machine Learning Model for Emotion Recognition" in: IJCTT 2019 – International Journal of Computer Trends and Technology
- In this work, we introduced a novel model of detecting the EEG emotions, with a clarification on the data filtering and classification, which resulted in higher accuracy than the state-of-the-art researches in the field of emotion recognition.
- Alongside the model, there is the data that has been collected, which considered as an asset for future researches
- As well as the software that has been developed to collect and manipulate the data before and after the classification, tagging the data and manipulate it is a lot of work.

7.5. Summary

In this chapter, we mentioned the conclusion as well as the list the contribution of this study.

REFERENCES

JOSEPH A. MIKELS, et al., 2005. Emotional category data on images from the International Affective Picture System. Behavior Research Methods.

M. A. B. S. Akhanda & S. M. F. Islam and M. M. Rahman, 2014. Detection of Cognitive State for Brain-Computer Interfaces. International Conference on Electrical Information and Communication Technology (EICT).

P. Ackermann, C. Kohlschein, J. Á. Bitsch & K. Wehrle and S. Jeschke, 2016. EEG-based automatic emotion recognition: Feature extraction selection and classification methods. 18th International Conference on e-Health Networking, Applications and Services (Healthcom).

Mohamed Ahmed Abdullah & Lars Rune Christensen, 2019. Novel Model Architecturefor EEG Emotion Classification. International Journal of Biotech Trends and Technology (IJBTT), 9(3), pp. 1-5.

Lesley Ann Axelrod, 2009. Emotional recognition in computing. s.l.:School of Information Systems, Computing and Mathematics Brunel University.

Anon., 2002. Emotion Recognition from Physiological Signal Analysis: A Review Electronic Notes in Theoretical Computer Science vol. Volume 343 pp. Pages 35-55 2019.

Anon., 2002. S. Koelstra DEAP: A database for emotion analysis; Using physiological signals IEEE Transactions on Affective Computing vol. Volume: 3 no. Issue: 1 pp. 18 - 31 2012.

Anon., 2014. http://bnci-horizon-2020.eu/database/data-sets.
[Accessed 12 March 2020].

Anon., 2014. https://www.ant-neuro.com/show-case/relation-between-self-paced-exercise-performance-and-eeg-activity-influence-hot-and.
[Accessed 12 March 2020].

B. B. Kemp, et al., 2000. Analysis of a Sleep-Dependent Neuronal Feedback Loop : The Slow-Wave Micro-continuity of the EEG. vol. 47, no. 9, pp. 1185–1194.

A. Barachant, 2013. Classification of covariance matrices using a Riemannian-based kernel for BCI applications To cite this version : HAL Id : hal-00820475 Classification of covariance matrices using a Riemannian-based kernel for BCI applications.

A. Barachant, et al., 2011. Channel Selection Procedure using Riemannian distance for BCI applications To cite this version : Channel Selection Procedure using Riemannian distance for BCI applications.

Byun, S. W., Lee, S. P. & Han, H. S., 2017. Feature Selection and Comparison for the Emotion Recognition According to Music Listening. International Conference on Robotics and Automation Sciences (ICRAS),.

Esmeralda C. Djamal & Poppi Lodaya, 2017. EEG based emotion monitoring using wavelet and learning vector quantization. 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI).

P. C. Petrantonakis and L. J. Hadjileontiadis, 2012. Adaptive Emotional Information Retrieval From EEG Signals in the Time-Frequency Domain. IEEE Transactions on Signal Processing.

G. Cheng, 2017. A closed-loop Brain-Computer Music Interface for continuous affective interaction. no. December, pp. 176–179.

M. Congedo, A. Barachant & and A. Andreev, 2013. A New Generation of Brain-Computer Interface Based on Riemannian Geometry. arXiv Prepr. arXiv1310.8115, vol. 33, no. 0.

Delorme, et al., 2004 . Interaction of Bottom-up and Top-down processing in the fast visual analysis of natural scenes. Cognitive Brain Research, 103-113.

Matthieu Duvinage, Thierry Castermans, Mathieu Petieau, Thomas Hoellinger, Guy Cheron, Thierry Dutoit, 2013. Performance of the Emotiv Epoc headset for P300-based applications. BioMedical Engineering Online, 12.

A. F. Rabbi, et al., 2009. Human performance evaluation based on EEG signal analysis: A prospective review. Annual International Conference of the IEEE Engineering in Medicine and Biology Society.

S. G. Mangalagowri and P. C. P. Raj, 2016 . EEG feature extraction and classification using feed forward backpropogation algorithm for emotion detection. International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT).

Y. Gao & H. J. Lee and R. M. Mehmood, 2015. Deep learninig of EEG signals for emotion recognition. IEEE International Conference on Multimedia and Expo Workshops (ICMEW).

Elise S. Dan-Glauser & Klaus R. Scherer, 2011. The Geneva affective picture database (GAPED): a new 730-picture database focusing on valence and normative significance. Behavior Research Methods.

B. H. Kim and S. Jo, 2018 . Deep Physiological Affect Network for the Recognition of Human Emotions. IEEE Transactions on Affective Computing.

S. H. Kim and N. A. N. Thi, 2016. Feature extraction of emotional states for EEG-based rage control. 39th International Conference on Telecommunications and Signal Processing (TSP).

Y. H. Liu, W. T. Cheng, Y. T. Hsiao & C. T. Wu and M. D. Jeng, 2014. EEG-based emotion recognition based on kernel Fisher's discriminant analysis and spectral powers. IEEE International Conference on Systems, Man, and Cybernetics (SMC).

Andreas Haag, Silke Goronzy, Peter Schaich & Jason Williams, 2004. Emotion recognition using bio-sensors: First steps towards an automatic system. Affective Dialogue Systems, pp. pages 36-48.

D. Handayani, H. Yaacob & A. Wahab and I. F. T. Alshaikli, 2015. Statistical Approach for a Complex Emotion Recognition Based on EEG Features. 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT).

A. Harati, et al., 2013. The Temple University Hospital EEG Corpus. pp. 29–32.

Ulrich Hoffmann, Jean-Marc Vesin, Touradj Ebrahimi, Karin Diserens, 2008. An efficient P300-based brain-computer interface for disabled subjects. Journal of Neuroscience Methods, 167(1), pp. 115-125.

D. Huang, C. Guan, Kai Keng Ang & Haihong Zhang and Yaozhang Pan, 2012. Asymmetric Spatial Pattern for EEG-based emotion detection. International Joint Conference on Neural Networks (IJCNN).

S. I. Alzahrani, 2016. P300 Wave Detection Using Emotiv Epoc+ Headset: Effects of Matrix Size Flash Duration and Colors. p. 76f.

B. J. B. B. & . H. B. Edelman, 2016. EEG Source Imaging Enhances the Decoding of Complex Right-Hand Motor Imagery Tasks. IEEE Transactions on Biomedical Engineering , vol. Volume: 63, no. Issue: 1.

K. J. Miller, et al., 2016. Spontaneous Decoding of the Timing and Content of Human Object Perception from Cortical Surface Recordings Reveals Complementary Information in the Event-Related Potential and Broadband Spectral Change. PLoS Comput. Biol., vol. 12, no. 1, pp. 1–20.

Z. J. Koles, M. S. Lazar & and S. Z. Zhou, 1990. Spatial patterns underlying population differences in the background EEG. Brain Topogr., vol. 2, no. 4, pp. 275–284.

R. Jenke & A. Peer and M. Buss, 2014. Feature Extraction and Selection for Emotion Recognition from EEG. IEEE Transactions on Affective Computing.

S. Jirayucharoensak, S. Pan-Ngum & and P. Israsena, 2014. EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. Sci. World J., vol. 2014.

Sunil Kalagi, et al., 2017. Brain computer interface systems using non-invasive electroencephalogram signal : A literature review. International conference on engineering technology and innovation (ICE/ITMC).

J. Kaur and A. Kaur, 2015. A review on analysis of EEG signals. International Conference on Advances in Computer Engineering and Applications.

Minho Kim, Byung Hyung Kim, and Sungho Jo, 2015. Quantitative Evaluation of a Low-Cost Noninvasive Hybrid Interface Based on EEG and Eye Movement. s.l., IEEE Transactions on Neural Systems and Rehabilitation Engineering.

Koelstra, S. et al., 2012 . DEAP : A Database for Emotion Analysis Using Physiological Signals. IEEE Transactions on Affective Computing .

Benedek Kurdi, Shayn Lozano & Mahzarin R. Banaji, 2017. Introducing the Open Affective Standardized Image Set (OASIS). Behavior research methods.

W. L. Zheng, et al., 2018. EmotionMeter: A Multimodal Framework for Recognizing Human Emotions. IEEE Trans. Cybern., pp. 1–13.

S. Lemm, B. Blankertz, G. Curio & and K. R. Müller, 2005. Spatio-spectral filters for improving the classification of single trial EEG. IEEE Trans. Biomed. Eng., vol. 52, no. 9, pp. 1541–1548.

X. Li, 2016. Emotion Recognition from Multi-Channel EEG Data through Convolutional Recurrent Neural Network. 2016 Ieee Int. Conf. Bioinforma. Biomed., pp. 352–359.

Ou Lin, Guang-Yuan Liu & Jie-Min Yang and Yang-Ze Du, 2015. Neurophysiological markers of identifying regret by 64 channels EEG signal. 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).

Yun Luo, Li-Zhen Zhu & Bao-Liang Lu, 2019. A GAN-Based Data Augmentation Method for Multimodal Emotion Recognition. Advances in Neural Networks – ISNN 2019 , pp. pages 141-150.

M. M. Bradley and P. J. Lang, 1994. Measuring emotion: The self-assessment manikin and the semantic differential. J. Behav. Ther. Exp. Psychiatry, vol. 25, no. 1, pp. 49–59.

Margaret M.Bradley & Peter J.Lang, 2002. Measuring emotion: The self-assessment manikin and the semantic differential. Journal of Behavior Therapy and Experimental Psychiatry.

Choubeila Maaoui & Alain Pruski, 2010. Emotion recognition through physiological signals for human-machine communication. Cutting Edge Robotics.

Artur Marchewka, Łukasz Żurawski, Katarzyna Jednoróg & Anna Grabowska, 2014. The Nencki Affective Picture System (NAPS): Introduction to a novel standardized wide-range high-quality realistic picture database. Behavior Research Methods.

Arturo Nakasone, Helmut Prendinger, Mitsuru Ishizuka & Mitsuru Ishizuka, 2005. Emotion Recognition from Electromyography and Skin Conductance. in Proceedings 5th International Workshop on Biosignal Interpretation (BSI-05).

Y. P. Lin and T. P. Jung, 2014. Exploring day-to-day variability in EEG-based emotion classification. IEEE International Conference on Systems, Man, and Cybernetics (SMC).

A. Patil & C. Deshmukh and A. R. Panat, 2016. Feature extraction of EEG for emotion recognition using Hjorth features and higher order crossings. Conference on Advances in Signal Processing (CASP).

Pan Peining, Gary Tan, Aung Aung & Aung aung Phyo wai, 2018. Evaluation of Consumer-Grade EEG Headsets for BCI Drone Control. International Society of Neurofeedback and Research (ISNR), vol. Vol 5, no. Issue 2.

Picard & R. W., 1995. Affective Computing. M.I.T Media Laboratory Perceptual Computing Section Technical Report, no. No 321.

H. Xu and K. N. (Kostas) Plataniotis, 2012. Affect recognition using EEG signal. IEEE 14th Int. Work. Multimed. Signal Process., pp. 299–304.

R. Q. Quiroga, A. Kraskov, T. Kreuz & and P. Grassberger, 2002. Performance of different synchronization measures in real data: A case study on electroencephalographic signals. vol. 65, pp. 1–14.

L. R. C. Mohamed Ahmed Abdullah, 2019. A Review of EEG Emotion Recognition. International Journal of Computer Trends and Technology (IJCTT), 67(6), pp. 41-47.

L. R. Christensen & M. A. Abdullah, 2018. EEG Emotion Detection Review. 2018 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol., pp. 1–7.

M. Rajya Lakshmi, Dr. T. V. Prasad & Dr. V. Chandra Prakash, 2014. Survey on EEG Signal Processing Methods. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE).

B. Rivet, A. Souloumiac, V. Attina & and G. Gibert, 2009. xDAWN algorithm to enhance evoked potentials: application to brain computer interface. Biomed Eng, IEEE Trans, vol. 56, pp. 1–9.

B. Rivet, et al., 2011. Theoretical analysis of XDAWN algorithm: Application to an efficient sensor selection in a P300 BCI BT - 19th European Signal Processing Conference EUSIPCO 2011 August 29 2011 - September 2 2011. Eur. Signal Process. Conf., no. Eusipco, pp. 1382–1386.

V. S. Kota, 2017. A Survey on Various Motor Imagery-Based Brain-Computer Interface Techniques. vol. Volume 3, no. Issue 5.

A. Saidatul, M. P. Paulraj & S. Yaacob and N. F. Mohamad Nasir, 2011. Automated System for Stress Evaluation Based on EEG Signal: A Prospective Review. IEEE 7th International Colloquium on Signal Processing and its Applications.

A. Samara & M. L. R. Menezes and L. Galway, 2016. Feature Extraction for Emotion Recognition and Modelling Using Neurophysiological Data. 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS).

Saeid Sanei, J.A. Chambers, 2007. EEG SIGNAL PROCESSING. s.l.:John Wiley & Sons.

Scherer & Klaus R. (2005)., 2005. What are emotions? and how can they be measured?. Social Science Information, 695-729.

Schneider, et al., 1997. Functional MRI reveals left amygdala activation during emotion. Psychiatry Research: Neuroimaging, vol. Volume 76, no. Issues 2–3, pp. Pages 75-82.

M. Tangermann, 2012. Review of the BCI competition IV. vol. 6, no. July, pp. 1–31.

Thomas & P., K., 2017. EEG-based motor imagery classification using subject-specific spatio-spectral features.. 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017, (pp. 2302-2307).

S. Vaid & P. Singh and C. Kaur, 2015. EEG Signal Analysis for BCI Interface: A Review. Fifth International Conference on Advanced Computing and Communication Technologies.

Chung-Ping Young ; Chao-Hsien Hsieh ; Hsu-Chuan Wang, 2009. A low-cost real-time closed-loop epileptic seizure monitor and controller. in IEEE Instrumentation and Measurement Technology Conference.

I. Wichakam and P. Vateekul, 2014. An evaluation of feature extraction in EEG-based emotion prediction with support vector machines. 11th International Joint Conference on Computer Science and Software Engineering (JCSSE).

Anon., n.d. F1 score. [Online]
Available at: https://en.wikipedia.org/wiki/F1_score
[Accessed 12 March 2020].

Anon., n.d. Harmonic mean. [Online]
Available at: https://en.wikipedia.org/wiki/Harmonic_mean
[Accessed 121 March 2020].

K. Yano and T. Suyama, 2016. Fixed low-rank EEG spatial filter estimation for emotion recognition induced by movies. International Workshop on Pattern Recognition in Neuroimaging (PRNI).

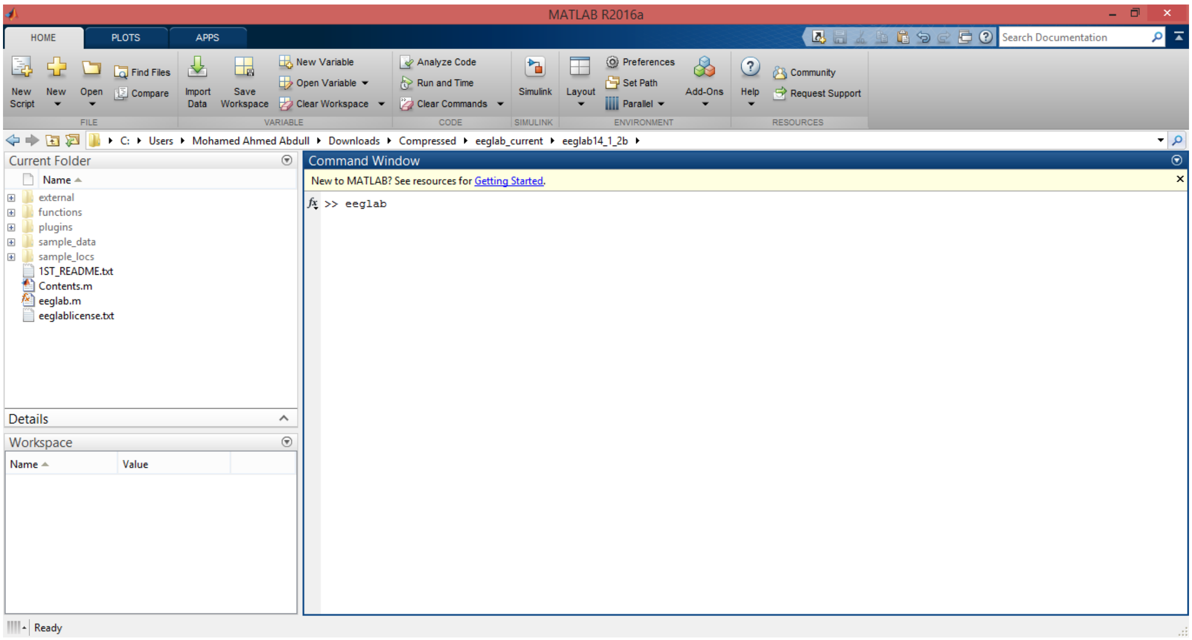
H. Zhang, 2017. A feasibility study of detecting brain signal in EEG during emotional self-regulation. pp. 184–187.

Qibin Zhao, Liqing Zhang & Andrzej Cichocki, 2009. EEG-based asynchronous BCI control of a car in 3D virtual reality environments. Chinese Science Bulletin, pp. pages 78-87.

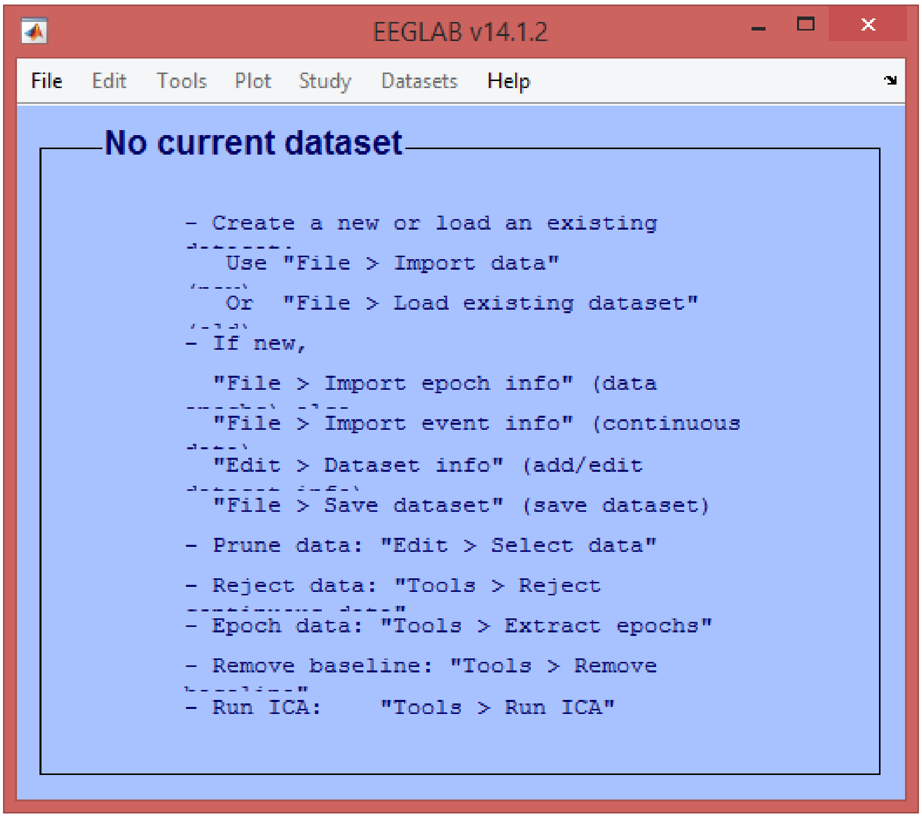
Zheng, A., 2015. Evaluating Machine Learning Models. s.l.:O'Reilly Media.

X. Zhuang & V. Rozgić and M. Crystal, 2014. Compact unsupervised EEG response representation for emotion recognition. International Conference on Biomedical and Health Informatics (BHI).

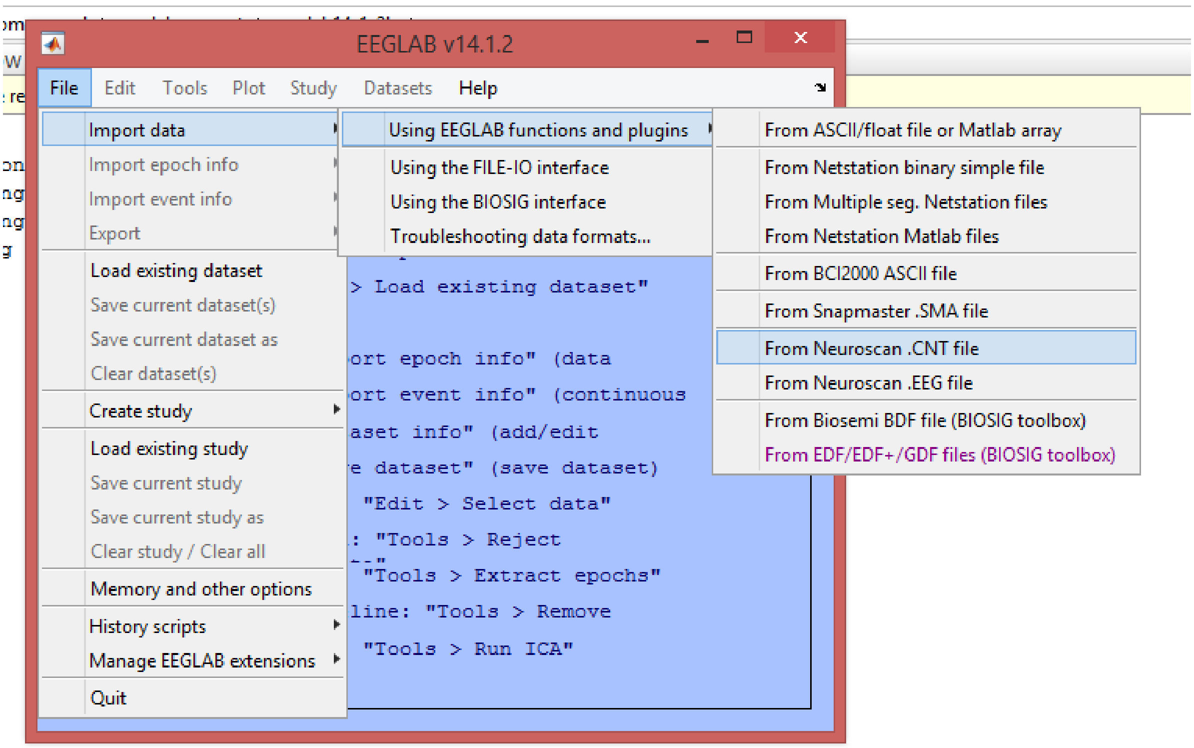
APPENDIX A: Using EEG Lab module to visualize, Tag and convert the signal to text format.



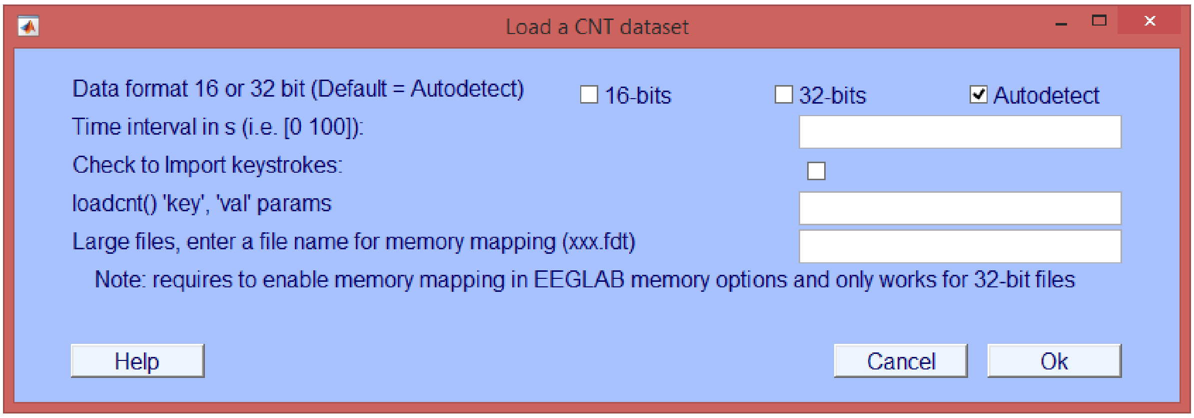
Appendix 1: EEG Lab in matlab



Appendix 2: EEGLAB main screen

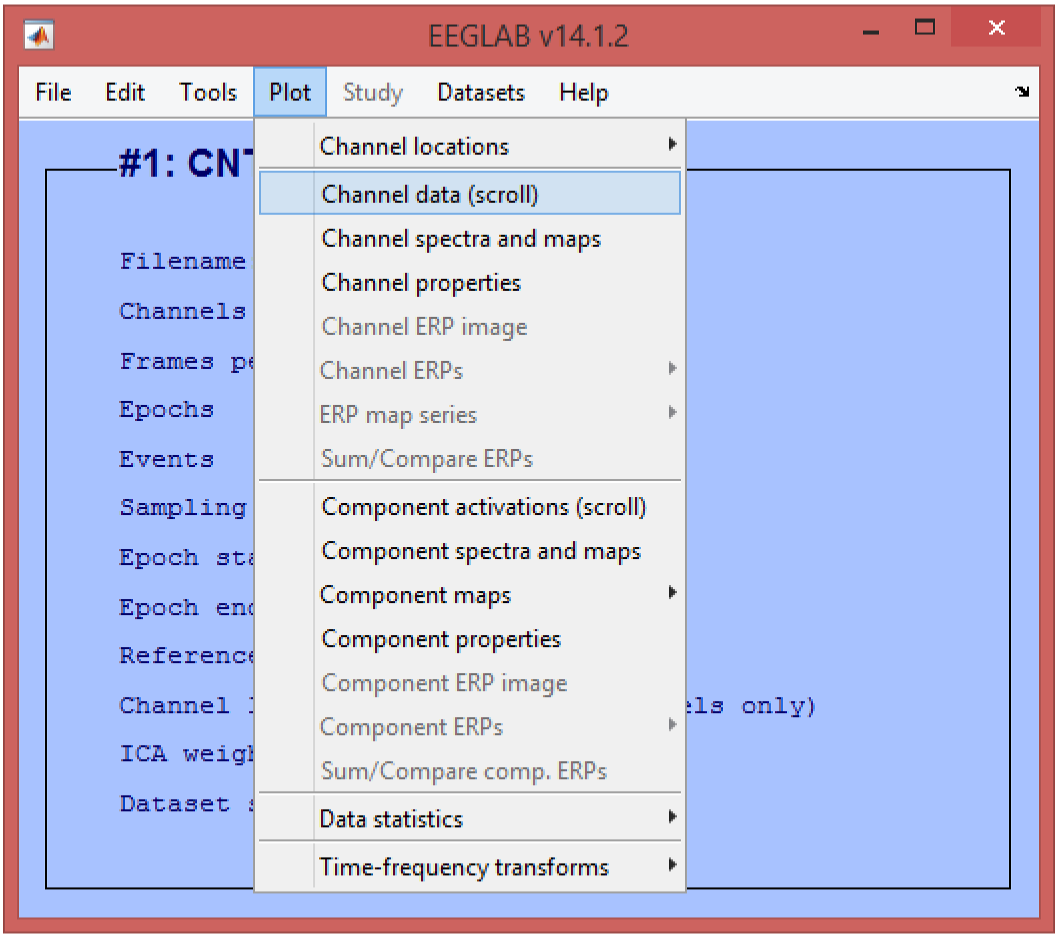


Appendix 3: Open .cnt file format

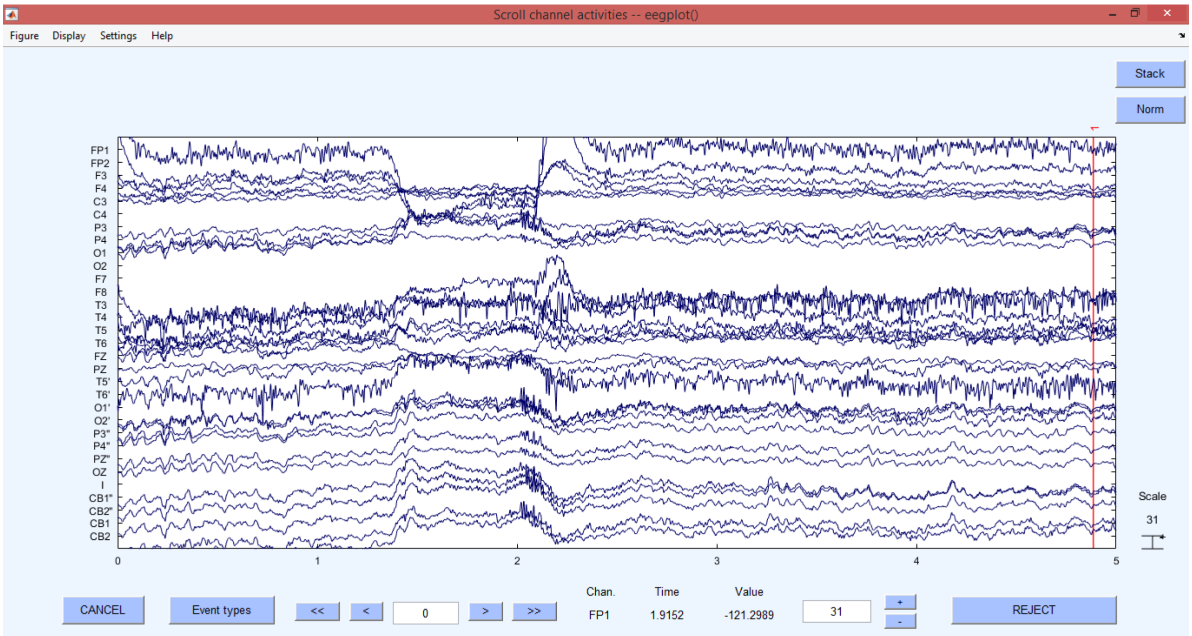


Appendix 4: Using the default parameters for the dataset

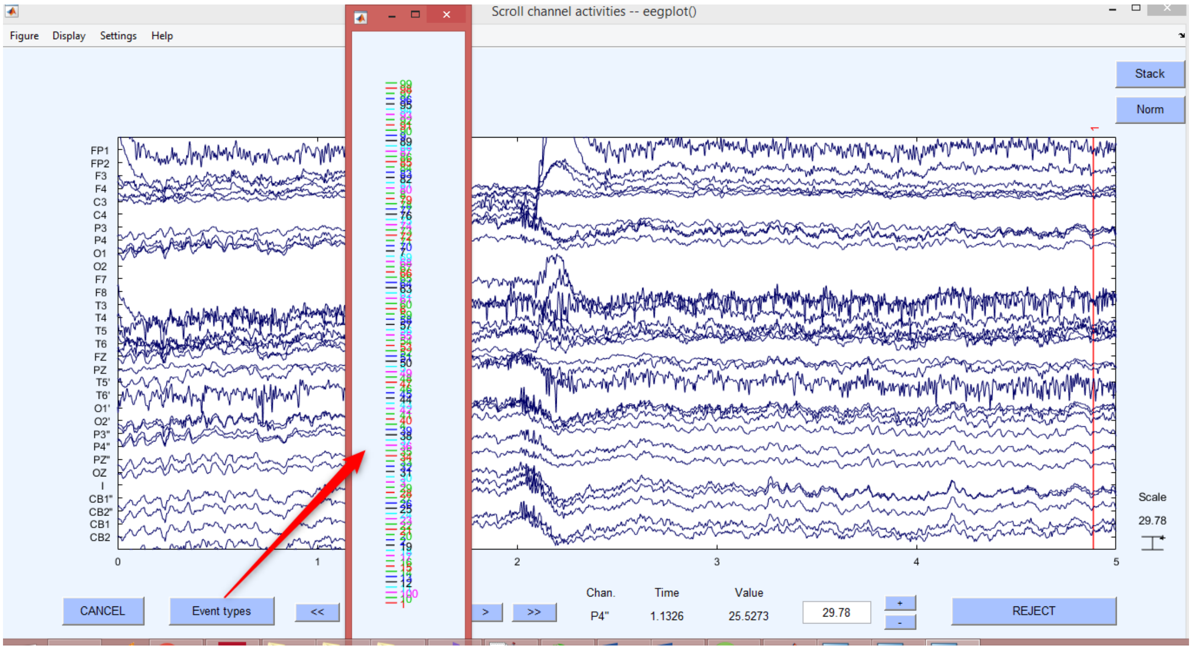
Appendix 5: Choosing the dataset format



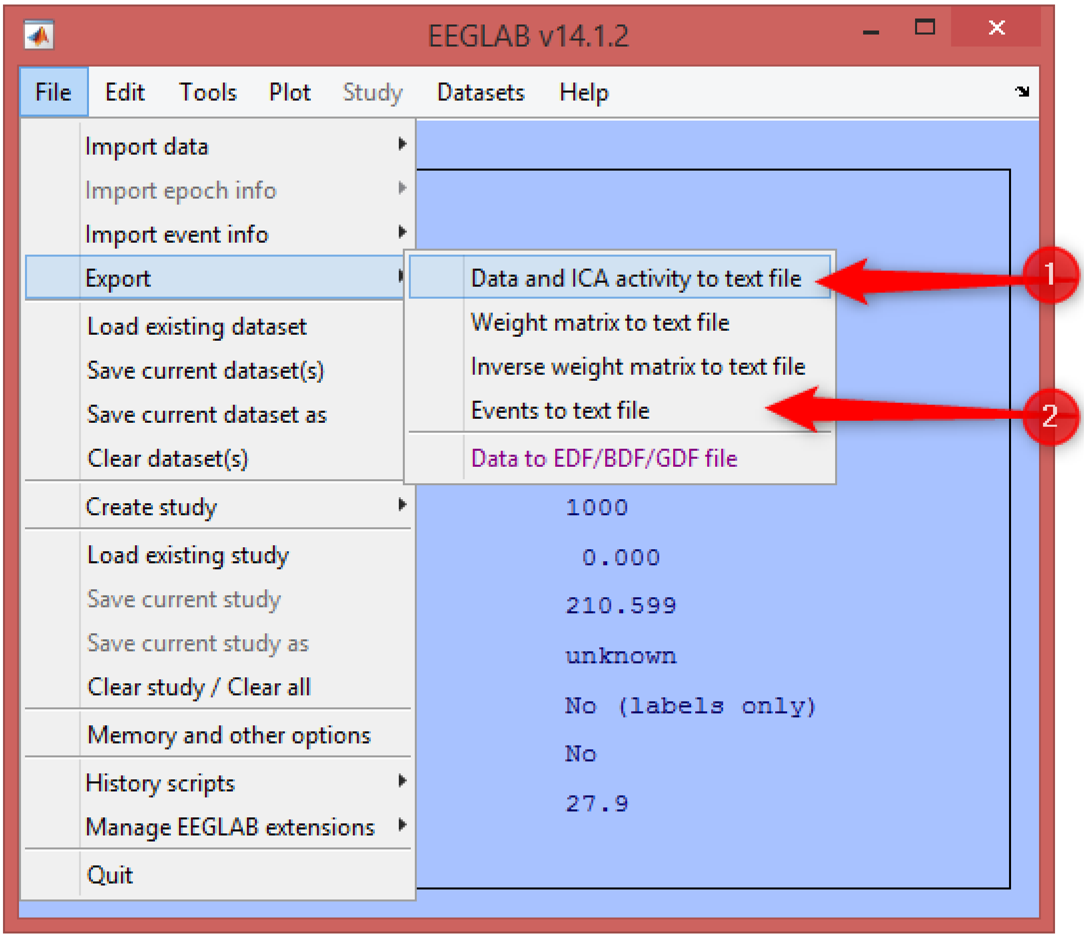
Appendix 6: Menu option to open the graph



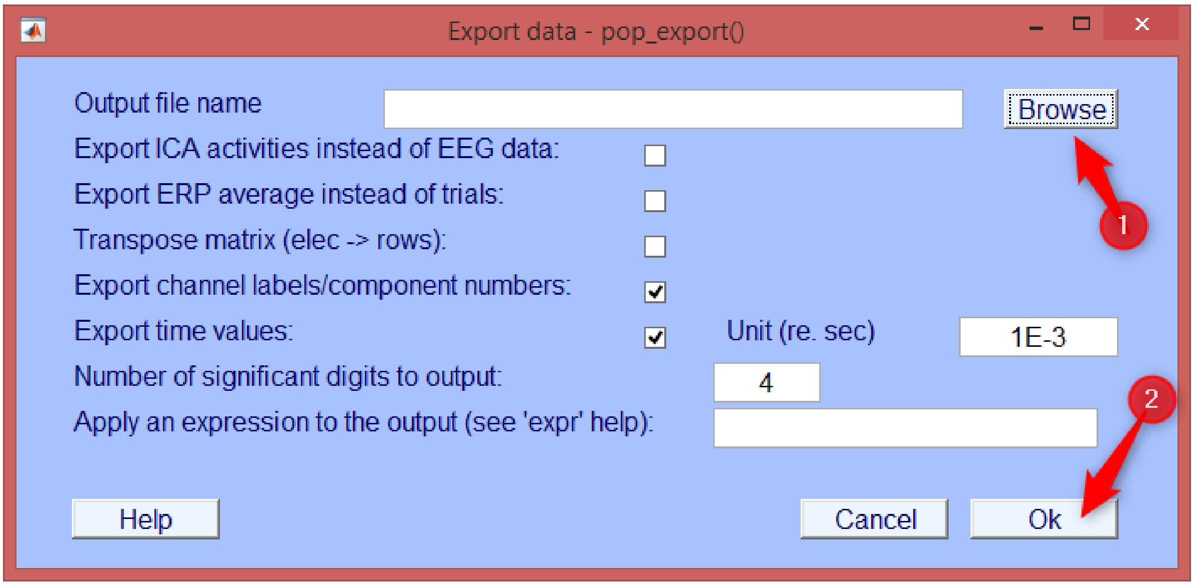
Appendix 7: EEG data graph



Appendix 8: Events on the EEG graph

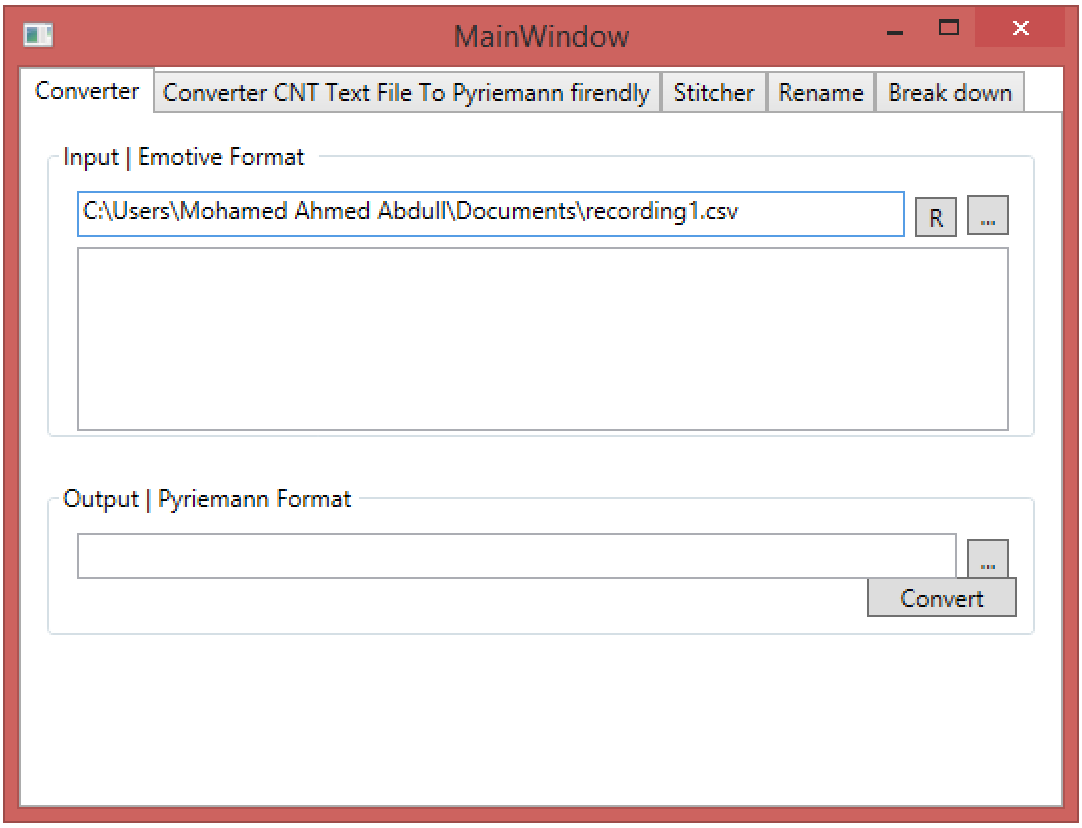


Appendix 9: Menu options to export the events

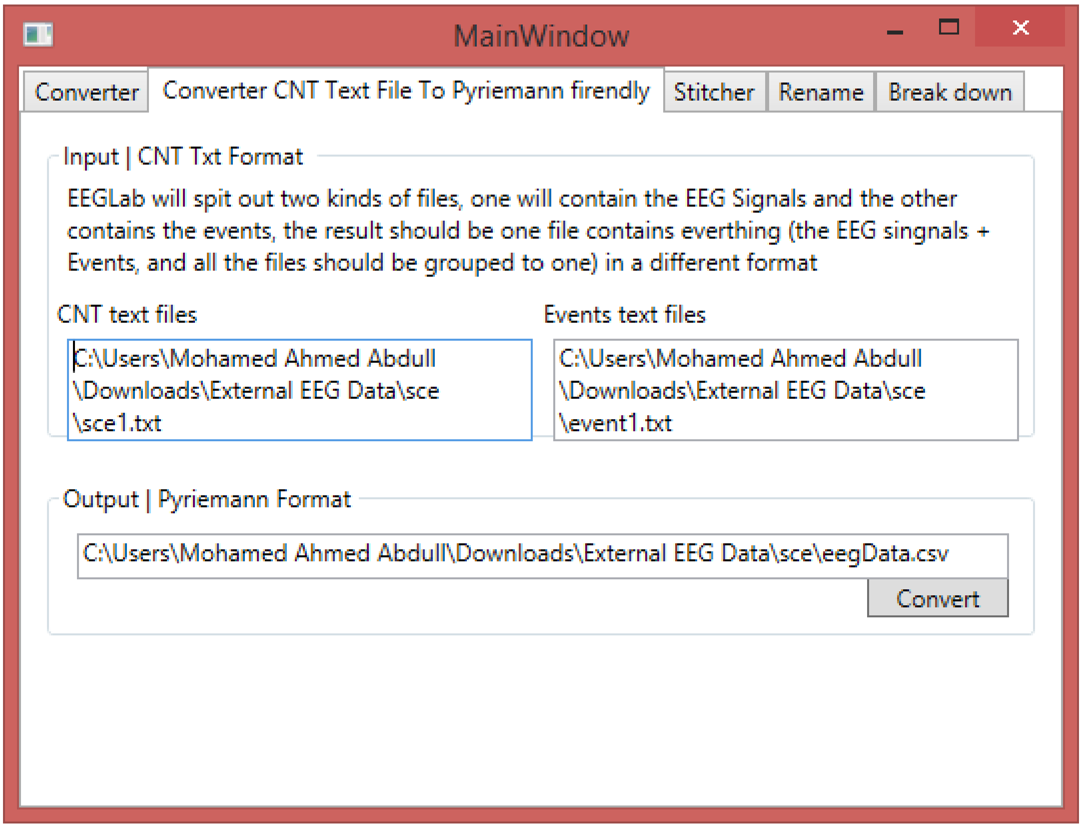


Appendix 10: Exporting the events data

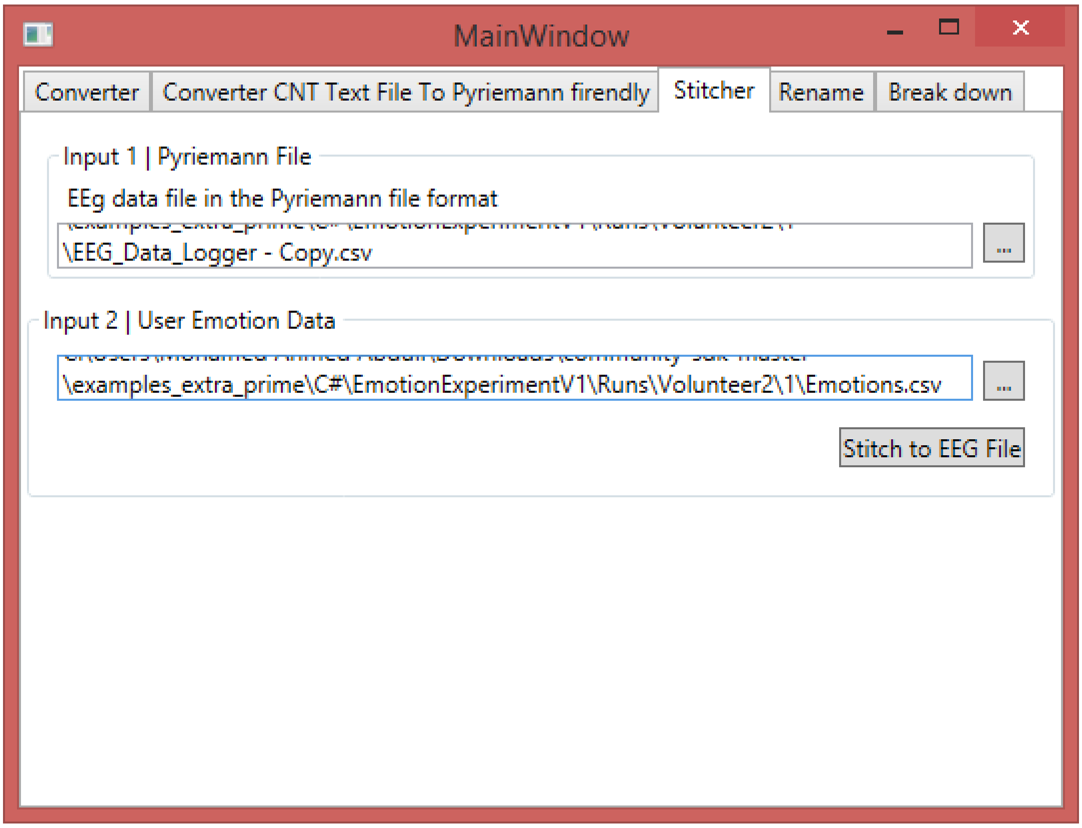
APPENDIX B: Application to convert the EEG signals to different formats.



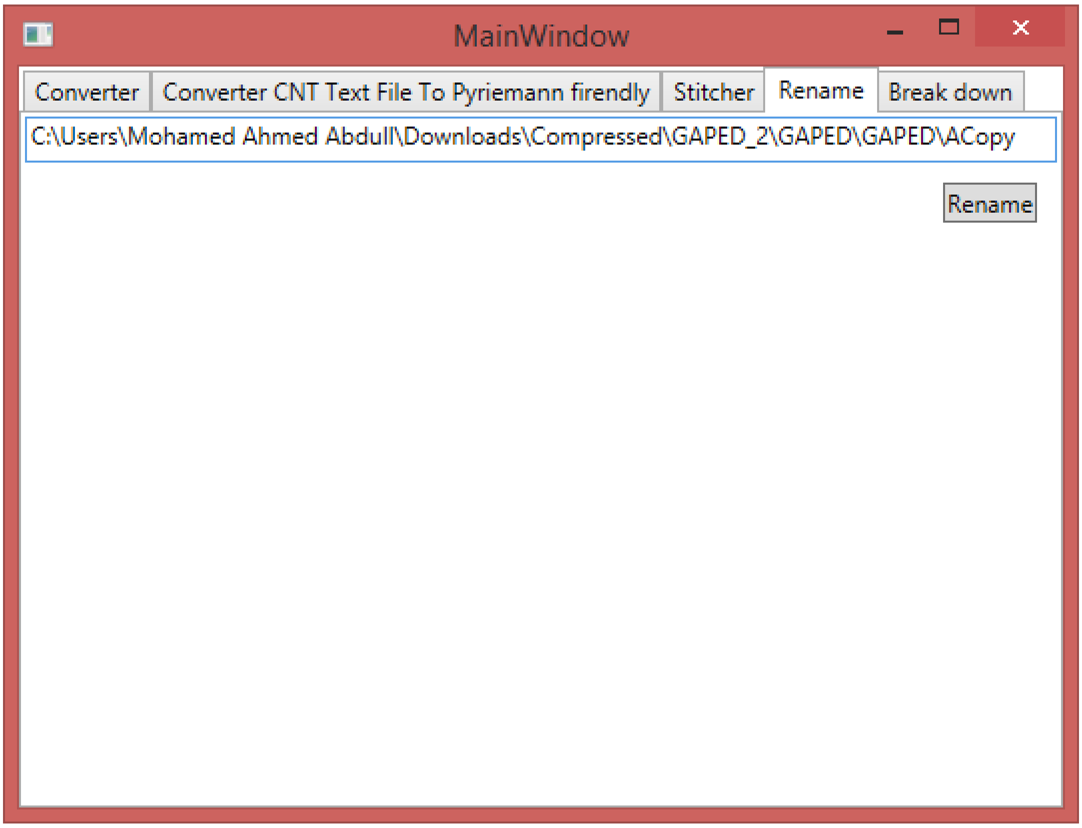
Appendix 11: Converting the data from the EEG device format to the ML classifier format



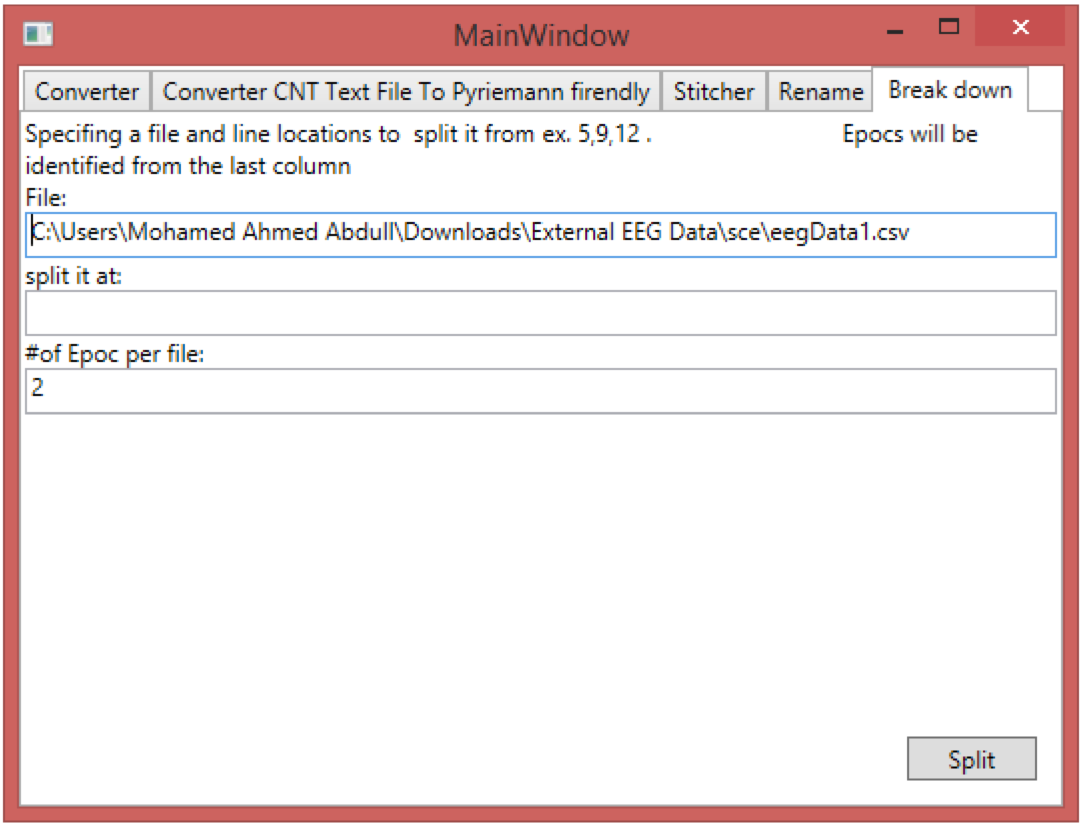
Appendix 12: Converting EEGLAB format to ML classifier format



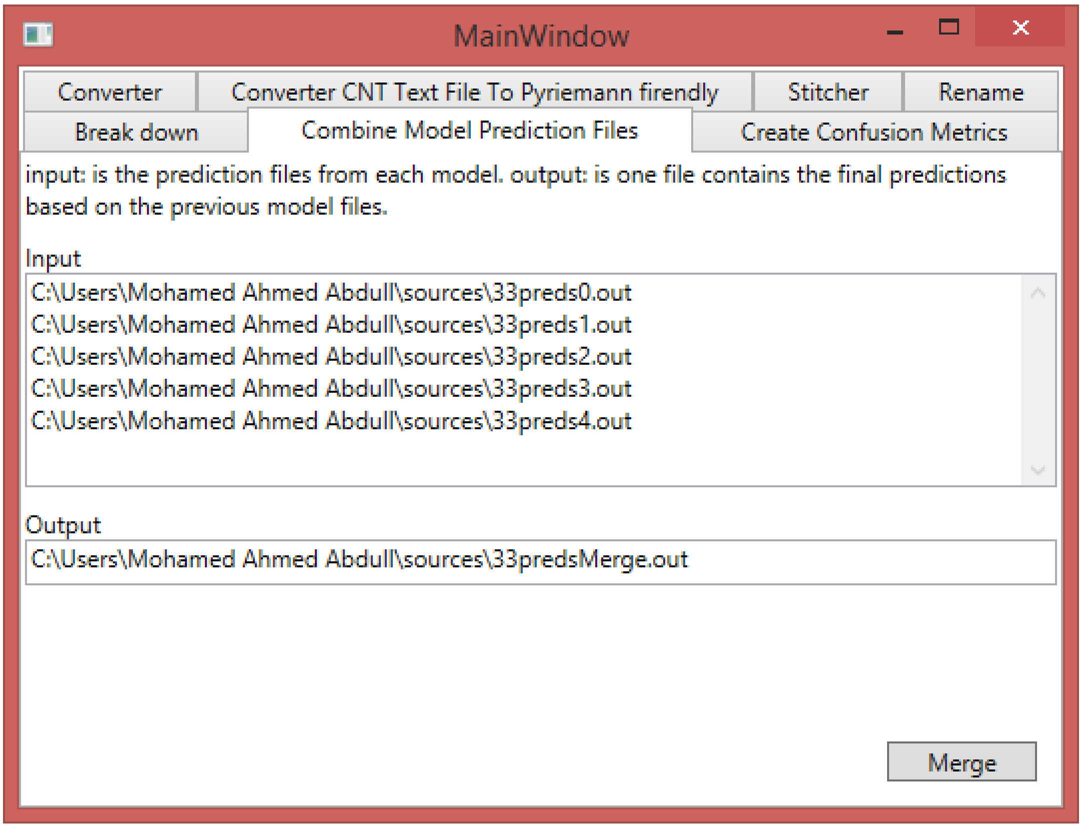
Appendix 13: Add the user surveys (SAM) selection the EEG dataset



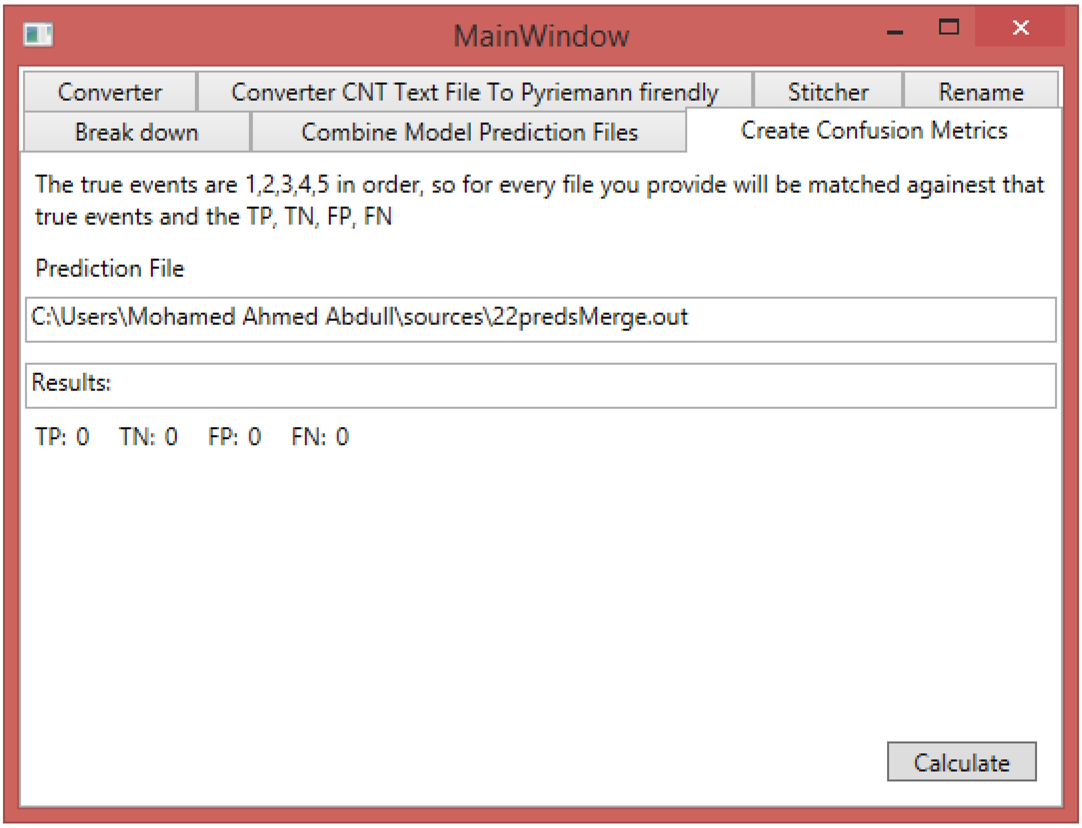
Appendix 14: Rename all the files inside the folder



Appendix 15: Break down the EEG dataset file to small parts to feed it to the ML classifier



Appendix 16: Combining the Model Prediction results



Appendix 17: Create the confusion metrics from the ML prediction files

APPENDIX C: APPLICATION to conduct the SAM experiment and collect the raw EEG data and prepare the data for Tagging.

Appendix 18: Application that collect the EEG data whole showing the images

Code:

```
<Window x:Class="EmotionExperimentV1.Collecter"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:EmotionExperimentV1"
             mc:Ignorable="d"  WindowState="Maximized" WindowStyle="None"
             d:DesignHeight="450" d:DesignWidth="800">
    <Grid Background="Black">
        <Image Name="Image" Source="{Binding DisplayedImage}" />
        <TextBlock Name="CountDownTextBlock" Text="5" HorizontalAlignment="Center"
VerticalAlignment="Center" Foreground="White"
                   FontSize="162"/>
        <Button Name="StartButton" Content="Start" HorizontalAlignment="Center"
VerticalAlignment="Center" FontSize="36"
                Margin="0,228,0,0" Padding="20,1" Click="StartButton_Click" />
    </Grid>
</Window>
```

Appendix 19: Collecter.xaml

```
using EmotionExperimentV1.Collector;
using System.ComponentModel;
using System.Threading;
using System.Windows;

namespace EmotionExperimentV1
{
    public partial class Collecter : Window, INotifyPropertyChanged
    {
        public float TimeBetweenPics = 2.5f; //in seconds
        public int CountDownCounter = 5;
        public string ImagesFolder = System.Environment.CurrentDirectory +
@"\Images";
        //public string imagesFormat = "png";

        private string _displayedImage;
        public string DisplayedImage
```

```
        {
            get { return _displayedImage; }
            set { _displayedImage = value; OnPropertyChanged("DisplayedImage"); }
        }
        public IEegLogger _eeg;

        #region NotifyProperty
        public event PropertyChangedEventHandler PropertyChanged;
        protected void OnPropertyChanged(string name)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(name));
        }
        #endregion

        public Collecter()
        {
            InitializeComponent();

            DataContext = this;

            _eeg = new EEG_Logger(TimeBetweenPics);
            //_eeg = new FakeEegLogger(TimeBetweenPics);


#if FakeEEG
            _eeg = new FakeEegLogger(TimeBetweenPics);
#endif
#if RealEEG
            _eeg = new EEG_Logger(TimeBetweenPics);

#endif
        }

        private void StartButton_Click(object sender, RoutedEventArgs e)
        {
            CountDown();
        }

        public void CountDown()
        {
            new Thread(new Counter().Start).Start(this);
        }

        public void Start()
        {
            CountDownTextBlock.Visibility = Visibility.Collapsed;
            StartButton.Visibility = Visibility.Collapsed;

            new Thread(new ImageRotation().Start).Start(this);
        }
    }
}
```

Appendix 20: Collecter.xaml.cs

```
using System;
using System.Threading;

namespace EmotionExperimentV1.Collector
{
    public class Counter
    {
        public void Start(object obj)
        {
```

```
                Collecter mainWindow = obj as Collecter;
                do
                {

                    System.Windows.Application.Current.Dispatcher.Invoke(new Action(() =>
                    {
                        mainWindow.CountDownTextBlock.Text = mainWindow.CountDownCounter
+ "";
                    }));

                    Thread.Sleep(1000);
                    mainWindow.CountDownCounter--;
                }
                while (mainWindow.CountDownCounter > 0);

                System.Windows.Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    mainWindow.Start();
                }));
            }
        }
}
```

Appendix 21: Counter.cs

```
using Emotiv;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Windows;

namespace EmotionExperimentV1.Collector
{
    public class EEG_Logger : IEegLogger
    {
        EmoEngine engine;    // Access to the EDK is via the EmoEngine
        int userID = -1;     // userID is used to uniquely identify a user's headset
        string filename = @"C:\Users\Mohamed Ahmed Abdull\Downloads\community-sdk-
master\examples_extra_prime\C#\EmotionExperimentV1\EEG_Data_Logger.csv"; // output
filename
        static string userName = "mohamedahmed";
        static string password = "Moha12!@";
        static string licenseKey = "cb08855f-14bc-4c91-b49c-e6744d8ca7e4";// Your
License Key
        float _bufferInSeconds = 0;

        public EEG_Logger(float bufferInSeconds)
        {
            _bufferInSeconds = bufferInSeconds + 1; //+1 is a safe margine

            // create the engine
            engine = EmoEngine.Instance;
            engine.UserAdded += new
EmoEngine.UserAddedEventHandler(engine_UserAdded_Event);

            // connect to Emoengine.
            engine.Connect();

            if (EmotivCloudClient.EC_Connect() != EdkDll.EDK_OK)
            {
                MessageBox.Show("Cannot connect to Emotiv Cloud.");
                return;
            }
```

```
            if (EmotivCloudClient.EC_Login(userName, password) != EdkDll.EDK_OK)
            {
                MessageBox.Show("Your login attempt has failed. The username or
password may be incorrect");
                return;
            }

            //Active license
            activateLicense();

            //Create a header for our output file
            WriteHeader();
        }

        static void activateLicense()
        {
            int result = EdkDll.IEE_AuthorizeLicense(licenseKey, 1);
            if (result == EdkDll.EDK_OK || result == EdkDll.EDK_LICENSE_REGISTERED)
            {
                //Console.WriteLine("Active/Debit successfully.");
            }
            else
            {
             //   MessageBox.Show("Active/Debit unsuccessfully. Errorcode : " +
result);
            }
        }

        void engine_UserAdded_Event(object sender, EmoEngineEventArgs e)
        {
            Debug.WriteLine("User Added Event has occured");

            // record the user
            userID = (int)e.userId;

            // enable data aquisition for this user.
            engine.DataAcquisitionEnable((uint)userID, true);

            // ask for up to 1 second of buffered data
            engine.DataSetBufferSizeInSec(_bufferInSeconds);
        }

        /// <summary>
        /// Collect EEG Data From the Buffer and save it to file
        /// </summary>
        public void Collect(int Stimulus_Type, int Stimulus_ID)
        {
            // Handle any waiting events
            engine.ProcessEvents();

            // If the user has not yet connected, do not proceed
            if ((int)userID == -1)
                return;

            Dictionary<EdkDll.IEE_DataChannel_t, double[]> data =
engine.GetData((uint)userID);

            if (data == null)
                return;

            int _bufferSize = data[EdkDll.IEE_DataChannel_t.IED_TIMESTAMP].Length;

            Debug.WriteLine("Writing " + _bufferSize.ToString() + " sample of data
");
```

```
            // Write the data to a file
            TextWriter file = new StreamWriter(filename, true);

            for (int i = 0; i < _bufferSize; i++)
            {
                // now write the data
                file.Write(userID + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_AF3][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_F7][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_F3][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_FC5][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_T7][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_P7][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_O1][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_O2][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_P8][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_T8][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_FC6][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_F4][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_F8][i] + ",");
                file.Write(data[EdkDll.IEE_DataChannel_t.IED_AF4][i] + ",");
                file.Write(Stimulus_Type + ",");
                file.Write(Stimulus_ID + "");

                file.WriteLine("");

            }
            file.Close();
        }

        public void WriteHeader()
        {
            TextWriter file = new StreamWriter(filename, false);

            //string header = "COUNTER, INTERPOLATED, RAW_CQ, AF3, F7, F3, FC5, T7,
P7, O1, O2, P8," +
            //    "T8, FC6, F4, F8, AF4, GYROX, GYROY, TIMESTAMP, MARKER_HARDWARE,
ES_TIMESTAMP, FUNC_ID, FUNC_VALUE, MARKER, SYNC_SIGNAL";

            string header =
@"""PatientID"",""Electrode_1"",""Electrode_2"",""Electrode_3"",""Electrode_4"",""Ele
ctrode_5"",""Electrode_6"",""Electrode_7"",""Electrode_8"",""Electrode_9"",""Electrod
e_10"",""Electrode_11"",""Electrode_12"",""Electrode_13"",""Electrode_14"",""Stimulus
_Type"",""Stimulus_ID""";

            file.WriteLine(header);
            file.Close();
        }
    }
}
```

Appendix 22: EegLogger.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EmotionExperimentV1.Collector
{
    public class FakeEegLogger : IEegLogger
    {
        public FakeEegLogger(float bufferInSeconds) { }
```

```
        public void Collect(int Stimulus_Type, int Stimulus_ID) { }
    }
}
```

Appendix 23: FakeEegLogger.cs

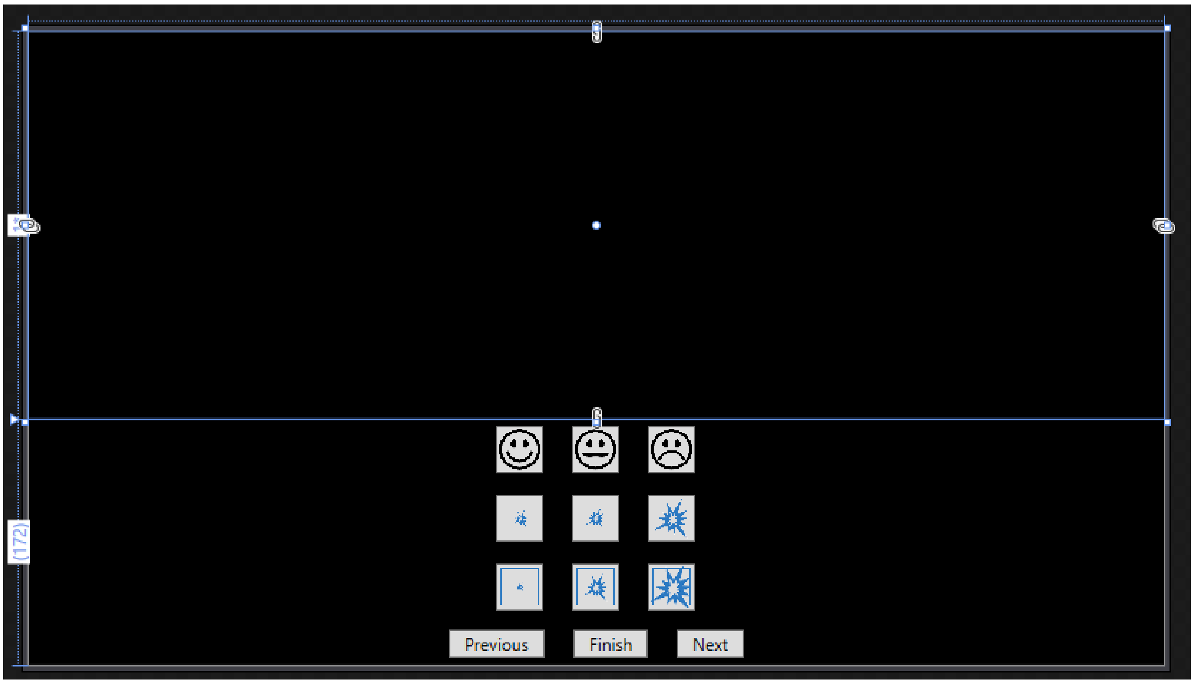
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace EmotionExperimentV1.Collector
{
    public interface IEegLogger
    {
        void Collect(int Stimulus_Type, int Stimulus_ID);
    }
}
```

Appendix 24: IEegLogger.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading;
using System.Windows;

namespace EmotionExperimentV1.Collector
{
    public class ImageRotation
    {
        public void Start(object obj)
        {
            Collecter mainWindow = obj as Collecter;
            var files = Directory.GetFiles(mainWindow.ImagesFolder).ToList();

            files = Shuffle(files);

            foreach (var imageFile in files)
            {
                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    mainWindow.DisplayedImage = imageFile;
                }));

                var imageName =
int.Parse(Path.GetFileNameWithoutExtension(imageFile));

                mainWindow._eeg.Collect(imageName, imageName);

                Thread.Sleep((int)(mainWindow.TimeBetweenPics * 1000));
            }

            Application.Current.Dispatcher.Invoke(new Action(() =>
            {
                mainWindow.CountDownTextBlock.Visibility = Visibility.Visible;
                mainWindow.CountDownTextBlock.Text = "The End";
            }));
        }

        private List<string> Shuffle(List<string> files)
```

```
        {
            Random rnd = new Random();
            return files.OrderBy(o => rnd.Next()).Select(s => s).ToList();
        }
    }
}
```

Appendix 25: ImageRotation.cs



Appendix 26: Application to collect the SAM survey data

```
<Window x:Class="EmotionExperimentV1.EmotionCollector.EmotionCollector"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:EmotionExperimentV1.EmotionCollector"
             mc:Ignorable="d"  WindowState="Maximized" WindowStyle="None"
             d:DesignHeight="450" d:DesignWidth="800">
    <Grid Background="Black">
        <Grid.RowDefinitions>
            <RowDefinition Height="1*"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>
        <Image Name="Image" Source="{Binding DisplayedImage}" Grid.Row="0" />

        <Grid Name="Controls" Grid.Row="2">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="33"/>
            </Grid.RowDefinitions>

            <StackPanel Orientation="Horizontal"
                        HorizontalAlignment="Center"
                        Grid.Row="0">
                <ToggleButton Name="smile1" Width="33" Height="33" Margin="10,5"
```

```
                                 Checked="ToggleButton_Checked"   >
                            <Image Source="../Resources/smile1.png" />
                        </ToggleButton>
                        <ToggleButton Name="smile2" Width="33" Height="33" Margin="10,5"
                                 Checked="ToggleButton_Checked">
                            <Image Source="../Resources/smile2.png" />
                        </ToggleButton>
                        <ToggleButton Name="smile3" Width="33" Height="33" Margin="10,5"
                                 Checked="ToggleButton_Checked">
                            <Image Source="../Resources/smile3.png" />
                        </ToggleButton>
                    </StackPanel>

                    <StackPanel Orientation="Horizontal"
                                HorizontalAlignment="Center"
                                Grid.Row="1" >
                        <ToggleButton Name="valence1" Checked="ToggleButton_Checked"
                                 Width="33" Height="33" Margin="10">
                            <Image Source="../Resources/expo1.png" />
                        </ToggleButton>
                        <ToggleButton Name="valence2" Checked="ToggleButton_Checked"
                                  Width="33" Height="33" Margin="10,5" >
                            <Image Source="../Resources/expo2.png" />
                        </ToggleButton>
                        <ToggleButton Name="valence3" Checked="ToggleButton_Checked"
                                  Width="33" Height="33" Margin="10,5" >
                            <Image Source="../Resources/expo3.png" />
                        </ToggleButton>
                    </StackPanel>

                    <StackPanel Orientation="Horizontal"
                                HorizontalAlignment="Center"
                                Grid.Row="2" >
                        <ToggleButton  Name="dominance1" Checked="ToggleButton_Checked"
                                  Width="33" Height="33" Margin="10,5" >
                            <Image Source="../Resources/dom1.png" />
                        </ToggleButton>
                        <ToggleButton Name="dominance2" Checked="ToggleButton_Checked"
                                 Width="33" Height="33" Margin="10,5" >
                            <Image Source="../Resources/dom2.png" />
                        </ToggleButton>
                        <ToggleButton Name="dominance3" Checked="ToggleButton_Checked"
                                 Width="33" Height="33" Margin="10,5" >
                            <Image Source="../Resources/dom3.png" />
                        </ToggleButton>
                    </StackPanel>

                    <StackPanel Orientation="Horizontal"
                                HorizontalAlignment="Center"
                                Grid.Row="3" >

                        <Button Content="Previous" Name="Previous" Margin="10,5"
                                HorizontalAlignment="Center" VerticalAlignment="Bottom"
Padding="10,1" Click="Previous_Click"/>

                        <Button Content="Finish" Name="GetFile" Margin="10,5"
                            HorizontalAlignment="Center" VerticalAlignment="Bottom"
Padding="10,1" Click="GetFile_Click"/>

                        <Button Content="Next" Name="Next" Margin="10,5"
                                HorizontalAlignment="Center" VerticalAlignment="Bottom"
Padding="10,1" Click="Next_Click"/>

                    </StackPanel>
```

```
        </Grid>
    </Grid>
</Window>
```

Appendix 27: EmotionCollector.xaml

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Text.RegularExpressions;
using System.Windows;
using System.Windows.Controls.Primitives;

namespace EmotionExperimentV1.EmotionCollector
{
    public partial class EmotionCollector : Window, INotifyPropertyChanged
    {
        public string ImagesFolder = System.Environment.CurrentDirectory +
@"\Images";
        //public string imagesFormat = "png";
        public string EmotionsFile = "Emotions.csv";
        private string _displayedImage;
        public string DisplayedImage
        {
            get { return _displayedImage; }
            set { _displayedImage = value; OnPropertyChanged("DisplayedImage"); }
        }

        #region NotifyProperty
        public event PropertyChangedEventHandler PropertyChanged;
        protected void OnPropertyChanged(string name)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(name));
        }
        #endregion

        public List<Sam> Sams { get; set; } = new List<Sam>();

        int CurrentImageIndex = 0;

        public EmotionCollector()
        {
            InitializeComponent();

            DataContext = this;

            foreach (var imageFile in Directory.GetFiles(ImagesFolder))
            {
                Sams.Add(new Sam { ImageUrl = imageFile });
            }

            ShowImage();
        }

        private void Previous_Click(object sender, RoutedEventArgs e)
        {
            if (CurrentImageIndex > 0)
            {
                CurrentImageIndex--;
                ShowImage();
                UnToggleAll();
            }
```

```
        }

        private void Next_Click(object sender, RoutedEventArgs e)
        {
            if (CurrentImageIndex < Sams.Count - 1)
            {
                CurrentImageIndex++;
                ShowImage();
                UnToggleAll();
            }
        }

        public void ShowImage()
        {
            DisplayedImage = CurrentSam().ImageUrl;
        }

        public Sam CurrentSam()
        { return Sams[CurrentImageIndex]; }

        private void ToggleButton_Checked(object sender, RoutedEventArgs e)
        {
            var button = sender as ToggleButton;
            if (button.Name.Contains("smile"))
            {
                UnToggleSmile(button);
                UpdateSamWithValance(button);
            }
            else if (button.Name.Contains("valence"))
            {
                UnToggleValence(button);
                UpdateSamWithArousal(button);

            }
            else if (button.Name.Contains("dominance"))
            {
                UnToggleDominance(button);
                UpdateSamWithDominance(button);
            }
        }

        private void UpdateSamWithDominance(ToggleButton button)
        {
            var Dominance = (Dominance)int.Parse(Regex.Match(button.Name,
@"\d").Value);
            CurrentSam().Dominance = Dominance;
        }

        private void UpdateSamWithArousal(ToggleButton button)
        {
            var Arousal = (Arousal)int.Parse(Regex.Match(button.Name, @"\d").Value);
            CurrentSam().Arousal = Arousal;
        }

        private void UpdateSamWithValance(ToggleButton button)
        {
            var valance = (Valance)int.Parse(Regex.Match(button.Name, @"\d").Value);
            CurrentSam().Valance = valance;
        }

        public void UnToggleSmile(ToggleButton but)
        {
            if (but != smile1) smile1.IsChecked = false;
            if (but != smile2) smile2.IsChecked = false;
            if (but != smile3) smile3.IsChecked = false;
```

```
                //if (but != smile4) smile4.IsChecked = false;
                //if (but != smile5) smile5.IsChecked = false;
            }

            public void UnToggleValence(ToggleButton but)
            {
                if (but != valence1) valence1.IsChecked = false;
                if (but != valence2) valence2.IsChecked = false;
                if (but != valence3) valence3.IsChecked = false;
                //if (but != valence4) valence4.IsChecked = false;
                //if (but != valence5) valence5.IsChecked = false;
            }

            public void UnToggleDominance(ToggleButton but)
            {
                if (but != dominance1) dominance1.IsChecked = false;
                if (but != dominance2) dominance2.IsChecked = false;
                if (but != dominance3) dominance3.IsChecked = false;
                //if (but != dominance4) dominance4.IsChecked = false;
                //if (but != dominance5) dominance5.IsChecked = false;
            }

            public void UnToggleAll()
            {
                smile1.IsChecked = false;
                smile2.IsChecked = false;
                smile3.IsChecked = false;
                //smile4.IsChecked = false;
                //smile5.IsChecked = false;

                valence1.IsChecked = false;
                valence2.IsChecked = false;
                valence3.IsChecked = false;
                //valence4.IsChecked = false;
                //valence5.IsChecked = false;

                dominance1.IsChecked = false;
                dominance2.IsChecked = false;
                dominance3.IsChecked = false;
                //dominance4.IsChecked = false;
                //dominance5.IsChecked = false;
            }

            private void GetFile_Click(object sender, RoutedEventArgs e)
            {
                //Print out the Sams
                File.Delete(EmotionsFile);
                foreach (var sam in Sams)
                {
                    File.AppendAllText(EmotionsFile,
Path.GetFileNameWithoutExtension(sam.ImageUrl)+",
"+GetEmotionIndex(sam)+Environment.NewLine);
                }
            }

            private string GetEmotionIndex(Sam sam)
            {
                return (((int)sam.Valance) + ((int)sam.Arousal * 10) +
((int)sam.Dominance * 100)) + "";
            }
        }

        public class Sam
        {
            public string ImageUrl { get; set; }
```

```
        public Valance Valance { get; set; }
        public Arousal Arousal { get; set; }
        public Dominance Dominance { get; set; }
    }

    public enum Valance
    {
        None,
        Happy,
        Meh,
        Sad,
    }

    public enum Arousal
    {
        None,
        Relaxed,
        Meh,
        Calm,
    }

    public enum Dominance
    {
        None,
        Controlled,
        Meh,
        Controlling,
    }
}
```

Appendix 28: EmotionCollector.xaml.cs

Appendix 29: Application selector

```
<Window x:Class="EmotionExperimentV1.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:EmotionExperimentV1"
        mc:Ignorable="d"
```

```
        Title="MainWindow" Height="450" Width="800">
    <Grid Background="Black">

        <Button Content="Collector" HorizontalAlignment="Center"
VerticalAlignment="Center"
                Padding="10,1" Margin="-100,0,0,0" Click="Button_Click"  />

        <Button Content="Tester" HorizontalAlignment="Center"
VerticalAlignment="Center"
                Padding="10,1" Margin="100,0,0,0" Click="Button_Click_1"  />

    </Grid>
</Window>
```

Appendix 30: MainWindow.xaml for Application selector

```
using EmotionExperimentV1.Collector;
using Emotiv;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Diagnostics;
using System.IO;
using System.Threading;
using System.Windows;

namespace EmotionExperimentV1
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            new Collecter().Show();
        }

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            new EmotionCollector.EmotionCollector().Show();
        }
    }
}
```

Appendix 31: MainWindow.xaml.cs for application selector

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" DefaultTargets="Build"
xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">x86</Platform>
    <ProductVersion>8.0.30703</ProductVersion>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{C915CADD-223F-474D-AA51-821F4E476316}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>EmotionExperimentV1</RootNamespace>
```

```xml
    <AssemblyName>EmotionExperimentV1</AssemblyName>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
    <TargetFrameworkProfile>Client</TargetFrameworkProfile>
    <FileAlignment>512</FileAlignment>
    <IsWebBootstrapper>false</IsWebBootstrapper>
    <PublishUrl>publish\</PublishUrl>
    <Install>true</Install>
    <InstallFrom>Disk</InstallFrom>
    <UpdateEnabled>false</UpdateEnabled>
    <UpdateMode>Foreground</UpdateMode>
    <UpdateInterval>7</UpdateInterval>
    <UpdateIntervalUnits>Days</UpdateIntervalUnits>
    <UpdatePeriodically>false</UpdatePeriodically>
    <UpdateRequired>false</UpdateRequired>
    <MapFileExtensions>true</MapFileExtensions>
    <ApplicationRevision>0</ApplicationRevision>
    <ApplicationVersion>1.0.0.%2a</ApplicationVersion>
    <UseApplicationTrust>false</UseApplicationTrust>
    <BootstrapperEnabled>true</BootstrapperEnabled>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|x86' ">
    <PlatformTarget>x86</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <OutputPath>..\..\..\bin\win32\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|x86' ">
    <PlatformTarget>x86</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <OutputPath>..\..\..\bin\win32\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x64'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>..\..\..\bin\win64\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x64</PlatformTarget>

<CodeAnalysisLogFile>bin\Debug\MotionDataLogger.exe.CodeAnalysisLog.xml</CodeA
nalysisLogFile>

<CodeAnalysisUseTypeNameInSuppression>true</CodeAnalysisUseTypeNameInSuppress
ion>
```

```
<CodeAnalysisModuleSuppressionsFile>GlobalSuppressions.cs</CodeAnalysisModuleSup
pressionsFile>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisRuleSetDirectories>;C:\Program Files (x86)\Microsoft Visual Studio
10.0\Team Tools\Static Analysis Tools\\Rule Sets</CodeAnalysisRuleSetDirectories>
    <CodeAnalysisIgnoreBuiltInRuleSets>true</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisRuleDirectories>;C:\Program Files (x86)\Microsoft Visual Studio
10.0\Team Tools\Static Analysis Tools\FxCop\\Rules</CodeAnalysisRuleDirectories>
    <CodeAnalysisIgnoreBuiltInRules>true</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x64'">
    <OutputPath>..\..\..\bin\win64\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <Optimize>true</Optimize>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x64</PlatformTarget>

<CodeAnalysisLogFile>bin\Release\MotionDataLogger.exe.CodeAnalysisLog.xml</Code
AnalysisLogFile>

<CodeAnalysisUseTypeNameInSuppression>true</CodeAnalysisUseTypeNameInSuppress
ion>

<CodeAnalysisModuleSuppressionsFile>GlobalSuppressions.cs</CodeAnalysisModuleSup
pressionsFile>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisRuleSetDirectories>;C:\Program Files (x86)\Microsoft Visual Studio
10.0\Team Tools\Static Analysis Tools\\Rule Sets</CodeAnalysisRuleSetDirectories>
    <CodeAnalysisIgnoreBuiltInRuleSets>true</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisRuleDirectories>;C:\Program Files (x86)\Microsoft Visual Studio
10.0\Team Tools\Static Analysis Tools\FxCop\\Rules</CodeAnalysisRuleDirectories>
    <CodeAnalysisIgnoreBuiltInRules>true</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'FakeEEG|x86'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x86\FakeEEG\</OutputPath>
    <DefineConstants>TRACE;DEBUG;FakeEEG</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'FakeEEG|x64'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x64\FakeEEG\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x64</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
```

```
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'RealEEG|x86'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x86\RealEEG\</OutputPath>
    <DefineConstants>TRACE;DEBUG;RealEEG</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'RealEEG|x64'">
    <DebugSymbols>true</DebugSymbols>
    <OutputPath>bin\x64\RealEEG\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x64</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>MinimumRecommendedRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>
  <ItemGroup>
    <BootstrapperPackage Include=".NETFramework,Version=v4.0,Profile=Client">
      <Visible>False</Visible>
      <ProductName>Microsoft .NET Framework 4 Client Profile %28x86 and
x64%29</ProductName>
      <Install>true</Install>
    </BootstrapperPackage>
    <BootstrapperPackage Include="Microsoft.Net.Client.3.5">
      <Visible>False</Visible>
      <ProductName>.NET Framework 3.5 SP1 Client Profile</ProductName>
      <Install>false</Install>
    </BootstrapperPackage>
    <BootstrapperPackage Include="Microsoft.Net.Framework.3.5.SP1">
      <Visible>False</Visible>
      <ProductName>.NET Framework 3.5 SP1</ProductName>
      <Install>false</Install>
    </BootstrapperPackage>
    <BootstrapperPackage Include="Microsoft.Windows.Installer.3.1">
      <Visible>False</Visible>
      <ProductName>Windows Installer 3.1</ProductName>
      <Install>true</Install>
    </BootstrapperPackage>
  </ItemGroup>
  <ItemGroup>
    <ProjectReference Include="..\DotNetEmotivSDK\DotNetEmotivSDK.csproj">
      <Project>{d3337309-9682-425e-8edc-2ecfacf79565}</Project>
      <Name>DotNetEmotivSDK</Name>
    </ProjectReference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
    <Reference Include="Microsoft.CSharp" />
```

```
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="System.Net.Http" />
    <Reference Include="System.Xaml">
      <RequiredTargetFramework>4.0</RequiredTargetFramework>
    </Reference>
    <Reference Include="WindowsBase" />
    <Reference Include="PresentationCore" />
    <Reference Include="PresentationFramework" />
  </ItemGroup>
  <ItemGroup>
    <ApplicationDefinition Include="App.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </ApplicationDefinition>
    <Page Include="EegCollector\Collecter.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
    <Page Include="EmotionCollector\EmotionCollector.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
    <Page Include="MainWindow.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Compile Include="App.xaml.cs">
      <DependentUpon>App.xaml</DependentUpon>
      <SubType>Code</SubType>
    </Compile>
    <Compile Include="EegCollector\Collecter.xaml.cs">
      <DependentUpon>Collecter.xaml</DependentUpon>
    </Compile>
    <Compile Include="EegCollector\Counter.cs" />
    <Compile Include="EegCollector\EegLogger.cs" />
    <Compile Include="EegCollector\FakeEegLogger.cs" />
    <Compile Include="EegCollector\IEegLogger.cs" />
    <Compile Include="EegCollector\ImageRotation.cs" />
    <Compile Include="EmotionCollector\EmotionCollector.xaml.cs">
      <DependentUpon>EmotionCollector.xaml</DependentUpon>
    </Compile>
    <Compile Include="MainWindow.xaml.cs">
      <DependentUpon>MainWindow.xaml</DependentUpon>
      <SubType>Code</SubType>
    </Compile>
  </ItemGroup>
  <ItemGroup>
    <Compile Include="Properties\AssemblyInfo.cs">
      <SubType>Code</SubType>
    </Compile>
```

```
    <Compile Include="Properties\Resources.Designer.cs">
      <AutoGen>True</AutoGen>
      <DesignTime>True</DesignTime>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="Properties\Settings.Designer.cs">
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
    <EmbeddedResource Include="Properties\Resources.resx">
      <Generator>ResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.cs</LastGenOutput>
    </EmbeddedResource>
    <None Include="Properties\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <LastGenOutput>Settings.Designer.cs</LastGenOutput>
    </None>
  </ItemGroup>
  <ItemGroup>
    <None Include="App.config" />
  </ItemGroup>
  <ItemGroup>
    <Content Include="Images\1.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\10.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\11.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\12.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\13.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\14.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\15.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\16.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\17.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\18.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
```

```
</Content>
<Content Include="Images\19.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\2.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\20.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\21.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\22.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\23.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\24.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\25.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\26.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\27.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\28.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\29.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\3.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\30.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\4.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\41.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\42.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
```

```
<Content Include="Images\43.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\44.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\45.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\46.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\47.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\48.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\49.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\5.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\50.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\51.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\52.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\53.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\54.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\55.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\56.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\57.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\58.bmp">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<Content Include="Images\59.bmp">
```

```
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\6.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\60.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\7.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\8.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Images\9.bmp">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\dom1.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\dom2.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\dom3.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\expo1.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\expo2.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\expo3.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\smile1.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\smile2.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="Resources\smile3.png">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
  </ItemGroup>
  <ItemGroup />
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
</Project>
```

Appendix 32: EmotionExperimentV1.csproj

```
import numpy as np
from copy import deepcopy
from sklearn.base import BaseEstimator, TransformerMixin


def epoch_data(data, events, stim_ID, tmin=-.2, tmax=0.399):
    """Epoch data."""
    ix_events = np.where(np.diff(np.int32(events > 0)) == 1)[0] + 1

    ix_min = int(tmin*1000)
    ix_max = int(tmax*1000)
    nsamp = ix_max - ix_min
    X = np.zeros((len(ix_events), data.shape[0], nsamp))
    y = np.int32(events[ix_events] > 50)
    st_id = np.int32(stim_ID[ix_events])

    for i, ix in enumerate(ix_events):
        sl = slice((ix + ix_min), (ix + ix_max))
        tmp = data[:, sl]
        X[i, :, 0:tmp.shape[1]] = tmp
    return X, y, st_id


class DownSampler(BaseEstimator, TransformerMixin):
    """Downsample transformer"""

    def __init__(self, factor=4):
        """Init."""
        self.factor = factor

    def fit(self, X, y):
        return self

    def transform(self, X):
        return X[:, :, ::self.factor]


class EpochsVectorizer(BaseEstimator, TransformerMixin):
    """Vectorize epochs."""

    def __init__(self):
        """Init."""

    def fit(self, X, y):
        return self

    def transform(self, X):
        X2 = np.array([x.flatten() for x in X])
        return X2


class CospBoostingClassifier(BaseEstimator, TransformerMixin):
```

```
    """Cospectral matrice bagging."""

    def __init__(self, baseclf):
        """Init."""
        self.baseclf = baseclf

    def fit(self, X, y):
        self.clfs_ = []
        for i in range(X.shape[-1]):
            clf = deepcopy(self.baseclf)
            self.clfs_.append(clf.fit(X[:, :, :, i], y))
        return self

    def predict_proba(self, X):
        proba = []
        for i in range(X.shape[-1]):
            proba.append(self.clfs_[i].predict_proba(X[:, :, :, i]))
        proba = np.mean(proba, axis=0)
        return proba

    def transform(self, X):
        proba = []
        for i in range(X.shape[-1]):
            proba.append(self.clfs_[i].predict_proba(X[:, :, :, i]))
        proba = np.concatenate(proba, 1)
        return proba
```

Appendix 33: utils.py

```
%reset  -f
import numpy as np
import pandas as pd
import joblib
from collections import OrderedDict
from copy import deepcopy

from pyriemann.estimation import (XdawnCovariances, HankelCovariances,
                          CospCovariances, ERPCovariances)
from pyriemann.spatialfilters import Xdawn, CSP
from pyriemann.tangentspace import TangentSpace
from pyriemann.channelselection import ElectrodeSelection

from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.cross_validation import KFold
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import cross_val_score
from scipy import stats

import os
import sys
module_path = os.path.abspath(os.path.join('..'))
```

```
if module_path not in sys.path:
    sys.path.append(module_path)

from utils import (DownSampler, EpochsVectorizer, CospBoostingClassifier, epoch_data)

pipeline = OrderedDict()

# ERPs models
pipeline['XdawnCov'] = make_pipeline(XdawnCovariances(6, estimator='oas'),
                                     TangentSpace('riemann'),
                                     LogisticRegression('l2'))

pipeline['Xdawn'] = make_pipeline(Xdawn(12, estimator='oas'),
                                  DownSampler(5),
                                  EpochsVectorizer(),
                                  LogisticRegression('l2'))

# Induced activity models

baseclf = make_pipeline(ElectrodeSelection(10, metric=dict(mean='logeuclid',
                                                           distance='riemann')),
                        TangentSpace('riemann'),
                        LogisticRegression('l1'))

pipeline['Cosp'] = make_pipeline(CospCovariances(fs=1000, window=32,
                                                 overlap=0.95, fmax=300,
                                                 fmin=1),
                                 CospBoostingClassifier(baseclf))

pipeline['HankelCov'] = make_pipeline(DownSampler(2),
                                      HankelCovariances(delays=[2, 4, 8, 12, 16], estimator='oas'),
                                      TangentSpace('logeuclid'),
                                      LogisticRegression('l1'))

pipeline['CSSP'] = make_pipeline(HankelCovariances(delays=[2, 4, 8, 12, 16],
                                                   estimator='oas'),
                                 CSP(30),
                                 LogisticRegression('l1'))

dataframe1 = pd.read_csv('ecog_train_with_labels.csv')
# dataframe1 = pd.read_csv('recordingpy.csv')
patients = dataframe1.PatientID.values
dataframe1.describe()


def local_data(data, events, stim_ID, tmin=-.2, tmax=0.399):
    ix_min = int(tmin*1000)
    ix_max = int(tmax*1000)
    nsamp = ix_max - ix_min #nsamp always 300

    X = np.zeros((len(events), data.shape[0], nsamp))
    y =  events
```

```
    st_id = stim_ID

    for i, ix in enumerate(events):
        fromColumn = ix + ix_min
        toColumn = ix + ix_max
        tmp = data[:, slice(fromColumn, toColumn)]
        X[i, :, 0:tmp.shape[1]] = tmp
    return X, y, st_id


ix = patients=='p1'
print(ix.shape)
eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)
print(eeg_data.shape)
print(eeg_data.shape[0])

ix = patients=='p2'
print(ix.shape)
eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)
print(eeg_data.shape)
print(eeg_data.shape[0])

ix = patients=='p3'
print(ix.shape)
eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)
print(eeg_data.shape)
print(eeg_data.shape[0])

ix = patients=='p4'
print(ix.shape)
eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)
print(eeg_data.shape)
print(eeg_data.shape[0])


joblib.dump(pipeline,'pipeline.pkl')

for p in np.unique(patients):
    print(p)
    clfsFull = joblib.load('pipeline.pkl')
    clfs = clfsFull.values()

    ix = patients==p

    eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)

    print("eeg_data.shape")
    print(eeg_data.shape)

    events   = np.int32(dataframe1.Stimulus_Type.loc[ix].values)

    stim_ID  = np.int32(dataframe1.Stimulus_ID.loc[ix].values)
```

```
    picks = (eeg_data!=-999999).mean(1)

    X, y, st_id = local_data(eeg_data[picks==1], events, stim_ID, tmin=0.099, tmax=0.399)

    print("X shape")
    print(X.shape)


#dataframe1 = pd.read_csv('recordingpy.csv')
dataframe1 = pd.read_csv('EEG_Data_Logger.csv')
patients = dataframe1.PatientID.values
#dataframe1.describe()
dataframe1.head()

dataframe1.shape[0]

clfsFull = joblib.load('pipeline.pkl')
clfs = clfsFull.values()
ix = np.ones((dataframe1.shape[0]), dtype=bool)
eeg_data = np.float64(dataframe1.loc[ix].values[:,1:-2].T)
print("eeg_data.shape", eeg_data.shape)
events   = np.int32(dataframe1.Stimulus_Type.loc[ix].values)
print("events.shape", events.shape)
stim_ID  = np.int32(dataframe1.Stimulus_ID.loc[ix].values)
print("stim_ID.shape", stim_ID.shape)
picks = (eeg_data!=-999999).mean(1)
print("picks.shape", picks.shape)
X, y, st_id = local_data(eeg_data[picks==1], events, stim_ID, tmin=0.099, tmax=0.399)
print("X.shape", X.shape)
# print("X", X)
print("y.shape", y.shape)
print("y unique", np.unique(y))
print("st_id.shape", st_id.shape)
print("st_id",st_id)
print("st_id unique", np.unique(st_id))

for clf in clfs:

    print("clf")
    clf.fit(X, y)
    preds = clf.predict_proba(X)
    try:
        scores = cross_val_score(clf,X, y,cv=5)
        print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
    except Exception as e:
        print(e)
```

Appendix 34: EEGClassification.ipynb

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Project ToolsVersion="15.0"
xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import
Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props"
Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Comm
on.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == " ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == " ">AnyCPU</Platform>
    <ProjectGuid>{DF529AF4-A2BB-493B-8308-FC52C27D3868}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>EmotivDataTrasformer</RootNamespace>
    <AssemblyName>EmotivDataTrasformer</AssemblyName>
    <TargetFrameworkVersion>v4.7.1</TargetFrameworkVersion>
    <FileAlignment>512</FileAlignment>
    <ProjectTypeGuids>{60dc8134-eba5-43b8-bcc9-bb4bc16c2548};{FAE04EC0-301F-
11D3-BF4B-00C04F79EFBC}</ProjectTypeGuids>
    <WarningLevel>4</WarningLevel>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
    <Reference Include="Microsoft.CSharp" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="System.Net.Http" />
    <Reference Include="System.Xaml">
      <RequiredTargetFramework>4.0</RequiredTargetFramework>
    </Reference>
    <Reference Include="WindowsBase" />
```

```
    <Reference Include="PresentationCore" />
    <Reference Include="PresentationFramework" />
  </ItemGroup>
  <ItemGroup>
    <ApplicationDefinition Include="App.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </ApplicationDefinition>
    <Page Include="MainWindow.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Compile Include="App.xaml.cs">
      <DependentUpon>App.xaml</DependentUpon>
      <SubType>Code</SubType>
    </Compile>
    <Compile Include="MainWindow.xaml.cs">
      <DependentUpon>MainWindow.xaml</DependentUpon>
      <SubType>Code</SubType>
    </Compile>
  </ItemGroup>
  <ItemGroup>
    <Compile Include="Properties\AssemblyInfo.cs">
      <SubType>Code</SubType>
    </Compile>
    <Compile Include="Properties\Resources.Designer.cs">
      <AutoGen>True</AutoGen>
      <DesignTime>True</DesignTime>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="Properties\Settings.Designer.cs">
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
    <EmbeddedResource Include="Properties\Resources.resx">
      <Generator>ResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.cs</LastGenOutput>
    </EmbeddedResource>
    <None Include="Properties\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <LastGenOutput>Settings.Designer.cs</LastGenOutput>
    </None>
  </ItemGroup>
  <ItemGroup>
    <None Include="App.config" />
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
</Project>
```

Appendix 35: EmotivDataTrasformer.csproj

```
<Window x:Class="EmotivDataTrasformer.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="MainWindow" Height="409.425" Width="538.444"
        xmlns:local="clr-namespace:EmotivDataTrasformer"
        >
    <TabControl Name="TabControl">
        <TabItem Header="Converter">
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="1*"/>
                    <RowDefinition Height="1*"/>
                </Grid.RowDefinitions>
                <GroupBox Header="Input | Emotive Format " Margin="10">
                    <Grid Margin="0,0,-2,-12">
                        <TextBox Height="23" Margin="10,10,62,0" x:Name="inputTextBox"
                         TextWrapping="Wrap" Text="" VerticalAlignment="Top"
TextChanged="TextChanged"/>
                        <Button Content="..." Margin="0,12,10,0"
                        VerticalAlignment="Top" HorizontalAlignment="Right"
                        Width="21" Name="selectInput" Click="selectInput_Click" />
                        <RichTextBox Margin="10,38,10,10" Name="inputMetadataRichTextBox"
>
                            <FlowDocument>
                                <Paragraph>
                                    <Run Text=""/>
                                </Paragraph>
                            </FlowDocument>
                        </RichTextBox>
                        <Button Content="R" ToolTip="Refresh" Margin="0,13,36,0"
                    VerticalAlignment="Top" HorizontalAlignment="Right"
                    Width="21" x:Name="Refresh" Click="Refresh_Click" />
                    </Grid>
                </GroupBox>
                <GroupBox Header="Output | Pyriemann Format" Margin="10,10,10,82"
Grid.Row="1">
                    <Grid Margin="0,0,-2,-12">
                        <Button Content="Convert" Margin="0,0,6,16"
                        Height="20" VerticalAlignment="Bottom"
                        HorizontalAlignment="Right" Width="75" Name="Convert"
Click="Convert_Click"/>
                        <TextBox Height="23" Margin="10,10,36,0" TextWrapping="Wrap"
Text=""
                         x:Name="outputTextBox"
                         VerticalAlignment="Top" TextChanged="TextChanged"/>
                        <Button Content="..." Margin="0,13,10,0" VerticalAlignment="Top"
```

```
                    HorizontalAlignment="Right" Width="21"  Name="selectOutput"
Click="selectOutput_Click"/>
                    <TextBlock HorizontalAlignment="Left" Margin="10,0,0,16"
                      TextWrapping="Wrap" Text="" Width="404" Height="20"
                      VerticalAlignment="Bottom"/>
                </Grid>
            </GroupBox>
        </Grid>
    </TabItem>
    <TabItem Header="Converter CNT Text File To Pyriemann firendly"
Name="CNTTab">
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="1*"/>
                <RowDefinition Height="1*"/>
            </Grid.RowDefinitions>
            <GroupBox Header="Input | CNT Txt Format " Margin="10">
                <Grid Margin="0,0,-2,-12">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto"/>
                        <RowDefinition Height="Auto"/>
                        <RowDefinition Height="*"/>
                    </Grid.RowDefinitions>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition/>
                        <ColumnDefinition/>
                    </Grid.ColumnDefinitions>
                    <TextBlock TextWrapping="Wrap"
                            Margin="5,5,5,10"
                            Text="EEGLab will spit out two kinds of files, one will contain the
EEG Signals and the other contains the events, the result should be one file contains
everthing (the EEG singnals + Events, and all the files should be grouped to one) in a
different format"
                            Grid.ColumnSpan="2"/>
                    <TextBlock Grid.Row="1" TextWrapping="Wrap" Text="CNT text files" />
                    <TextBlock Grid.Column="1" Grid.Row="1" TextWrapping="Wrap"
Text="Events text files"/>
                    <TextBox Name="CNTFile" Margin="5" Grid.Row="2"
TextWrapping="Wrap" AcceptsReturn="True" Text=""/>
                    <TextBox Name="EventFile" Margin="5" Grid.Column="1" Grid.Row="2"
AcceptsReturn="True" TextWrapping="Wrap" Text=""/>
                </Grid>
            </GroupBox>
            <GroupBox Header="Output | Pyriemann Format" Margin="10,10,10,82"
Grid.Row="1">
                <Grid Margin="0,0,-2,-12">
                    <Button Content="Convert" Margin="0,0,10,16"
                    Height="20" VerticalAlignment="Bottom"
                    HorizontalAlignment="Right" Width="71" Name="ConvertCNTToPyr"
Click="ConvertCNTToPyr_Click"/>
                    <TextBox Height="23" Margin="10,10,10,0" TextWrapping="Wrap"
Text=""
```

```
                 x:Name="outputTextBoxCNTToPyr"
                 VerticalAlignment="Top" TextChanged="TextChanged"/>
                <TextBlock HorizontalAlignment="Left" Margin="10,0,0,16"
                  TextWrapping="Wrap" Text="" Width="404" Height="20"
                  VerticalAlignment="Bottom"/>
              </Grid>
            </GroupBox>
          </Grid>
        </TabItem>
        <TabItem Header="Stitcher" Name="Stitcher">
          <Grid>
            <Grid.RowDefinitions>
              <RowDefinition Height="Auto"/>
              <RowDefinition Height="Auto"/>
              <RowDefinition Height="1*"/>
            </Grid.RowDefinitions>
            <GroupBox Header="Input 1 | Pyriemann File" Margin="10">
              <Grid>
                <Grid.RowDefinitions>
                  <RowDefinition Height="Auto"/>
                  <RowDefinition Height="*"/>
                </Grid.RowDefinitions>

                <TextBlock Text="EEg data file in the Pyriemann file format"
                       Grid.Row="0" Margin="5"/>

                <TextBox Height="23"  x:Name="input1PyriemannFile"
                      TextWrapping="Wrap" Text="" VerticalAlignment="Top"
                      TextChanged="TextChanged" Grid.Row="1"
                      Margin="0,0,26,0"/>
                <Button Content="..."
                     VerticalAlignment="Top" HorizontalAlignment="Right"
                     Width="21" Name="selectinput1PyriemannFile"
                     Grid.Row="1" />
              </Grid>
            </GroupBox>
            <GroupBox Header="Input 2 | User Emotion Data" Grid.Row="1">
              <Grid>
                <Grid.RowDefinitions>
                  <RowDefinition Height="Auto"/>
                  <RowDefinition Height="*"/>
                </Grid.RowDefinitions>
                <TextBox Height="23" Margin="10,10,36,0" TextWrapping="Wrap"
Text=""
                      x:Name="input2UserEmotionDataTextBox"
                      VerticalAlignment="Top" TextChanged="TextChanged"/>
                <Button Content="..." Margin="0,13,10,0" VerticalAlignment="Top"
                     HorizontalAlignment="Right" Width="21"
                     Name="UserEmotionDataInput" />
                <TextBlock HorizontalAlignment="Left" Margin="10,0,0,16"
                  TextWrapping="Wrap" Text="" Width="404" Height="20"
                  VerticalAlignment="Bottom"/>
```

```
                <Button Content="Stitch to EEG File" Click="Stitch_Click"
                    Height="20" VerticalAlignment="Bottom"
                    HorizontalAlignment="Right" Name="Stitch"
                    ToolTip="Update the Emotion File"
                    Grid.Row="1" Margin="10,10" />
            </Grid>
        </GroupBox>
    </Grid>
</TabItem>
<TabItem Header="Rename" Name="Rename">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>

        <TextBox Height="23" TextWrapping="Wrap"
              x:Name="FolderTextBox"
              VerticalAlignment="Top" TextChanged="TextChanged"/>

        <Button Content="Rename"
                    Height="20" VerticalAlignment="Bottom"
                    HorizontalAlignment="Right" Name="RenameButton"
                    ToolTip="Will Remove the Char Prefex From the Files names"
                    Grid.Row="1" Margin="10,10" Click="RenameButton_Click" />

    </Grid>
</TabItem>
<TabItem Header="Break down" Name="BreakDown">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <TextBlock TextWrapping="Wrap" Text="Specifing a file and line locations to
split it from ex. 5,9,12 .
                Epocs will be identified from the last column"/>

        <TextBlock Grid.Row="1" TextWrapping="Wrap" Text="File:"/>
        <TextBox Height="23" TextWrapping="Wrap" Name="SplitThisFile"
              VerticalAlignment="Top" TextChanged="TextChanged" Grid.Row="2"/>

        <TextBlock TextWrapping="Wrap" Text="split it at:" Grid.Row="3"/>
```

```
            <TextBox Height="23" TextWrapping="Wrap" Name="SplitTheFileTo"
                    VerticalAlignment="Top" TextChanged="TextChanged" Grid.Row="4"/>


            <TextBlock TextWrapping="Wrap" Text="#of Epoc per file:" Grid.Row="5"/>
            <TextBox Height="23" TextWrapping="Wrap" Name="EpocCount"
                    VerticalAlignment="Top" TextChanged="TextChanged" Grid.Row="6"/>

            <Button Content="Split" Padding="20,2"
                   VerticalAlignment="Bottom"
                        HorizontalAlignment="Right" Name="SplitButton"
                        ToolTip="Will split the file into parts"
                        Grid.Row="7" Margin="10,10" Click="SplitButton_Click" />

        </Grid>
      </TabItem>
      <TabItem Header="Combine Model Prediction Files" Name="ClassesMerge">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="*"/>
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="*"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="*"/>
            </Grid.RowDefinitions>

            <TextBlock Grid.ColumnSpan="2"
                     HorizontalAlignment="Left" Text="input: is the prediction files from each
model. output: is one file contains the final predictions based on the previous model files."
TextWrapping="Wrap" />

            <TextBlock HorizontalAlignment="Left"
                     Grid.Row="1" TextWrapping="Wrap"
                     Text="Input" Margin="0,10,0,0"
                     VerticalAlignment="Top" Grid.ColumnSpan="2"/>
            <TextBox Name="InputClasses"
                    Grid.Row="2" TextWrapping="Wrap" VerticalScrollBarVisibility="Visible"
                    Text="TextBox" Grid.ColumnSpan="2"/>

            <TextBlock HorizontalAlignment="Left"
                     Grid.Row="3" TextWrapping="Wrap"
                     Text="Output" Margin="0,10,0,0"
                     VerticalAlignment="Top" Grid.ColumnSpan="2"/>
            <TextBox Height="23" Grid.Row="4"
                    Name="OutputClasses"
                    TextWrapping="Wrap"
                    Text="TextBox"
```

```
                    VerticalAlignment="Top"
                    Grid.ColumnSpan="2"/>
            <Button Content="Merge" Margin="0,0,10,10" Grid.Row="5"
                   Grid.Column="1" Height="20"
                   VerticalAlignment="Bottom"
                   HorizontalAlignment="Right" Width="75" Click="Button_Click"/>


        </Grid>
      </TabItem>
      <TabItem Header="Create Confusion Metrics" Name="CreateConfusionMetrics">
        <Grid >
          <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
          </Grid.ColumnDefinitions>
          <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
          </Grid.RowDefinitions>
          <TextBlock TextWrapping="Wrap" Margin="5"
                     Text="The true events are 1,2,3,4,5 in order, so for every file you provide
will be matched againest that true events and the TP, TN, FP, FN "
                     VerticalAlignment="Top"
                     Grid.ColumnSpan="2"/>

          <TextBlock HorizontalAlignment="Left"
                    Height="23" Margin="5,5,0,0"
                    Grid.Row="1" TextWrapping="Wrap"
                    Text="Prediction File"
                    VerticalAlignment="Top"
                    Grid.ColumnSpan="2"/>
          <TextBox Height="23" Grid.Row="2"
                    TextWrapping="Wrap"
                    Text=""  Name="PredictionFile"
                    VerticalAlignment="Top"
                    Grid.ColumnSpan="2"/>
          <Button Content="Calculate" Grid.Column="1" Margin="0,0,10,10"
                   Grid.Row="8" HorizontalAlignment="Right"
                   Width="75" Height="20" VerticalAlignment="Bottom"
Click="Button_Click_1"/>

          <TextBox Height="23" Grid.Row="3" Margin="0,10,0,0"
                    TextWrapping="Wrap"
                    Text="Results:"
                    VerticalAlignment="Top"
                    Grid.ColumnSpan="2"/>
```

```
            <StackPanel  VerticalAlignment="Top"
                    Grid.Row="4"
                    Orientation="Horizontal"
                    Margin="0,0,0,0">

                <TextBlock TextWrapping="Wrap" Margin="5" Text="TP:" />
                <TextBlock Name="TP" Text="0" Margin="0,5,10,0" TextWrapping="Wrap"
/>

                <TextBlock TextWrapping="Wrap" Margin="5" Text="TN:" />
                <TextBlock Name="TN" Text="0" Margin="0,5,10,0" TextWrapping="Wrap"
/>

                <TextBlock TextWrapping="Wrap" Margin="5" Text="FP:" />
                <TextBlock Name="FP" Text="0" Margin="0,5,10,0" TextWrapping="Wrap"
/>

                <TextBlock TextWrapping="Wrap" Margin="5" Text="FN:" />
                <TextBlock Name="FN" Text="0" Margin="0,5,10,0" TextWrapping="Wrap"
/>

            </StackPanel>
        </Grid>
      </TabItem>
   </TabControl>
</Window>
```

Appendix 36: MainWindow.xaml

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace EmotivDataTrasformer
{
   public partial class MainWindow : Window
```

```
{
    public static Random Random = new Random();
    public MainWindow()
    {
        InitializeComponent();

        //Tab 1
        inputTextBox.Text = Properties.Settings.Default.inputPath;
        outputTextBox.Text = Properties.Settings.Default.outputPath;

        //Tab 2
        input1PyriemannFile.Text = Properties.Settings.Default.Input1;
        input2UserEmotionDataTextBox.Text = Properties.Settings.Default.Input2;

        //Tab 3
        FolderTextBox.Text = Properties.Settings.Default.FolderTextBox;

        TabControl.SelectedItem = CreateConfusionMetrics;

        SplitThisFile.Text = @"C:\Users\Mohamed Ahmed Abdull\Downloads\External
EEG Data\sce\eegData1.csv";
        //SplitTheFileTo.Text = @"6780,10825,43186,83815,124736,162361";
        EpocCount.Text = "2";

        CNTFile.Text = @"C:\Users\Mohamed Ahmed Abdull\Downloads\External EEG
Data\sce\sce1.txt
C:\Users\Mohamed Ahmed Abdull\Downloads\External EEG Data\sce\sce2.txt";
        EventFile.Text = @"C:\Users\Mohamed Ahmed Abdull\Downloads\External EEG
Data\sce\event1.txt
C:\Users\Mohamed Ahmed Abdull\Downloads\External EEG Data\sce\event2.txt";
        outputTextBoxCNTToPyr.Text = @"C:\Users\Mohamed Ahmed
Abdull\Downloads\External EEG Data\sce\eegData.csv";

        InputClasses.Text = @"C:\Users\Mohamed Ahmed Abdull\sources\33preds0.out
C:\Users\Mohamed Ahmed Abdull\sources\33preds1.out
C:\Users\Mohamed Ahmed Abdull\sources\33preds2.out
C:\Users\Mohamed Ahmed Abdull\sources\33preds3.out
C:\Users\Mohamed Ahmed Abdull\sources\33preds4.out";
        OutputClasses.Text = @"C:\Users\Mohamed Ahmed
Abdull\sources\33predsMerge.out";

        PredictionFile.Text = @"C:\Users\Mohamed Ahmed
Abdull\sources\22predsMerge.out";
    }

    private void Refresh_Click(object sender, RoutedEventArgs e)
    {
        var metaData = new Transformer().GetHeaders(inputTextBox.Text);
        inputMetadataRichTextBox.AppendText(metaData);
    }

    private void selectInput_Click(object sender, RoutedEventArgs e)
```

```
        {

        }

        private void selectOutput_Click(object sender, RoutedEventArgs e)
        {

        }

        private void Convert_Click(object sender, RoutedEventArgs e)
        {
            new Transformer().PyriemannTransform(inputTextBox.Text, outputTextBox.Text);
        }

        private void TextChanged(object sender, TextChangedEventArgs e)
        {
            TextChanged();
        }

        public void TextChanged()
        {
            Properties.Settings.Default.inputPath =
string.IsNullOrWhiteSpace(inputTextBox.Text) ?
                                    Properties.Settings.Default.inputPath :
                                    inputTextBox.Text;
            Properties.Settings.Default.outputPath =
string.IsNullOrWhiteSpace(outputTextBox.Text) ?
                                    Properties.Settings.Default.outputPath :
                                    outputTextBox.Text;

            Properties.Settings.Default.Input1 =
string.IsNullOrWhiteSpace(input1PyriemannFile.Text) ?
                                  Properties.Settings.Default.Input1 :
                                  input1PyriemannFile.Text;

            Properties.Settings.Default.Input2 =
string.IsNullOrWhiteSpace(input2UserEmotionDataTextBox.Text) ?
                                  Properties.Settings.Default.Input2 :
                                  input2UserEmotionDataTextBox.Text;

            Properties.Settings.Default.FolderTextBox =
string.IsNullOrWhiteSpace(FolderTextBox.Text) ?
                                  Properties.Settings.Default.FolderTextBox :
                                  FolderTextBox.Text;

            Properties.Settings.Default.Save();
        }

        private void Stitch_Click(object sender, RoutedEventArgs e)
        {
            new Stitch(input1PyriemannFile.Text, input2UserEmotionDataTextBox.Text);
        }
```

```
        private void RenameButton_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                foreach (var file in Directory.GetFiles(FolderTextBox.Text))
                {
                    var name = System.IO.Path.GetFileNameWithoutExtension(file);
                    var extention = System.IO.Path.GetExtension(file);
                    var digits = Regex.Matches(name, @"\d+");
                    var newName = "";
                    foreach (Match digit in digits)
                        newName += digit.Value;
                    var newNameAsNumber = long.Parse(newName);
                    var distination = System.IO.Path.GetDirectoryName(file) + @"\" +
newNameAsNumber + extention;
                    File.Move(file, distination);
                }
            }
            catch (Exception ee)
            {
                MessageBox.Show(ee.Message + ee.StackTrace);
            }
        }

        #region Converter CNT Text File To Pyriemann firendly
        private void ConvertCNTToPyr_Click(object sender, RoutedEventArgs e)
        {
            new CntTransformer().CntTransform(CNTFile.Text.Split('\n').ToList(),
                                  EventFile.Text.Split('\n').ToList(),
                                  outputTextBoxCNTToPyr.Text);
        }
        #endregion

        #region Break down
        private void SplitButton_Click(object sender, RoutedEventArgs e)
        {
            new SplitFile().SplitByEpoc(SplitThisFile.Text, EpocCount.Text );

            //SplitTheFileTo
        }
        #endregion

        #region Combine Model Prediction Files
        private void Button_Click(object sender, RoutedEventArgs e)
        {
            new Merger().Merge(InputClasses.Text, OutputClasses.Text);
        }
        #endregion

        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
```

```
            var cm = new CalculateConfusionMatrics().Calculate(PredictionFile.Text);
            //TP.Text = cm.TP+"";
            //TN.Text = cm.TN+"";
            //FP.Text = cm.FP+"";
            //FN.Text = cm.FN+"";
        }
    }

    public class Transformer
    {
        public string GetHeaders(string path)
        {
            //Get the Headers
            return File.ReadAllLines(path)[0];
        }

        public bool PyriemannTransform(string input, string output)
        {
            try
            {
                var lines = File.ReadAllLines(input).Skip(1).ToList();
                var newLines = new List<string>();

//@"""PatientID"",""Electrode_1"",""Electrode_2"",""Electrode_3"",""Electrode_4"",""Elec
trode_5"",""Electrode_6"",""Electrode_7"",""Electrode_8"",""Electrode_9"",""Electrode_10
"",""Electrode_11"",""Electrode_12"",""Electrode_13"",""Electrode_14"",""Stimulus_Type
"",""Stimulus_ID"""

newLines.Add(@"""PatientID"",""Electrode_1"",""Electrode_2"",""Electrode_3"",""Electro
de_4"",""Electrode_5"",""Electrode_6"",""Electrode_7"",""Electrode_8"",""Electrode_9"",""
""Electrode_10"",""Electrode_11"",""Electrode_12"",""Electrode_13"",""Electrode_14"",""S
timulus_Type"",""Stimulus_ID""");
                foreach (var line in lines)
                {
                    newLines.Add(PyriemannTransform(line));
                }

                File.WriteAllLines(output, newLines);

                return true;
            }
            catch (Exception exception)
            {
                MessageBox.Show(exception.Message);
                return false;
            }
        }

        private string PyriemannTransform(string line)
        {
            //"PatientID","Electrode_1",...,"Electrode_64","Stimulus_Type","Stimulus_ID"
            string newLine = "";
```

```
        //[0]: "title:recording1"
        //[1]: " recorded:17.06.18 15.04.07"
        //[2]: " timestamp started:2018-06-17T15:04:07.453+08:00;1529219047.453"
        //[3]: " sampling:128"
        //[4]: " subject:mohamed"
        //[5]: " labels:COUNTER INTERPOLATED AF3 F7 F3 FC5 T7 P7 O1 O2 P8 T8
FC6 F4 F8 AF4 RAW_CQ GYROX GYROY MARKER MARKER_HARDWARE SYNC
TIME_STAMP_s TIME_STAMP_ms TimeStamp CQ_AF3 CQ_F7 CQ_F3 CQ_FC5
CQ_T7 CQ_P7 CQ_O1 CQ_O2 CQ_P8 CQ_T8 CQ_FC6 CQ_F4 CQ_F8 CQ_AF4
CQ_CMS CQ_DRL"
        //[6]: " chan:40"
        //[7]: " samples:11264"
        //[8]: " units:emotiv"

        //4171.281738, 4141.025391, 4136.410156, 4086.153809, 4129.743652,
4141.538574, 2949.000000, 8202.000000, 8228.000000, 1.000000,
        var data = line.Split(',');

        newLine += "p1,"; //PatientID
        newLine += data[2] + ","; //AF3
        newLine += data[3] + ","; //F7
        newLine += data[4] + ","; //F3
        newLine += data[5] + ","; //FC5
        newLine += data[6] + ","; //T7
        newLine += data[7] + ","; //P7
        newLine += data[8] + ","; //O1
        newLine += data[9] + ","; //O2
        newLine += data[10] + ","; //P8
        newLine += data[11] + ","; //T8
        newLine += data[12] + ","; //FC6
        newLine += data[13] + ","; //F4
        newLine += data[14] + ","; //F8
        newLine += data[15] + ","; //AF4
        var random = MainWindow.Random.Next(4);
        newLine += GetStimulusType(random); //Stimulus_Type
        newLine += GetStimulusId(random); //Stimulus_ID

        return newLine;
    }

    public string GetStimulusType(int r)
    {
        switch (r)
        {
            case 0: return "94,";
            case 1: return "101,";
            case 2: return "90,";
            case 3: return "91,";
            default:
                throw new Exception("no class for StimulusType, review GetStimulusType
function");
        }
```

```
        }

        public string GetStimulusId(int r)
        {
            switch (r)
            {
                case 0: return "3";
                case 1: return "4";
                case 2: return "5";
                case 3: return "6";
                default:
                    throw new Exception("no class for StimulusId, review GetStimulusId
function");
            }
        }
    }

    public class CntTransformer
    {
        public bool CntTransform(List<string> cntFiles, List<string> eventsFiles, string
output)
        {
            try
            {
                //var newLines = new List<string>();

//newLines.Add(@"""PatientID"",""Electrode_1"",""Electrode_2"",""Electrode_3"",""Electr
ode_4"",""Electrode_5"",""Electrode_6"",""Electrode_7"",""Electrode_8"",""Electrode_9"",
""Electrode_10"",""Electrode_11"",""Electrode_12"",""Electrode_13"",""Electrode_14"",""
Stimulus_Type"",""Stimulus_ID""");

                //32 Channels > data per channel from all the files
                List<List<double>> EEGData = new List<List<double>>();

                //Init the Memory
                for (int i = 0; i < 31; i++)
                    EEGData.Add(new List<double>());

                //Loop throw the CNT files
                foreach (var cntFile in cntFiles)
                {
                    var cntFileLines = File.ReadAllLines(cntFile.Trim());
                    var lineIndex = 0;

                    //Loop throw the lines
                    foreach (var cntLine in cntFileLines.Skip(1))
                    {
                        var data = new List<double>();
                        var txtData = cntLine.Split('\t').Skip(1);
                        data.AddRange(txtData.Select(s =>
                        {
                            try
```

```
                {
                    return double.Parse(s);
                }
                catch (Exception e)
                {
                    return double.MinValue;
                }
            }));

            EEGData[lineIndex].AddRange(data);
            lineIndex++;
        }
    }

    var events = new List<Event>();
    var previousFileEndAt = 0;
    //Loop thorw the event Files
    foreach (var eventFile in eventsFiles)
    {
        var eventFileLines = File.ReadAllLines(eventFile.Trim());

        //Loop throw the lines
        foreach (var eventLine in eventFileLines.Skip(1).Where(w =>
!string.IsNullOrWhiteSpace(w)))
        {
            //number | type | latency | urevent
            var txtData = eventLine.Split('\t');
            events.Add(new Event
            {
                Latency = intParse(txtData[2]) + previousFileEndAt,
                EventType = intParse(txtData[3])
            });
        }
        previousFileEndAt += events.Last().Latency;
    }


    ////Transform EEGData
    //var EEGDataTransformed = new List<EEGData>();
    //for (int i = 0; i < EEGData.Count; i++)
    //{
    //    EEGDataTransformed.Add(new EEGData { Channels = EEGData[i] });
    //}

    var newLines = new List<string>();
    var ElectrodeTitles = "";
    for (int i = 1; i <= 31; i++)
        ElectrodeTitles += $"\"Electrode_{i}\",";

    var header = @"""PatientID"","
        + ElectrodeTitles
        + @"""Stimulus_Type"",""Stimulus_ID""";
```

```
        newLines.Add(header);
        try
        {
            for (int i = 0; i < EEGData[0].Count; i++)
            {
                //Patient Id
                var line = $"1,";
                for (int j = 0; j < EEGData.Count; j++)
                {
                    line += $"{EEGData[j][i]},";
                }

                //determine the event based on raw number i
                var Type = DeterminTheType(events, i);

                newLines.Add( $"{line}{Type},{Type}");
            }
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }

        File.AppendAllLines(output, newLines);

        MessageBox.Show("Done");
        return true;
    }
    catch (Exception exception)
    {
        MessageBox.Show(exception.Message);
        return false;
    }
}

private int DeterminTheType(List<Event> events, int i)
{
    return events[events.FindIndex(f => f.Latency >= i)].EventType;
}

private int intParse(string text)
{
    try
    {
        return (int) double.Parse(text);
    }
    catch
    {
        return int.MinValue;
    }
}
```

```
    }

    public class EEGData
    {
        /// <summary>
        /// FP1 | FP2 | ...
        /// 1   |  2  | ...
        /// </summary>
        public List<double> Channels { get; set; }

        /// <summary>
        /// In this example the stimula and the
        /// </summary>
        public int Type { get; set; }
    }

    public class Event
    {
        public int Latency { get; set; }
        public int EventType { get; set; }
    }

    public class Stitch
    {
        public Stitch(string eegFile, string emotionFile)
        {
            try
            {
                //Emotion File Format
                //Image Name, Emotion Number(Sam number)

                //EEG File Format
                //PatientId, EEG channel1, EEG Channel2, ... EEG Channel14, Stimulus_Type,
Stimulus_ID

                //Mission is to replace "Stimulus_Type" and "Stimulus_ID" with the "Emotion
Number" in the EEG File

                var emotions = GetEmotion(emotionFile);

                var lines = File.ReadLines(eegFile).Where(w =>
!string.IsNullOrWhiteSpace(w)).ToList();
                var outputFile = System.IO.Path.GetDirectoryName(eegFile) + @"\" +
                        System.IO.Path.GetFileNameWithoutExtension(eegFile) +
"Converted.csv";
                File.Delete(outputFile);

                File.AppendAllText(outputFile, lines[0] + Environment.NewLine);
                var lineCounter = lines.Count() - 1;
                foreach (var line in lines.Skip(1))
                {
```

```
                File.AppendAllText(outputFile, UpdateEmotion(line, emotions) +
Environment.NewLine);
                Debug.WriteLine(lineCounter--);
            }
            MessageBox.Show("Done :)");
        }
        catch (Exception exc)
        {
            MessageBox.Show(exc.Message + exc.StackTrace);
        }
    }

    private string UpdateEmotion(string eegLine, List<Emotion> emotions)
    {
        var parts = eegLine.Split(',');

        //15, 16 is the indexes of Stimulus_Type and Stimulus_ID
        var emotion = emotions.FirstOrDefault(f => f.ImageName == int.Parse(parts[15]));
        if (emotion != null)
        {
            parts[15] = emotion.SamEmotionId + "";
            parts[16] = emotion.SamEmotionId + "";
        }

        var newLine = parts.Aggregate((a, b) => (a + ", " + b));
        return newLine;
    }

    public List<Emotion> GetEmotion(string fromFile)
    {
        var lines = File.ReadAllLines(fromFile)
                .Where(w => !string.IsNullOrWhiteSpace(w));

        var emotions = new List<Emotion>();
        foreach (var line in lines)
        {
            var parts = line.Split(',');
            emotions.Add(new Emotion
            {
                ImageName = int.Parse(parts[0]),
                SamEmotionId = int.Parse(parts[1])
            });
        }

        return emotions;
    }
}

public class Stimulus
{
    public int Type { get; set; }
    public int Id { get; set; }
```

```
    }

    public class Emotion
    {
        public int ImageName { get; set; }
        public int SamEmotionId { get; set; }
    }

    public class SplitFile
    {
        public void Split(string file, string atTxt)
        {
            var allLinesWithHeader = File.ReadAllLines(file.Trim());
            var Header = allLinesWithHeader.First();
            var allLines = allLinesWithHeader.Skip(1);
            var extention = System.IO.Path.GetExtension(file.Trim());
            var partsPatialNames = System.IO.Path.GetFileNameWithoutExtension(file.Trim());
            var at = atTxt.Split(',').Select(s => int.Parse(s)).ToList();
            at.Insert(0, 0);

            //var from = 0;
            for (int to = 0; to < at.Count(); to++)
            {
                var skip = at[to];

                //Last One?
                var take = (at.Count() == to + 1) ?
                        allLines.Count() - skip :
                        at[to + 1] - skip;

                var lines = allLines.Skip(skip).Take(take);

File.AppendAllLines($"{System.IO.Path.GetDirectoryName(file.Trim())}\\{partsPatialNam
es}{to}{extention}", lines);
            }
        }

        public void SplitByEpoc(string file, string countTxt)
        {
            var allLinesWithHeader = File.ReadAllLines(file.Trim());
            Split(file, GetEpocs(allLinesWithHeader, int.Parse(countTxt)));
        }

        private string GetEpocs(string[] EegData, int count)
        {
            var epocs = "";
            var counter = 0;
            var previousEvent = GetEvent(EegData[1]);
            var i = 0;
            do
            {
                i++;
```

```
            if (previousEvent == GetEvent(EegData[i]))
            {
                //Same as previous
            }
            else
            {
                //Changed from the previous
                counter++;
                previousEvent = GetEvent(EegData[i]);
            }

            if (count == counter)
            {
                //Reset counter
                counter = 0;
                epocs += $",{i}";
            }

        } while (i < EegData.Length-1);

        return new string(epocs.Skip(1).ToArray());
    }

    private string GetEvent(string line)
    {
        return line.Split(',').Last().Trim();
    }
}

public class Merger
{
    public void Merge(string inputFiles, string output)
    {
        //1. Read Files
        var filesPath = inputFiles.Split('\n');

        //5 classes and the content of each file
        var files = new List<List<Classes>>();
        foreach (var filePath in filesPath)
        {
            var file = new List<Classes>();
            var lines = File.ReadAllLines(filePath.Trim());
            foreach (var line in lines)
            {
                var parts = line.Split(',');
                file.Add(new Classes {
                    Class1 = float.Parse(parts[0]),
                    Class2 = float.Parse(parts[1]),
                    Class3 = float.Parse(parts[2]),
                    Class4 = float.Parse(parts[3]),
                    Class5 = float.Parse(parts[4]),
```

```
                });
            }

            files.Add(file);
        }

        //2. Merge The Data
        var total = new List<Classes>();
        for (int i = 0; i < files.First().Count; i++)
        {
            var maxC1 = Max(files[0][i].Class1, files[1][i].Class1, files[2][i].Class1,
files[3][i].Class1, files[4][i].Class1);
            var maxC2 = Max(files[0][i].Class2, files[1][i].Class2, files[2][i].Class2,
files[3][i].Class2, files[4][i].Class2);
            var maxC3 = Max(files[0][i].Class3, files[1][i].Class3, files[2][i].Class3,
files[3][i].Class3, files[4][i].Class3);
            var maxC4 = Max(files[0][i].Class4, files[1][i].Class4, files[2][i].Class4,
files[3][i].Class4, files[4][i].Class4);
            var maxC5 = Max(files[0][i].Class5, files[1][i].Class5, files[2][i].Class5,
files[3][i].Class5, files[4][i].Class5);
            total.Add(new Classes { Class1 = maxC1, Class2 = maxC2, Class3 = maxC3,
Class4 = maxC4, Class5 = maxC5 });
        }

        //3. Write on file
        File.AppendAllLines(output, total.Select(s =>
$"{s.Class1},{s.Class2},{s.Class3},{s.Class4},{s.Class5}"));

        MessageBox.Show("Done.");
    }

    public static float Max(params float[] values)
    {
        return values.Max();
    }
}

public class Classes
{
    public float Class1 { get; set; }
    public float Class2 { get; set; }

    public float Class3 { get; set; }

    public float Class4 { get; set; }

    public float Class5 { get; set; }

}

public class CalculateConfusionMatrics
{
```

```
public ConfusionMatrics Calculate(string file)
{
    var lines = File.ReadAllLines(file.Trim());

    //The max Class index from the file
    var predClasses = new List<int>();

    var previousLine = "";
    foreach (var line in lines)
    {
        if (line == previousLine)
            continue;

        var parts = line.Split(',');

        //Had to execlude the last predictions cuz it's scuing the results
        var pred = MaxIndex(float.Parse(parts[0]), float.Parse(parts[1]),
float.Parse(parts[2]), float.Parse(parts[3])); //, float.Parse(parts[4]));

        //changin the index to the class
        predClasses.Add(pred +1 );
        previousLine = line;
    }

    var trueClasses = new List<int> { 1, 2, 3, 4, 5 };
    var _1Classindex = 0;
    var _2Classindex = 1;
    var _3Classindex = 2;
    var _4Classindex = 3;
    var _5Classindex = 4;
    var matrix = new ConfusionMatrics { };

    for (int i = 0; i < predClasses.Count; i++)
    {
        if (predClasses[i] == trueClasses[i])
        {
            //Match
            matrix.Matrix[i][i] += 1;
        }
        else
        {
            //Not a match
            matrix.Matrix[i][predClasses[i]] += 1;
        }
    }

    return matrix;
}

public int MaxIndex(params float[] values)
{
    return values.ToList().IndexOf(values.Max());
```

```
        }
    }

    public class ConfusionMatrics
    {
        public ConfusionMatrics()
        {
            Matrix = new Dictionary<int, Dictionary<int, int>>();
            Matrix.Add(0,new Dictionary<int, int>());
            Matrix.Add(1,new Dictionary<int,int>());
            Matrix.Add(2,new Dictionary<int,int>());
            Matrix.Add(3,new Dictionary<int,int>());
            Matrix.Add(4, new Dictionary<int,int>());
        }
        //public int TP { get; set; }
        //public int TN { get; set; }
        //public int FP { get; set; }
        //public int FN { get; set; }

        /// <summary>
        /// true then predictions
        /// </summary>
        public Dictionary<int,Dictionary<int,int>> Matrix { get; set; }
    }
}
```

Appendix 37:MainWindow.xaml.cs