University of Hertfordshire UH

PROJECT REPORT

BEng (Hons) Electronics and Communication Engineering

Name: Flavio Grillo Marín

Supervisor: PhD Iosif Mporas

Development of an AI model based on Machine Learning to detect schizophrenia through an EEG

14/05/2020

BACHELOR OF ENGINEERING DEGREE/DEGREE WITH HONOURS
BEng (Hons) Electronics and Communication Engineering

Department of Engineering
School of Engineering and Computer Science
University of Hertfordshire

Development of an AI model based on Machine Learning to detect schizophrenia through an EEG

Report by
FLAVIO GRILLO MARÍN

Supervisor
PhD IOSIF MPORAS

Date
14/05/2020

DECLARATION STATEMENT

I certify that the work submitted is my own and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged (ref. UPR AS/C/6.1, Appendix I, Section 2 – Section on cheating and plagiarism)

Student Full Name: FLAVIO GRILLO MARÍN

Student Registration Number: 18058567

Signed: …… FLAVIO GRILLO MARÍN ……

Date: ……14/05/2020…….

ABSTRACT

This project proposes an architecture for the classification of electroencephalograms between schizophrenics and healthy patients. To achieve a classifier model, the necessary steps are explained, among which are the essential processes such as signal pre-processing, obtaining features and creating the model. The features are divided into three types: Time Domain, Frequency Domain and Functional and Effective features. The latter are connectivity measures that are processed with statistical formulas based on graph theory. To obtain the best model, an experiment is carried out to study the most common algorithms in machine learning given the previously calculated data set. It is also studied whether the application of Principal Component Analysis (PCA) improves the model. The result of the investigation is the model obtained with the Weighted KNN classifier with k = 35 and without PCA, which generates an accuracy of 83.4% and a recall of 85.2%. As an additional outcome, a user interface is developed, in order to provide greater usability to the project.

Keywords: Schizophrenia, EGG, AI, Signal Processing

ACKNOWLEDGEMENTS

I would like to give my most sincere and profound thanks to all the teachers who have brought me to this point in my life. Without their dedication and effort to spread their knowledge, neither I nor my colleagues would have gone that far. I would especially like to thank PhD. Iosif Mporas, Senior Lecturer in Information Engineering in the Hertfordshire University. It was him who offered me this project. He also provided me with the methodology and an invaluable guide to carry out the project.

Also, I want to thank my family for pushing me to achieve my goals and supporting me in every difficult time of my life. Since I was little I have been given the tools and ideas necessary to develop myself as an independent, hard-working and enthusiastic person. They are a constant support, which I can always count on for everything I need. Without their efforts, none of what I've accomplished in my life would be possible.

I would also like to thank Aitana, who since I met her in class in September 2014, I already knew that she was going to be someone relevant in my life, but now I look back in time and at that moment I had not the slightest idea of how important it would be having her by my side, for all the support and love she have given me all these years.

Finally, I would like to thank all my friends, those who are always there for good and bad. Those who make you get a smile and distract yourself in moments of stress. Here I especially want to thank Javier Balbás and Javier González. Javier Balbás from the first day we met when entering the university, we have been close friends and we have always helped each other both inside and outside the university, and I am sure that we will continue doing it every day and every year. Javier González has also been a very close friend from the beginning of the university and this past year he has shown me, through mutual support, that friends like these last for a long time.

I would like to name all those relevant people who have also contributed to my personal development, but the list is too long. For this reason, I want to thank all those who are alluded to when reading these words. Without you I would not be the same person.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

- **AI: Artificial Intelligence**
- **EEG:** Electroencephalogram
- **fMRI:** Functional magnetic Resonance Imaging
- **CT Scan:** Computerized Tomography
- **EDF:** European Data Format
- **EEA format:** Encrypted e-mail attachment
- **MFCCs:** Mel-frequency cepstral coefficients
- **HPS:** Harmonic Product Spectrum
- **COR:** Pearson's Correlation Coefficient
- **xCOR:** Correlation
- **COH:** Coherence
- **iCOH:** Imaginary Coherence
- **PLV:** Phase Locking Value
- **PLI:** Phase Locking Index
- **wPLI:** weighted Phase Locking Index
- **RHO:** ρ index
- **FFT:** Fast Fourier Transform
- **GS:** Generalized Syncronization
- **MI:** Mutual Information
- **IT:** Information Theory
- **PMI:** Partial Mutual Information
- **TE:** Transfer Entropy
- **PTE:** Partial Transfer Entropy
- **SVM:** Support Vector Machine
- **KNN:** k-nearest neighbours
- **PCA:** Principal Component Analysis

1. INTRODUCTION

Research has given great joy to mankind over the years, but there are still two major barriers that resist us: outer space and the human brain. One of them is millions of kilometres away from us and the other is the reason for who we are.

The human being through research takes small steps that, although they may seem irrelevant, are the bases on which future research will be based. The brain is our central computer and neuroscience is in charge of understanding how that computer works. From neuron theory until now, many scientists have done their research and theories on how the brain works. It can be said that we know more and more, but there is still much to know.

Within the field of neuroscience, mental illnesses are unknown. In ancient times they were considered to be demonic possessions and to heal the sick they subjected them to confinement and torture. As neuroscience progressed, some scientists theorized that mental illness could be the cause of mental disorders, so the way of acting was drastically changed. Currently we know a wide variety of mental illnesses, we know the symptoms and we know how to act against them.

Technology has played an essential role in neuroscience. Thanks to psychology and neuroimaging techniques, very interesting conclusions have been drawn about the functioning of the brain. Currently, Artificial Intelligence is booming, which in its branch of Machine Learning, has already proven to be very useful for obtaining complex results faster and in a simpler way.

Throughout this research, the electroencephalogram technique will be unified with the use of machine learning, to obtain a new diagnostic method that helps doctors diagnose schizophrenia patients with greater precision. To do so, an investigation will be made of what is schizophrenia and what are the main neuroimaging techniques, followed by an experimental approach that unites the digital signal processing with the most widely used machine learning algorithms.

The result of the project is the creation of an artificial intelligence model that predicts with great precision whether a patient suffers from schizophrenia or not. A user interface will also be developed so that doctors and health technicians can use the results of this research.

2. SCHIZOPHRENIA

The word "Schizophrenia" comes from the Greek and means "split mind." From this translation you can see the root of the problem it causes. Schizophrenia is a mental disorder that affects the patient making him feel, act and think in a different way. This disorder is suffered by almost 1% (WHO, 2019) of the world's population, which makes it one of the Top15 leading causes of disability worldwide (Naghavi, 2019).

2.1. Causes

The cause is still unclear. Some of the investigations that are being carried out highlight different causes:

- Genetics: The disorder can be inherited from parents to children. It can also be caused by viral infections or environmental events. Like other genetic diseases, it also develops with hormonal and physical changes, which makes sense because the age at onset typically begin in early adulthood (15-25 years old). (Sham, MacLean and Kendler, 1994)

- Chemistry: People with this disorder have an imbalance in the levels of serotonin and dopamine, which are neurotransmitters. This imbalance affects how the person reacts to stimuli that a healthy person easily controls. These stimuli, such as a lot of noise or a lot of light, can lead to hallucinations and illusions in the patient.(Stuart and Laraira, 2006)

In addition to these factors, they also affect prenatal and social circumstances or drug addiction among others.

2.2. Symptoms

Symptoms occur over months or years and the patient does not have to suffer from all the symptoms to have the disorder. At an early stage, typical symptoms are irritability, difficulty concentrating and difficulty sleeping. As the disease continues, this affects the brain in a more serious way, causing problems in thinking, emotions and behaviour. Symptoms are divided into two classes:

- Negative symptoms: Are those that lead to a loss in psychomotor activity. Some clear manifestations are social alienation, apathy, lack of emotional response or lack of motivation.

- Positive symptoms: They are what carry a different perception of reality. The distortion of reality is presented in the form of illusions, hallucinations or in speech and meaningless thinking, creating new words.

In addition to these symptoms, people who suffer from the disease are more likely to develop other disorders such as anxiety attacks and depression (Sim *et al.*, 2006).

2.3. Diagnosis

Not knowing the cause of the disorder and the symptoms vary from one patient to another, make the diagnosis of the disease very complicated. There is currently no medical test through which to diagnose schizophrenia. To obtain a diagnosis, a psychiatrist must analyse the patient and find characteristic symptoms of the disorder, which are defined by the American Psychiatric Association and the World Health Organization.

Neuroimaging methods are currently being used to rule out possible diseases, but not for diagnostic purposes. With these methods characteristic patterns of the disease can be found. The problem is that each patient has different brain patterns and today only a few are known.(Keepers *et al.*, 2020)

2.4. Living with Schizophrenia

People suffering from schizophrenia can live relatively normal lives as long as they comply with the medication. Scientific advances have managed to create drugs that reduce the positive symptoms, such as hallucinations or delusions. However, negative symptoms cannot be medicated. This leads to schizophrenic people in society, generally relate less.

People with schizophrenia are between 2 and 2.5 times more likely to die at an early age, according to (WHO, 2019). This is due to cardiovascular, metabolic and infectious diseases, but also suicide, since people suffering from schizophrenia are 50% more likely to attempt suicide.(Naghavi, 2019)

3. Neuroimaging Techniques

To analyse the brain there are many analysis techniques. The three most used are fMRI, CT Scan and EEG. Doctors use one or the other depending on the patient's symptoms. In addition to differentiating the tests by their physical principle, they can be differentiated by cost per test, accuracy, duration, portability of the equipment and necessary training, among others. A combination of these characteristics makes each test more suitable to each use.

3.1. fMRI

The fMRI measures brain activity by according to blood flow inside the brain. It uses the Blood-Oxygen level dependent contrast, which detect what neurons are active by the oxygen they demand to the circulatory system. Active neurons demand more oxygen than inactive one due the lack of energy reserves (*fMRI*, 2018).

The way it's done is with powerful magnets which makes two different magnetic fields, one to align nuclei in the area of study and the other one to locate the rest of nuclei. After the magnetic fields has been setter is needed a radiofrequency pulse which makes the nuclei get in a higher energy level. Once the nuclei have received enough energy to get to that higher level, the RF is removed, and the nuclei return to the equilibrium state, emitting energy which is measured with large coils. The data obtained is processed with specific software and the result is an image with the position of the nuclei.

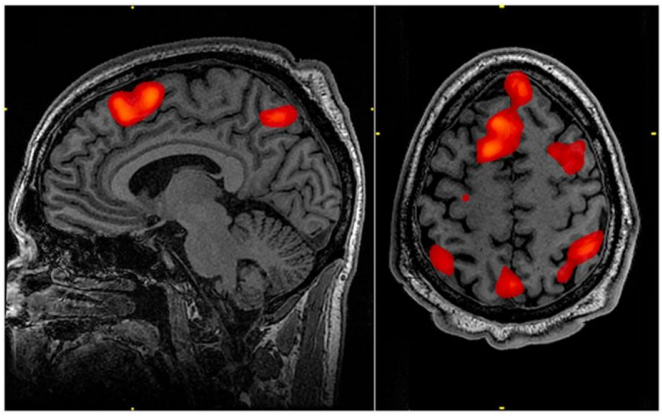


Figure 1: fMRI

The fMRI is a noninvasive test, which means that no break in the skin is done and there is no cavity exploration by an artificial or natural orifice. Being a noninvasive test it's probably the main advantage of this technique followed by the fact that fMRI don't use radiation, as X-rays. The fMRI has other advantages like the precision, which it can be as much as 1 millimeter, easy to use and effectiveness.

As all the tests, fMRI also have some disadvantages, which are:

- Doing a fMRI is very expensive.
- The patient needs to be still for a long time.
- Difficulty to understand the results obtained.
- Detects groups of thousands active neurons, which it is too much.

The common uses for the fMRI are various, in which the most notable are the monitoring of brain tumors and to help physicians to diagnose and watch cerebrovascular diseases as strokes or degenerative diseases as Alzheimer. Also is widely used in pre-surgery stages and on scientific purposes.

3.2. CT Scan

CT Scan which produces cross-sectional images radiating the body with X-rays. This special x-ray test allows the medical staff to watch inside the patient. The plain fil X-ray film has obstacles as bones or organs that interfere in the medical image, so watching a cleaner image (CT) will give different points of view, improving the diagnosis. (OpenLearn, 2019)

The physical principle within CT and plain X-Ray is that in an X-Ray the beam generator stays still, and in the CT Scan the generator moves around the patient so with the help of specific software a clean image can be obtained.

Computerized Tomography Scan is used to diagnose tumors, internal injuries such as clots or bleedings, fractures or internal trauma. (Johns Hopkins Medicine, 2017)

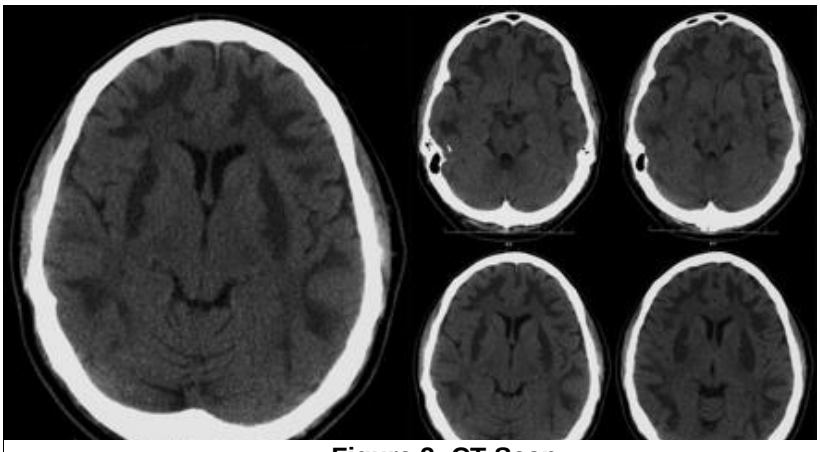


Figure 2: CT Scan

3.3. EEG

The EEG is the only one, between CT Scan and fMRI that it is not a medical image, but a wave pattern which records the electrical pattern in the brain. Neurons or brain cells use electricity pulses to transmit information between each other, so in the EEG test record those pulses using electrodes that are placed in the patient´s head. This test is noninvasive as it only needs a helmet with normally 16 channels and 21 electrodes, displayed in the International 10/20 Standard way . (Morley Andrew Morley Hons and Hill, 2016)

The electrical signals from the ionic current between neurons are called brain waves. These waves are received by the electrodes and amplified by the EEG machine. The result is shown in graph paper or a digital screen. The EEG does not introduce electricity into the patient´s brain, so it's totally safe to use it. There are different tests that are diagnosed depending on what the doctor is looking for.

The common one is the Routine EEG which takes around 30 minutes, in which the patient must be in rest and the doctor will ask for different actions to be performed, such as closing eyes or breathing deeply.

The second one is the Sleep EEG, which takes a whole night to do it. In this test, the patient will sleep in a controlled room, sometimes being video-recorded, and tested with an EEG machine. This Sleep EEG test is very useful for sleep disorders and if the Routine EEG doesn't give the doctors enough valuable information.

The last one is the Ambulatory EEG which takes as much as the medical staff need to obtain real results. This test will be done without interfering the normal life of the patient, so the EEG machine in this case is portable. In all this test, video-recording is sometimes used to obtain extra information, such as peak neuronal activity with eye-blinking or so. (National Health System, 2018)

EEG's have the real information of how the brain is working, but the main problem is that it has recently begun the investigation of what are the brain's responses to stimuli and disease. In order to make a diagnosis, medical staff locate the origin of the electrical activity, which is crucial to make a good diagnosis. Afterwards, they must locate abnormal waves, which can be found in the contradiction of paired channels, in other words, paired channels that point to opposite directions. If there is an abnormality, it can be caused by focal or generalize slowing, delirium, dementia, coma or anesthetic patterns among others yet to be investigated.

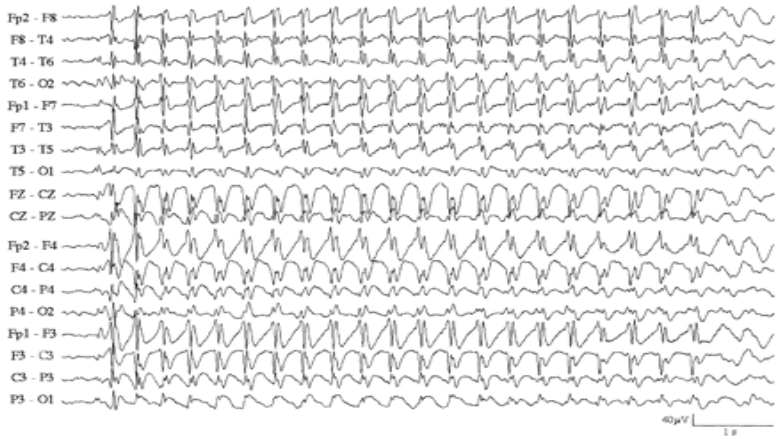


Figure 3: EEG with 18 channels

3.4. Comparison between Imaging Techniques

At certain times doctors must choose which of the tests is the most appropriate to diagnose the patient, and many times two or more tests can be used for the same purpose, so that extra-medical factors get involved. The following table expresses the relationship of each of the tests applied to the brain according to each of these factors.

	EEG	fMRI	CT Scan
Cost	Low	High	Medium
Expertise Needed	Medium	High	High
Radiation is Used	NO	NO	YES
Portable	YES	NO	NO
Duration of the Test	DEPENDS ON TEST	30-40 minutes	20 minutes
Temporal Resolution	High	Low	Low
Spatial Resolution	Low	High	High

Table 1: Comparison between EEG, fMRI and CT Scan.

In some cases, it will prevail more than one, for example, in case of applying test to children, you must take into account that the duration is not excessive; or in people who will have to have many of these tests, we will be primary the radiation factor used.

The main difference between CT and fMRI are costs, radiation and accuracy. Computed tomography is cheaper, but uses radiation that at high levels could harm the patient. In addition, fMRI takes longer to complete, but has a better spatial resolution than CT.

Regarding the EEG, both fMRI and CT Scan are more expensive, need more experience to be used and is portable. The EEG is therefore a very useful brain diagnostic tool in extra medical factors. Now, reading EEGs is very complicated, because as explained above, they are investigating the typical

patterns of each disease, in addition to the fact that an EEG can be very confusing, even for experienced doctors. Broadly speaking, the EEG is the best for obtaining data from the brain in a cheap and fast way, but much more needs to be done on the patterns of each disease.

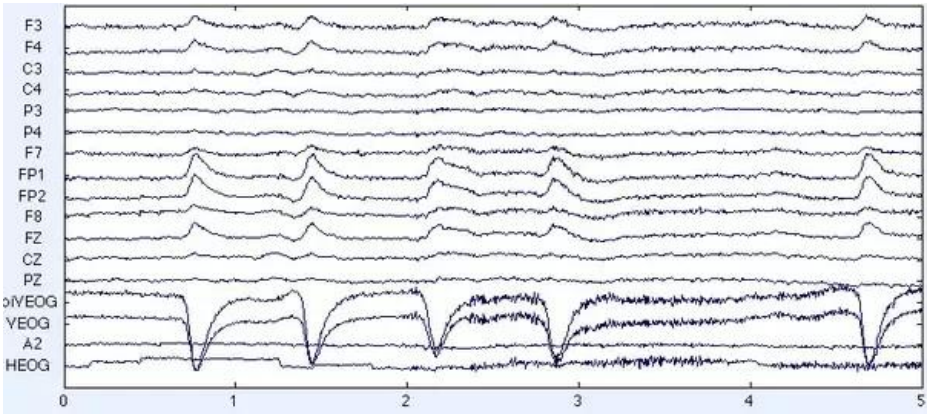


Figure 4: Blink Artifact in EEG test

4. SYSTEM ARCHITECTURE

As explained in the first chapter, schizophrenia cannot be diagnosed from the use of neuroimaging techniques. The objective of this project is to create a signal processing system which apply artificial intelligence which classify, with a high percentage of success, whether a patient has schizophrenia. For this, among all neuroimaging techniques, the one that stands out for cost, mobility and easy access to real data, is the EEG.

In order to achieve the objective of classifying patients, the EEG will be digitally processed. Then, features of each patient will be obtained and finally use different machine learning algorithms to find the one that gives the best result for our data.

An EEG can have an analog or digital output, in this case a digital output, that stores the voltage values for each channel, and that these are accessible, is needed. Nowadays the electroencephalographs are almost all digitals, and the most common output format is '*.edf*'. For this project it is essential to have data of schizophrenic people and healthy people, since from having two classes of EEG, it can be compared one against each other to find the characteristic differences.

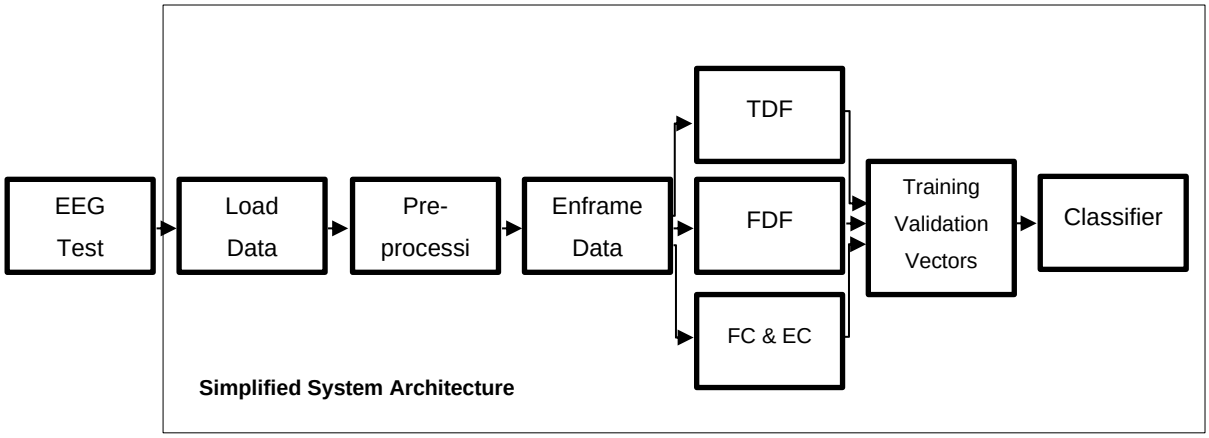


Figure 5: Simplified System Scheme

The program to be used is MATLAB and different libraries, functions and functionalities will be used, in order to process and classify the signal. In each of the phases, it will be explained what has been used and for what purpose. The diagram in Figure 5 shows a diagram of the different phases of the code. In each of the phases, EEG transformations are made in order to obtain a vector feature with which to make a model using the different Machine Learning algorithms, to then use that model and be able to predict whether a patient is Healthy or Schizophrenic.

For this last step, in which an external EEG is analysed, an interface has been programmed because the user who manipulates the program may not be a user with programming knowledge.
Everything used for the realization of the project is ordered as follows:

- *main.m*: This Matalb file contains the main structure. In the course of this function the different functions are called. To be able to access the data, functions and libraries of other folders, it has been used the addpath command at the beginning of the file.

- EDF_Files Folder: This folder contains the EEGs used for the experiment. They come from(Olejarczyk and Jernajczyk, 2017) and they are in *'.edf'* format.

- EEA_Files Folder: This folder contains the EEGs that were discarded for the experiment. They come from (M.V.Lomonosov Moscow State University, no date)and are in '.*eea'* format.

- myFunctions: This folder contains different functions used throughout the '*main.m'.*

- test Folder: In this folder you will find the different external tests that have been performed in order to check the functionality of the code before being added to the code already in operation. Not all files have ended up being included in the final code and this is due to different reasons such as that it may no longer be useful or may not work and an alternative would have been found.

- Toolboxes Folder: In this folder you will find the different Toolboxes that are used in the project. These Toolboxes are external to Matlab, since Matlab's own are installed next to the program.

  All used Toolboxes are:

  - Audio Component Analisys (ACA)
  - Audio Analysis Library
  - Audio Feature Extraction
  - Voicebox
  - Hermes

- Train Folder: In this folder are the classification models with the highest percentage of success that have been obtained. There is also a Word file, which shows the screenshots of the results obtained from each model.
- Intermediate Data Folder: This file contains the EEGs processed separately, before being introduced into the machine learning algorithms.

- Interface Folder: In this folder is the interface code created for the final user, in which a file is chosen and analysed, returning if the result is positive or negative.

- Features Functions and Dimensions.txt: This text file specifies the dimensions of each of the features obtained for each EEG.

5. LOAD DATA

To begin with the processing of data from the EEG, you must first have access to the data. Most digital EEGs conform to the European Data Format (EDF) which is a format created in 1992 and designed specifically for the storage of biological and physical signals that require multiple channels ((*European Data Format (EDF)*, no date). The EDF format consists of two main parts: header and data.

The header stores the information related to the EEG. Table 2 shows the description of the information stored in the different fields and their maximum length according to the standard(Kemp *et al.*, 1992).

Field	Description	Max Length
Ver	Version of the EDF format	8 ASCII
PatientID	Patient ID	80 ASCII
recordID	Recording ID	80 ASCII
Startdate	Start date of the recording in format dd.mm.yy	8 ASCII
Starttime	Start time of the recording in format hh.mm.ss	8 ASCII
Bytes	Bytes of the header	8 ASCII
Records	Number of data records	8 ASCII
Duration	Duration of the data record. Unit: seconds	8 ASCII
Ns	Number of signals (channels)	4 ASCII
Label	Label of the signal (normally the name of the electrode)	Ns*16 ASCII
Transducer	Transducer Type	Ns*80 ASCII
Units	Units of the records	Ns*8 ASCII
physicalMin	Physical minumun	Ns*8 ASCII
physicalMax	Physical maximun	Ns*8 ASCII
Prefilter	How has been the signal prefiltered	Ns*80 ASCII
Samples	Number od samples of the data record	Ns*8 ASCII
Frequency	Sample Frequency. Units: Hz	Ns*32 ASCII

Table 2: Description of the header fields with its maximum lengths.

It can be seen how from the number of signals the following data depends on ns. It is because they depend on the total number of signals that the EEG has.

The Data contains the discrete values of each signal that is being stored. In the case of EEG, it contains the signals received by the different electrodes. The total electrodes will be the total number of channels.

Not every electrode has to be in the head. For example, in some types of EEG explained in Section 3.3 the electrodes should be placed on facial muscles to detect patient blinking. Unlike the Header, the Data has no size limitation so that all the necessary signals can be stored, with necessary length, for what each patient requires.

In order to access the data of an EDF file, the '*edfReader'* function, available in Mathworks, has been used, which requires the name of the file as input. This function automatically saves the Header and Data variables in the Workspace.

In addition to the EDF format, throughout the investigation another format has been found that is also used to store EEG information. The format is EEA and unlike EDF it is only Data, in other words, it does not have the specifications stored in the file itself. In order to obtain a complete tool, EEA format has been used, which to be able to import the '*importdata'* function must be used and subsequently modify the number of rows and columns, according to the samples and channels available to the EEG

6. PRE-PROCESSING

Signal pre-processing is a key factor in the development of a Machine Learning model, since from how clean the data is, the model will know how to classify better or worse. In this phase of the project, once we have imported the original signal, certain corrections should be applied in order to:

- Give a common format for all samples
- Modify data that may lead to classifying errors
- Remove invalid data

For the application of ML in image classification, it is essential to regulate the size of the images, so that they are enlarged or reduced to a common size. In the case of EEG, the signal does not depend on pixels, but depends on the duration of the test and the number of channels. The time that the test lasts can be longer or shorter depending on the test and in case of cutting the signal in a certain time, it could incur a significant loss of data, making it a variable that we cannot limit. In the case of the number of channels it is necessary to limit it, because the values that each electrode produces are going to be evaluated. It is also important that the electrodes are placed in the same order as the EEGs with which the model has been trained.

To simplify the cases, the data will also be enframed in a certain order. This is because we have a continuous signal and we must divide it into frames. The way to divide the signal into frames depends on the experiment, but in all cases of pre-processing for the model, the same must be done for the cases to be classified as what was done with the training data.

To perform the framing of the data, a Mathworks function will be used. This function involves modifying the signal so that the signal is divided into overlapping frames every certain length. This function greatly reduces the number of samples that are nearby, so it does not lead to a loss of relevant data.

As explained above, EEGs measure electrical signals in the brain. And since all electrical signals have a frequency, it can be differentiated by delta (1-3Hz), theta (3.1-7.9Hz), alpha (8-13Hz), beta (13.1-29.9Hz) and gamma (30-100Hz) waves. In some studies, as in (Bougou *et al.*, 2020)during signal processing, only delta and theta bands are used. The justification is that (Koenig *et al.*, 2001) showed that there is a significant difference activity in those bands and (Newson and Thiagarajan, 2019) showed that there is an increase of lower frequencies in psychiatric patients.

In this project, all frequencies will be processed, because the exact frequency of the characteristic features of schizophrenia is unknown. Once the pre-processing is finished, the result is a signal that has been improved through the enframe function and a band-pass filter. The signal obtained after this step is the signal that will be used in the following steps, which are obtaining features through different methods.

7. TIME DOMAIN FEATURES

Looking at the main scheme of the project, this phase is in parallel with Frequency Domain and Hermes features. In this section the time domain features will be explained. Time domain is a term to classify mathematical functions, signals or statistical data whose independent variable is time. In our case, the EEGs are in the time domain, since it is micro volts with respect to time.

The different functions that are applied to the signal provide us with new data with the main objective that the differences between positive and negative EEGs increase. This does not always happen, in certain cases the differences decrease, but that will be handled later by the AI, which through the different algorithms will give more weight to some than to others. In the experiment, characteristics will be compared with each other to see which of the characteristics is most different, in order to reduce the processing time to a great extent.

The formulas applied to the signal obtained at Section 6 are described below.

7.1. Minimum

The minimum function returns the minimum value of each of the frames of our signal. The implemented function is “min ()” and it is a MATLAB built-in function, so it does not depend on any external toolbox. The way in which the minimum value of each frame is obtained is by comparing the values with each other.

The function input is the frame matrix obtained in the pre-processing step and the output is a vector of one row for the number of EEG channels.

7.2. Maximum

The maximum function returns the maximum value of each of the frames of our signal. The implemented function is “max ()” and it is a MATLAB built-in function, so it does not depend on any external toolbox. The way in which the maximum value of each frame is obtained is by comparing the values with each other.

The function input is the frame matrix obtained in the pre-processing step and the output is a vector of one row and the number of EEG channels as columns.

7.3. Standard Deviation

The standard deviation is a measure of how the values are distributed from the mean. This value can be calculated mathematically using (Eq. 1). A low result will mean that the sample values are close to the mean. In the other hand, a high value means that the values are widely distributed relative to the mean.

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \quad \textbf{(Eq. 1)}$$

In the standard deviation formula:

- N is the number of values in the sample.
- $\overline{x}$ is the mean value of all the values in the sample.
- x_i are the individual values of the sample.

MATLAB has its own function for the standard Deviation, to which the data matrix will be introduced, and the output will be one row and the number of EEG channels as columns.

7.4. Quartile

Quartiles are values that divide a data sample into four equal parts. Using quartiles, you can quickly assess the dispersion and central tendency of a data set.

The first quartile (Q1) represents that 25% of the sample data is less than or equal to the value of Q1. The second quartile, also called the median mark that 50% of the data is less than or equal to the value of Q2. Finally, the third quartile, represents that 75% of the data is less than or equal to that value.

MATLAB has its own function for quartiles, to which the data matrix will be introduced, along with the percentages of each quartile, and the output will be three rows and the number of EEG channels as columns.

7.5. Percentile

Percentiles are a statistical measure that indicates the value of the variable below which a percentage is found. For example, the twenty percentile is the value below which 20% of the observations are found. The percentiles and quartiles are similar, since a percentile-25 (P25) is the same as Q1. In the

experiment values other than twenty-five, fifty and seventy-five because those are the values that are calculated in the quartiles.

MATLAB has its own function for percentiles, to which the data matrix will be introduced, along with the percentiles, and the output will be as many rows as percentiles are desired to obtain and the number of EEG channels as columns.

7.6. Zero Crossing Rate

The Zero Crossing rate is the rate at which the signal changes sign in each frame. In other words, it is the rate of the times it crosses the zero value. This feature is widely used in voice recognition to classify percussion sounds, so it can be a very good feature for EEGs.

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{\mathbb{R}_{<0}}(s_t s_t - 1); \quad \textbf{(Eq. 2)}$$

where s is the signal with length T and $1_{\mathbb{R}_{<0}}$ it is an indicator of positive or negative.

The (Eq. 2) is implemented in a code found in the "*myFunctions*" folder. The function input in MATLAB is the pre-processed signal, and the output is a one row and the number of EEG channels as columns.

7.7. Energy

According to the definition the energy of a continuous signal is defined by the area under the curve of the squared signal (Medina, 2012) is previously stated, the framed signal is a discrete signal. The equivalent of the integral in a discrete signal is the sum.

$$Es = \sum_{n=-\infty}^{\infty} |x(n)|^2 \quad \textbf{(Eq. 3)}$$

The (Eq. 3) therefore represents the energy of the signal, which depends solely on the signal, in this case each frame and n which is each sample of the frame.

The implementation in MATLAB does not require a separate function since it is a simple formula that can be written on a single line. The output of the function is one row and the number of EEG channels as columns.

7.8. Band Energy

The section 7.7, explained how the energy of the complete signal is calculated. The next feature to calculate is the energy of each of the frequency bands. For this, the filtered signal must be obtained in each of the bands, and then calculate the energy using the formula (Eq. 3)

To obtain the filtered signal in each of the bands, a bandpass filter must be created with the frequencies of Delta, Theta, Alpha, Beta and Gamma. The filter to be used is a Chebyshev type II order, with a passband ripple (Rp) equal to 10, and an attenuation of the suspension band (Rs) equal to 40. Once the signal has been filtered in the different bands.

The created function calculates the five bands in the same function call. The input is the signal obtained after pre-processing and the Nyquist frequency. The output are five vectors of one row and the number of EEG channels as columns.

7.9. Band Power

The power of a signal is not limited in time, unlike energy. In the case of signal energy, it can be zero, such as a sine function. The power (Eq. 4) provides a measurement with respect to time. This feature will only be calculated in the bands, because a certain energy band may be zero, but it is highly unlikely that the signal energy is zero. To implement band power, a MATLAB function is used, which has as inputs the signal, the sampling frequency and the frequency band. The result is a one row and the number of EEG channels as columns.

$$P = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=-\infty}^{\infty} |x(n)|^2 ; \qquad \textbf{(Eq. 4)}$$

where N are the total number of values, and x(n) the individual values.

8. FREQUENCY DOMAIN FEATURES

The Frequency Domain Features are the characteristics that are obtained through mathematical formulas that depend on the frequency. In the previous chapter the different formulas regarding the time that have been used in the project have been explained. The formulas implemented in the code to obtain the features in the frequency domain are explained below.

8.1. Spectral Magnitude

In the frequency domain, the signal must be according to frequency. As previously discussed, an EEG is a signal in time so a time-frequency base change must be made. This change of base is made by applying the Fourier transform (Eq. 5).

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-iwt} dt\,; \qquad \textbf{(Eq. 5)}$$

where w is f (w) and f (t) are the signal with according to frequency and time, respectively.

Explained in a simple way, the Fourier transform changes the original signal for a set of sines. The result of applying the Fourier transform on a real signal is a complex signal, since all possible inputs must be able to be represented both in phase and in amplitude. In this case, the phase of the sines is what makes the signal complex. Since working with complex numbers is always problematic, it has been decided to only work with amplitude. In this case, it is magnitude, since what it is obtained from each frame is a scalar, and not a vector.

For the application of the spectral magnitude in MATLAB, the built-in function *'fft()'* has been used. This function performs the Fourier transform. After the FFT the function '*abs()*', with which the phase is eliminated of the result of the transform. This feature is calculated by inputting the signal after pre-processing, and the output is a matrix of 35150 rows times the number of channels as columns.

8.2. Periodogram

According to the definition, the periodogram (Eq. 6) is the squared Fourier transform divided by the total number of samples (N) (Smith, 2011). The periodogram is very similar to the Fourier transform, but is optimized for unevenly sampled data. Periodograms are used in many areas of knowledge, such as astronomy, for which it is useful for finding the periods of stars or planets (Plavchan, 2011). As far as the project is concerned, the periodogram can be useful in showing the repetition periods of each EEG channel.

$$P(w) = \frac{1}{N}|\sum_{n=0}^{N-1} x_n e^{-j2\pi wn}|^2; \quad \textbf{(Eq. 6)}$$

Where w is the frequency, N the total number of samples and n each individual sample.

This feature is calculated by inputting the signal after pre-processing, and the output is a matrix of 32769 rows times the number of channels of the EEG.

8.3. Centroid

The spectral centroid is one of the spectral descriptors available in the Audio Component Analysis (ACA) Toolbox. According to (*Spectral Descriptors*, 2014), the centroid is the frequency weighted sum normalized by the unweighted sum. The formula (Eq. 7) is applied in the 'FeatureSpectralCentroid ()' function that has been used in the code.

$$Centroid = \frac{\sum_{k=b1}^{b2} f_k s_k}{\sum_{k=b1}^{b2} s_k}; \quad \textbf{(Eq. 7)}$$

where f_k is the frequency in Hz of the frame k, s_k is the spectral value of the frame k and $b1\ and\ b2$ are the edges of the frame.

The centroid or geometric centre is the average position of all points in the coordinate directions (Protter and Morrey, 1977). In the project, the centroid is used as the centre of gravity between the points of the frame. The output of the spectral centroid is a vector of one row time the number of channels of the EEG.

8.4. Crest

The spectral crest (Eq. 8) is one of the spectral descriptors available in the Audio Component Analysis (ACA) Toolbox. The spectral crest is the ratio between the maximum spectrum and the arithmetic mean of the spectrum.(Peeters, 2014)

$$crest = \frac{max(s_{k[b1,b2]})}{\frac{1}{b2-b1}\sum_{k=b1}^{b2} s_k}; \quad \textbf{(Eq. 8)}$$

Where s_k is the spectral value of the frame k and $b1\ and\ b2$ are the edges of the frame.

The spectral crest is used to find the peak of the spectrum. In audio processing, the higher the crest, the more tonality. In the project this feature is useful for comparing the differences between the

frequency (tone) peaks of the EEG electrical signals. The output of the spectral crest is a vector of one row times the number of channels of the EEG.

8.5. Decrease

The Spectral decrease (Eq. 9) represents how much the spectrum decreases, paying special attention to the lower frequencies. This feature is very useful for instrument recognition, since each instrument has a different decrease in spectral amplitude. The output of the spectral decrease is a vector of one row times the number of channels of the EEG.

$$decrease = \frac{\sum_{k=b1+1}^{b2} \frac{s_k - s_{b1}}{k-1}}{\sum_{k=b1+1}^{b2} s_k}; \quad \textbf{(Eq. 9)}$$

Where s_k is the spectral value of the frame k and $b1\ and\ b2$ are the edges of the frame.

8.6. Flatness

The Spectral Flatness (Eq. 10) measures the ratio between the geometric mean of the spectrum and its arithmetic mean. It is commonly used in audio processing as a measure of noise. The Spectral Flatness formula is implemented in the '*FeatureSpectralFlatness'* function that is available in the Audio Component Analysis (ACA) Toolbox.

$$flatness = \frac{(\prod_{k=b1}^{b2} s_k)^{\frac{1}{b2-b1}}}{\frac{1}{b2-b1}\sum_{k=b1}^{b2} s_k}; \quad \textbf{(Eq. 10)}$$

Where s_k is the spectral value of the frame k and $b1\ and\ b2$ are the edges of the frame.

This feature is commonly applied to speech recognition (Lehner, Widmer and Sonnleitner, 2014).The output of the function is a vector of one row of complex numbers rows times the number of channels of the EEG. As done before, the complex component is eliminated, and as a result, only the real part of the output is obtained.

8.7. Flux

Spectral flux (Eq. 11) measures how the spectrum varies with time. It is commonly used in onset detection (Dixon, 2006) and audio segmentation (Tzanetakis and Cook, 1999), so in the project it will be useful to find the way in which the spectrum varies in each of the frames. The output of the function is a vector of one row times the number of channels of the EEG.

$$flux(t) = (\sum_{k=b1}^{b2} |s_k(t) - s_k(t-1)|^p)^{\frac{1}{p}}; \quad \textbf{(Eq. 11)}$$

Where s_k is the spectral value of the frame k, $b1\ and\ b2$ are the edges of the frame and p is the norm type.

8.8. Kurtosis

The Spectral Kurtosis is a statistical function that indicates the presence of series of transients in the frequency domain (Antoni, 2006). The function used, found in the Audio Component Analysis (ACA) Toolbox, follows (Eq. 12) calculated using the fourth central moment. A statistical moment is the probability of distribution of a random variable with respect to the mean. The third and fourth moments are used for the standardization of the variables and are used in sections 8.8 and 8.11. In the case of Spectral Kurtosis, the fourth moment is used. The output of the function is a vector of one row times the number of channels of the EEG.

$$Kurtosis = \frac{\sum_{k=b1}^{b2}(f_k - \mu_1)^4 s_k}{{\mu_2}^4 \sum_{k=b1}^{b2} s_k}; \quad \textbf{(Eq. 12)}$$

Where:

- f_k is the frequency in Hz of the frame.
- s_k is the spectral value of the frame.
- $b1\ and\ b2$ are the edges of the frame.
- μ_1 is the spectral centroid.
- $\mu_2$is the spectral spread.

8.9. Mel Frequency Cepstral Coefficients (MFCCs)

The MFFC is a representation of the spectrum using several different coefficients (Peeters, 2014). The Cepstrum is the Fourier transform of the logarithm of the signal spectrum, and the Mel-Cepstrum is the cepstrum calculated in the Mel bands instead of the Fourier spectrum. The Mel bands differ from the Fourier spectrum in the spacing between the bands. In Mel's, they are arranged in a way that is closer to how humans hear. To calculate the MFCCs, the code follows the steps in the Figure 6.

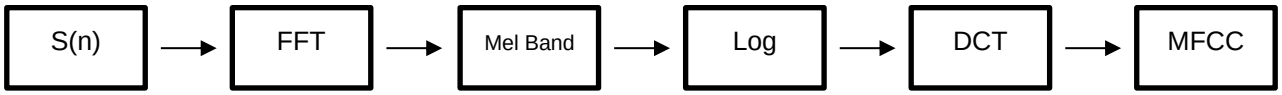


Figure 6: Steps to calculate MFCCs

Esta feature es utilizada para reconocimiento de y es útil para el proyecto debido a que podría reconocer anomalías en las canales del EEG. The output of the function es thirteen rows, which are the different coefficients, and the number of EEG channels as columns.
This feature is used for speech recognition (Vergin, O'Shaughnessy and Farhat, 1999) and is useful for the project because it could recognize anomalies in EEG channels. The output of the function is a matrix of thirteen rows, which are the different coefficients, times the number of channels of the EEG.

8.10. Roll-off

In the time domain, the percentile and quartile have been explained in sections 7.4 and 7.5.These are the values by which a certain percentage of the signal falls below that value. The Spectral Roll-off is similar to the percentile and quartile so that the result of applying the (Eq. 13) is the frequency by which 85% of the signal falls below that frequency. The output of the function is a vector of one row times the number of channels of the EEG.

$$Rolloff = i\ ;\ such\ that\ \sum_{k=b1}^{i} |s_k| = 0.85 \sum_{k=b1}^{b2} s_k\ ; \qquad \textbf{(Eq. 13)}$$

Where s_k is the spectral value of the frame k and $b1\ and\ b2$ are the edges of the frame.

8.11. Skewness

Spectral skewness (Eq. 14) is a measure of the asymmetry of a distribution with respect to its centroid. As in Kurtosis, use the statistical moments to calculate the probability of distribution, but in this case, the centroid and the third moment are used instead of the mean and the fourth moment.

$$Skewness = \frac{\sum_{k=b1}^{b2}(f_k - \mu_1)^3 s_k}{{\mu_2}^3 \sum_{k=b1}^{b2} s_k}; \quad \textbf{(Eq. 14)}$$

Where:

- f_k is the frequency in Hz of the frame.
- s_k is the spectral value of the frame.
- $b1\ and\ b2$ are the edges of the frame.
- μ_1 is the spectral centroid.
- $\mu_2$is the spectral spread.

The output of the function is a vector of one row times the number of channels of the EEG.

8.12. Slope

The Spectral Slope (Eq. 15) is the coefficient of the imaginary line between the first and last value of the spectral amplitude of the frame. Spectral Slope is commonly used in speech analysis, specifically in recognition of the speaker(Murthy *et al.*, 1999). The output of the function is a vector of one row times the number of channels of the EEG.

$$Slope = \frac{\sum_{k=b1}^{b2}(f_k - \mu_f)(s_k - \mu_a)}{\sum_{k=b1}^{b2}(f_k - \mu_f)^2}; \quad \textbf{(Eq. 15)}$$

Where:

- f_k is the frequency in Hz of the frame.
- s_k is the spectral value of the frame.
- $b1\ and\ b2$ are the edges of the frame.
- μ_s is the mean spectral value.
- $\mu_f$is the mean frequency.

8.13. Spread

The Spectral Spread (Eq. 16) is the measure between the standard deviation and the centroid. The standard deviation is a measure of how the values are distributed from the mean and the centroid is the frequency weighted sum normalized by the unweighted sum. This ratio gives us the instantaneous value of the bandwidth of the spectrum. The output of the function is a vector of one row times the number of channels of the EEG.

$$Spread = \sqrt{\frac{\sum_{k=b1}^{b2}(f_k - \mu_f)^2}{\sum_{k=b1}^{b2} s_k}} \quad \textbf{(Eq. 16)}$$

Where:

- f_k is the frequency in Hz of the frame.
- s_k is the spectral value of the frame.
- $b1\ and\ b2$ are the edges of the frame.
- μ_1 is the spectral centroid.

8.14. Tonal Power Ratio

The tonal power ratio measures the tone of each frame according to the power of the spectrum(Lerch, 2012). The formula shows an arbitrary definition of tonal energy. Follows the relation of the higher the ratio, the greater the tonality of the signal. This feature is another measure of characterizing the signal in frequency. The function is in the Audio Component Analysis (ACA) Toolbox and the is a vector of one row times the number of channels of the EEG.

8.15. Pitch Chroma

The Harmonic Pitch Class profiles are a group of characteristics from a signal, which represent a profile of the tonality of the frame. This profile is divided into 12 tone classes, which are called chroma. The Spectral Pitch Chroma is used for chord recognition (Fujishima, 1999) and in this project, it defines the tonality for each of the EEG channels. The function is in the Audio Component Analysis (ACA) Toolbox and the output of the function is a vector of one row times the number of channels of the EEG.

8.16. Harmonic Product Spectrum (HPS)

The Harmonic Product Spectrum measures the maximum coincidence of the harmonics with respect to the spectrum from the (Eq. 17). This obtains a periodic correlation, from which the maximum is seek, which is the fundamental frequency estimate of the frame.

$$Y(w) = \prod_{r=1}^{R} |X(wr)|; \quad \textbf{(Eq. 17)}$$

Where R is the number of harmonics.

The function is in the Audio Component Analysis (ACA) Toolbox and the output of the function is a vector of one row times the number of channels of the EEG.

9. FUNCTIONAL AND EFFECTIVE FEATURES

Brain research has increased significantly in recent years. One of the last great contributions to the brain has been connectivity, which defines different organizational forms of the brain that are associated with each other (Horwitz, 2003). This distribution of the brain consists of three categories:

- Structural connectivity (SC)
  It is based on the physical connectivity of the brain through the set of neurons. Its distribution is very similar to the anatomical distribution of the brain.

- Functional Connectivity (FC)
  It is based on statistical connectivity between brain signals derived from two or more different brain units. The FC does not require the existence of the physical signal between the brain networks.

- Effective Connectivity (EC)
  It is based on casual interactions between different brain units. The causal relationship held in the EC, unlike the FC, and there can be observed in the physical signals.

In the project, some of the most used FC and EC features will be obtained, using the HERMES Toolbox. This toolbox was created by the Polytechnic University of Madrid for the environment of MATLAB. Although many others have been evaluated, such as EEGLAB or FCLAB, HERMES is the one with the most documentation as well as accessibility to the code.

HERMES toolbox is intended to be a user interface, but it is also opened to researchers in order to further study the human brain. The indices or features to be obtained in this section are divided into four categories: Classic Measures (CM), Phase Synchronization (PS), Generalized Synchronization (GS) and Information Theory (IT). Each of these categories will be a single function in which the features are calculated. Each group of features characterizes the pre-processed EEG signal. After calculating the different features, statistical measures are applied in order to obtain better results in a smaller number of samples.

To calculate these features, a single call is made from the main function to the '*FunctionalAndEffective.m*' function, providing it with the variables of the pre-processed signal and the sampling frequency. In the '*FunctionalAndEffective.m*' function it is obtained the total number of channels of which the signal consists, and the samples of each of the channels. Afterwards, calls are made to each of the functions of each group of features, which return the statistical values, explained in point 9.5, of each of the features.

The code used, which is found in the '*myFunctions'* folder, has been modified to a great extent, because certain parameters and structures have to be created, since being a toolbox programmed for use by the user interface, these structures and parameters are obtained from different functions. These parameters and structures are found at the beginning of each of the individual functions.

9.1. Classic Measures

In the Classical Measures category are the most used methods for calculating FC. These features obtain linear dependencies of the EEG. The parameters that have been entered by default are the maximum number of lags for the cross-correlation (xCOR), which is equal to the number of samples divided by twenty.

9.1.1. Pearson's Correlation Coefficient (COR)

The linear dependence to be calculated with the COR is in the time domain, between two signals without lag. The interval of this function is between -1 and 1, where -1 is an inverse correlation and 1 a direct correlation.

$$COR = \frac{1}{N}\sum_{n=1}^{N} x(n)y(n)\,; \qquad \textbf{(Eq. 18)}$$

Where N is the total number of samples.

9.1.2. Cross-Correlation (xCOR)

COR and xCOR measure the linear correlation between two signals in the time domain. The difference between them is that COR measures it without any applied delay and xCOR sets a time delay parameter. The interval of the function is the same as that of the COR.

$$xCOR = \frac{1}{N-\tau}\sum_{n=1}^{N-\tau} x(n+\tau)y(n)\,; \qquad \textbf{(Eq. 19)}$$

Where N is the total number of samples and τ is the time delay.

9.1.3. Coherence (COH)

In the previous Classic Measures features, the correlation between the two signals in the time domain is calculated. In contrast, coherence is calculated in the frequency domain, using the ratio of the power spectral density squared (SPD) between the signals and the SPD of each of the signals.

$$COH_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)}; \quad \textbf{(Eq. 20)}$$

where S is the SPD of the indicated signals.

9.1.4. Imaginary Coherence (iCOH)

Imaginary Coherence measures, like COH, the linear correlation in function of frequency. The difference between the two is that the COH measures the real part, which is the magnitude, of the ratio between the SPD and the iCOH measures the imaginary part of the same ratio, which is the phase.(Sakellariou *et al.*, 2016)

9.2. Phase Synchronization

The first time the phase synchronization was used, it was as a coupling measure between two channels, and the phase locking condition was defined as (Rosenblum, Pikovsky and Kurths, 1996):

$$\Delta\phi(t) = \left|\phi_x(t) - \phi_y(t)\right| < cte; \quad \textbf{(Eq. 21)}$$

Where $\phi(t)$ is the phase.

The phase difference used as a reference is always in the interval [0.2π). To obtain the different PS indexes, another pre-processing must be done before the signal, which is only applicable for this section. This second signal processing is based on obtaining the analytical signal from each of the EEG channels. The analytical signal is made up of a real part, which is the value of the channel itself; and an imaginary part, which is obtained by making the Hilbert transform to the channels. To finish, the formulas (Eq. 22) and (Eq. 23) must be applied to obtain the instantaneous amplitude and phase.

$$A(t) = \sqrt{x_H(t)^2 + x(t)^2}; \quad \textbf{(Eq. 22)}$$

$$\phi_x(t) = arctg\frac{x_H(t)}{x(t)}; \quad \textbf{(Eq. 23)}$$

Where x_H is the imaginary part of the analytic signal, and x is the real part.

9.2.1. Phase Locking Value (PLV)

PLV is a measure of the phase value in the unit circle. For its calculation it only requires the relative phase difference. The PLV interval is between zero and one, with zero being a uniformly distributed phase difference value and one being a constant phase difference value. This feature is ineffective in the presence of common sources.

$$PLV = \frac{1}{N}\sum_{n=1}^{N} e^{i\Delta\phi_{ref}(t_n)}; \quad \textbf{(Eq. 24)}$$

Where N is the total number of observations.

9.2.2. Phase Locking Index (PLI)

Unlike PLV, which is not very robust against common sources, PLI is, and it is achieved by discarding phase distributions that are close to zero with a pi module. PLI and PLV are complementary features, since one can detect what the other does not detect. PLIs problem is that the noise sensitivity is reduced due to discontinuity.

The index has as a interval (0,1), where 0 is that there is no coupling in the phase difference, and 1 is that there is coupling in the phase difference (from 0 mod pi).

$$PLI = |\frac{1}{N}\sum_{n=1}^{N} sign(\Delta\phi_{ref}(t_n))|; \quad \textbf{(Eq. 25)}$$

Where N is the total number of observations.

9.2.3. Weighted PLI (wPLI)

In order to parameterize the phase differences with the smallest possible error, wPLI improves the main problem of PLI, which is to distinguish the relative phase that is close to zero mod pi. The way to solve PLIs problem is not to discontinue the values by eliminating the phase and the amplitude of the imaginary component of cross spectrum. The result, always within the interval (0,1), means that there is no synchronization, if it is 0, and that there is synchronization if it is 1.

Unlike the rest of the PS features, wPLI uses both the phase and the amplitude of the analytical signals obtained in the processing carried out for PS.

9.2.4. ρ Index (RHO)

Rho index is a measure based on the Shannon entropy. This feature indicates the degree of phase distribution and therefore the degree of synchronization. It is calculated through the difference between the relative phase distribution and the uniform distribution. The interval is (0,1), zero being a uniform distribution, which means that there is no synchronization, and one, a Dirac distribution, which means that there is total synchronization.

$$\rho = \frac{S_{max} - S}{S_{max}}; \quad \textbf{(Eq. 26)}$$

where $S_{max}$is the maximun entropy of the distribution and S is the entropy of the $\Delta\phi_{ref}$

9.2.5. DPI

In all the PS features explained above, the relative phase distribution is used in some way to calculate the index. However, DPI uses the temporal evolution of the phase to calculate the directionality of the phase(Rosenblum *et al.*, 2002).

The directionality is calculated through the analysis of the degree of coupling between the two channels since a phase increase is obtained that can be modelled through periodic functions of both phases. The calculation model to obtain the directionality value is the EMA (Rosenblum *et al.*, 2002). Other models, such as the IPA model (Rosenblum *et al.*, 2002), have been studied, but have been ruled out due to the technical complexity of the code.

9.3. Generalized Synchronization (GS)

The set of features in the GS category evaluates the relationship between two brain signals, which are dynamic, and one is a function of the other, according to (Eq. 27). The existence of GS leads to the conclusion that similar states tend to occur at similar times. In other words, from one state of a signal the state of the other signal can be known to some degree.

$$y(n) = f\big(x(n)\big); \quad \textbf{(Eq. 27)}$$

Where y(n) and x(n) are different brain channels.

As in PS, a pre-processing of the signal is required, and will only be used for the GS indices. The first step is to obtain, from each individual signal, a phase-space difference vector (Eq. 28). The Euclidian mean distance of the independent signal is then calculated according to (Eq. 29). The Euclidean mean distance from the other signal is obtained with the same formula, establishing the relation condition, which is that the value of the neighbours of x (n) is the same as that of y (n) (Eq. 30). Finally, (Eq. 31) is calculated, which is the value of the radius of reconstructed X.

$$x_n = [x(n), x(n-\tau) \ldots., x(n-(d-1)\tau)]; \quad \textbf{(Eq. 28)}$$
$$y_n = [y(n), y(n-\tau) \ldots., y(n-(d-1)\tau)];$$

Where d is the embedding dimension, which a constant in the project, and τ the time delay.

$$R_n^{(k)}(X) = \frac{1}{k}\sum_{j=1}^{k}\left|x_n - x_{r_{nj}}\right|^2; \quad \textbf{(Eq. 29)}$$

Where $x_{r_{nj}}$ is the k nearest neighbours of x_n.

$$R_n^{(k)}(X|Y) = \frac{1}{k}\sum_{j=1}^{k}\left|x_n - x_{s_{nj}}\right|^2; \quad \textbf{(Eq. 30)}$$

Where $x_{s_{nj}}$ is the k nearest neighbours of y_n.

$$R_n(X) = \frac{1}{N-1}\sum_{\substack{j=1 \\ j\neq 1}}^{k}\left|x_n - x_j\right|^2; \quad \textbf{(Eq. 31)}$$

Where x_j is the k nearest neighbours.

GS features are calculated through indexes that correlate the previous distances. The indices are purely mathematical, and their practical use is explained in each of the sections.

9.3.1. S

This index is described in (Arnhold *et al.*, 1999). Its interval is (0.1], zero being a value indicating the independence between the two EEG channels, and one being full dependence This feature is affected by signal noise and signal length. S index is calculated using (Eq. 32).

$$S^{(k)}(X|Y) = \frac{1}{N}\sum_{n=1}^{N}\frac{R_n^{(k)}(X)}{R_n^{(k)}(X|Y)}; \quad \textbf{(Eq. 32)}$$

9.3.2. H

This index is described in (Arnhold *et al.*, 1999). Its interval is (0, ∞), with zero being a value that suggests independence between the two EEG channels, and infinity, a full dependency. Unlike feature S, H is not standardized, and this makes it less affected by signal noise. It is calculated using (Eq. 33).

$$H^{(k)}(X|Y) = \frac{1}{N}\sum_{n=1}^{N}\frac{log(R_n(X))}{R_n^{(k)}(X|Y)}; \quad \textbf{(Eq. 33)}$$

9.3.3. N

This index is described in (Quian Quiroga *et al.*, 2002). Its interval is [0,1), with zero being a value that indicates the independence between the two EEG channels, and one, a synchronization of the channels. This feature is not affected by noise as much as S index, and it is normalized. It is calculated using (Eq. 34)

$$N^{(k)}(X|Y) = \frac{1}{N}\sum_{n=1}^{N}\frac{R_n(X) - R_n^{(k)}(X|Y)}{R_n(X)}; \quad \textbf{(Eq. 34)}$$

9.3.4. M

The M index defined in (Andrzejak *et al.*, 2003) is another way to normalize the distances between the two signals. The interval of M is [0,1], and like the index N, zero means an independence between the signals and one, a total synchronization.

$$N^{(k)}(X|Y) = \frac{1}{N}\sum_{n=1}^{N}\frac{R_n(X) - R_n^{(k)}(X|Y)}{R_n(X) - R_n^{(k)}(X)}; \quad \textbf{(Eq. 35)}$$

9.3.5. L

The L index, defined in (Chicharro and Andrzejak, 2009), is calculated through the distance between ranks of the phase-space vectors. Its interval is [0,1], having the same meaning as M and N. The L index improves the detection of directionality with respect to the rest of the GS features.

$$L^{(k)}(X|Y) = \frac{1}{N}\sum_{n=1}^{N}\frac{G_n(X) - G_n^{(k)}(X|Y)}{G_n(X) - G_n^{(k)}(X)}; \quad \textbf{(Eq. 36)}$$

The average rank of X, $G_n(X)$, is N divided by 2, and the minimal average rank is the number of nearest neighbours, k, plus one, divided by two.

9.4. Information Theory (IT)

The features included in the Information Theory (IT) category are measures of the quantification of the information from a random variable, obtained through the Shannon entropy. In IT there are two main measures, Mutual Information (MI) and Transfer Entropy (TE) and two secondary ones that are obtained from the partial measures of the main ones. All four features are FC measures.

In order to adjust the original code to this project, major modifications have had to be made and two variables have been established as constants. These constants are the Embedding dimension and the Time delay.

9.4.1. Mutual Information (MI)

Mutual Information specifically measures the amount of information shared by two EEG channels. Shannon entropy is essential for calculating MI since the information shared between the two channels is at the total intersection of the set. To calculate MI, the (Eq. 37) must be applied. The MI interval is [0, ∞), where zero means that the channels are independent and infinite, that they are dependent.

$$MI_{xy} = \sum\nolimits_i p(x,y) log \frac{p(x,y)}{p(x)p(y)} \quad \textbf{(Eq. 37)}$$

Where p is the entropy.

9.4.2. Partial Mutual Information (PMI)

PMI is a feature produced from MI, which includes a new variable. This new variable, Z, provides a measure of whether the information shared between the two channels is shared in a direct or indirect way. In the case of MI, this variable was given as a constant equal to zero, which meant that the channels did not depend on whether the relationship was direct or indirect.

9.4.3. Transfer Entropy (TE)

TE is a feature that measures the flow of information from one channel to another. What distinguishes TE of other features within the Hermes toolbox is that TE assumes the existence of a dependency between the channels, and which is a nonparametric estimate, i.e. not depending on parameters on the distribution of the data. This quality of TE makes it a good feature in case of detecting abnormal and transformation resistant data. The main problem presented by TE is that for its calculation it is necessary to pre-calculate the probabilities of all the values of each channel, which takes a lot of computing time.

9.4.4. Partial Transfer Entropy (PTE)

Just as obtaining PMI from MI, it can also be obtained PTE from TE, using the same concept of including a new variable, Z. This variable aims to establish whether the flow of information between channels is direct or indirect. In case of Z = 0, the value of TE would be obtained.

9.5. Graph Theory Measures

After having calculated each of the features of each category, huge matrices are obtained, which make the computation time required to obtain a classifier model unfeasible. For this reason, a series of statistical measures will be applied to each one of the obtained features, used in (Bougou *et al.*, 2020).

To simplify the code, a function has been created that calculates all the statistical measurements for each feature, which is entered as input, and returns a matrix of ninety-seven rows and the number of channels as columns. This function is executed individually for each feature, at the end of each of the categories explained above.

The statistical measurements explained below are based on graph theory. This theory analyses brain connectivity data, through a mathematical network that can be summarize in nodes with links between them (Fallani *et al.*, 2010). In the project, the nodes are the EEG channels, and the mathematical network is summarized in a connectivity matrix, which contains the data calculated below.

9.5.1. Global and Local Efficiency

The efficiency of a network quantifies the exchange of information around the network. Global efficiency is the inverse average length of the shortest path within any network (Lazega, 1995). Local efficiency is global efficiency, but calculated around neighbouring nodes(Watts and Strogatz, 1998). These two measures are used to determine the cost of connections within the network.

9.5.2. Eigenvector and Betweenness Centrality

Centrality identifies the most important nodes in the network. Now, there are different ways of measuring how important each node is, so there are different measures of centrality. Eigenvector Centrality assigns a value to the nodes based on the connections it has, so if it has adjacent relevant nodes, the value goes up. This measure of centrality determines the importance of each node by the influence the node has on the network. The Betweenness Centrality determines the importance by the total number of connections of each node, so a node that is connected to much is more important than one that is connected only to another.(Sporns and Kötter, 2004)

9.5.3. Clustering Coefficient and Transitivity

Clustering Coefficient is a measure of the degree to which nodes in the network are grouped. This measure is applied in groups of three nodes and it is calculated how a node is connected to a group of three nodes. Transitivity measures the same as Clustering Coefficient, but in such a way that the obtained measure is the relation of a node, with a group of three nodes, through other groups of nodes.(Sporns and Kötter, 2004)

9.5.4. Modularity

Modularity is a measure of the degree of network density, such that a network is highly modular if it has clearly separated groups of nodes, and within each group, there are many connections between the nodes that make up that group of nodes.(Sporns and Kötter, 2004)

10. TRAINING AND VALIDATION

A classification model based on Machine Learning (ML) needs learning before it can classify autonomously. Training a model requires training and validation data.

Training data are the different features of what sort is intended. In the case of this project, the training matrix is made up of all the features that have been explained in chapters 7, 8 and 9. The training matrix has all the features of all the EEGs that are used to train the model.

The validation data is the result that the machine must reach through training. In other words, it is the result of each one of the data entered for the training. In this project, the validation data will be whether the test being analysed is Healthy or Schizophrenic.

From an external point of view, we clearly know that EEG is Healthy and which Schizophrenic, and this information must be transferred to a data vector, according to the number of features that are obtained in total. For this, the names of the EEGs in EDF format must be named so that the first letter is "h" or "s", if it is Healthy or Schizophrenic respectively. After that, it is recommended to follow a numbering, to identify each EEG. The developed code stores the name in a variable and, after calculating all the features, creates the vector of the form so that it assigns zeros or ones, if it is Healthy or Schizophrenic respectively.

In the case of the code that has been developed, for the next step, the training matrix and the validation vector must be entered in the same matrix.

11. CLASSIFIERS

The last step of the project, before being able to classify new EEGs, is to create an ML model from the data obtained in previous steps. In ML there are different ways to train a model. In this project the most common ones will be tested, and will be compared to obtain the best result. The most common algorithms are Support Vector Machine (SVM), Naive Bayes Classifier, K-nearest neighbours (KNN) and Decision Tree. These algorithms differ in the way of establishing the class to which each data belongs, and therefore each of the algorithms is better or worse depending on the distribution of the data.

11.1. Support Vector Machine (SVM)

SVM is a supervised learning algorithm that can be used for both classification and regression tasks. It is based on the search of a hyperplane that separates the data set, leaving each of the classes on each side.

In a two-dimensional data series, the hyperplane is considered a line dividing the data. Support vectors are formed by the points closest to the hyperplane. To obtain the hyperplane, the support vectors are essential, since the objective of the SVM is to find a hyperplane with the greatest possible margin between the hyperplane and the support vectors. Mathematically this is summarized by calculating the distances between the support vectors.

In the absence of a clear 2D hyperplane, which occurs when the data set is not linearly separable, the three-dimensional view is used. In this case, the kernelling method is applied, which is based on a non-linear classification of the data, so the line that will be seen in two dimensions will not be a straight line. In this case, the hyperplane becomes a three-dimensional plane.

When the hyperplane is created, the classification depends on which side the new data to classify is on. Six types of SVM algorithms have been used in this project, which vary in how the hyperplane is calculated.

Advantages	Disadvantages
Accuracy	Not useful for large datasets
Best algorithm on small and clear datasets	Less effective when noisy data

Table 3: Advantages and Disadvantages of using SVM

11.2. Naive Bayes Classifiers

Naive Bayes is one of the most common algorithms used for classification models. Like SVM it is a supervised learning algorithm, because it works with pre-tagged data. Naive Bayes relies on Bayes' theorem to classify the data. This algorithm is totally statistical, and its main characteristic is that it assumes conditional independence between the characteristics, so its mathematical calculation is reduced to only the probabilities of each characteristic by itself.

The way to calculate the probability of belonging to one class or another is based on the analysis of each of the features of the training data. From the features, it obtains the probability of each one independently and associates it with one of the classes. From there, Bayes' Theorem is applied, to find the probability: $P(class|features)$.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}; \quad \textbf{(Eq. 38)}$$

Where P (A) is the a priori probability, P (B | A) is the conditional probability, P (B) is the total probability, and P (A | B) is the probability a posteriori.

With an example it is always better understood. We want to create a model that classifies between lions and deer. Both classes share certain characteristics, such as "having four legs" or "hair", but there are others that differ, such as "horns" or "tail length". The algorithm will obtain probability values for each of these characteristics, from the training data set. This means that the probability of “having four legs” is 1, while the probability of “having horns” is the number of deer divided by the total number of samples. Once the independent probabilities of each of the characteristics have been established, a new data is introduced, for example, a dog, which has four legs, long tail and hair, but does not have horns. The probability a posteriori $P(deer|\ 4L, Long\ tail, Hair)$ is going to be less than $P(lion|\ 4L, Long\ tail, Hair)$, so “Lion” is the predicted class for a dog.

Advantages	Disadvantages
One of the fastest algorithms	Assume that all features are independent
Works well in classifying many classes	Need a big data set to be trained

Table 4: Advantages and Disadvantages of using Naive Bayes

11.3. K-Nearest Neighbors (KNN)

KNN is a supervised machine learning algorithm, which bases its operation on a voting system, in which voters are the k closest neighbours. Being k a value that defines the number of neighbours that must vote on the new data. The number k is defined by the developer and there is no predefined number, because for each model the optimal value of k must be found, since the distribution is different. This causes that for the optimization of this model a study of the k value must be carried out.

The way to classify a data in one class or another is based on the environment of the data. First, the value of k must be established, which defines the number of neighbours who have to “vote” on the new value. To know which are those neighbours, the Euclidean distance is calculated, and the k neighbours with the smallest distance are obtained. Once the closest neighbours have been established, it is calculated what percentage of neighbours are from each of the classes, the highest percentage being assigned to the new value.

The training phase is minimal, since it is always evaluated on a new value, and not between the training values. This makes the test phase require all the training data to be able to predict.

Advantages	Disadvantages
Fast and simple to implement	Needs lots of computational power because it needs all the training data to predict a new case
No need to build a model	High memory requirement

Table 5: Advantages and Disadvantages of using KNN.

11.4. Decision Tree

Tree diagrams have been in use for a long time. Its main function is that through each of the divisions, a smaller number of samples are represented. Therefore, as more divisions are made, it will be possible to establish with more probability what class each group of samples belongs to. A simple example is the classification of the animal kingdom, in which there are vertebrates and invertebrates. Within the vertebrate class, there are warm-blooded or cold-blooded animals. And so on until we reach each of the animal species.

In ML decision trees can be classification or regression. Classification trees are based on discrete division of the data, such as YES / NO. However, the regression trees have continuous data divisions, so the decision is made if a variable is greater or less than a number.

Regression trees have been used in this project, since the calculated features are continuous values. The ML algorithm uses recursive partitioning, which divides the data into subsets until the subsets are sufficiently homogeneous, or there are no more values to subdivide.

Advantages	Disadvantages
Extremely fast in the training and test phase	It depends heavily on the training data
Does not take into account irrelevant features	Large trees are difficult to interpret

Table 6: Advantages and Disadvantages of using Decision Tree

12. EXPERIMENTAL SETUP

This main objective of this project was established in creating a classification model using the MATLAB programming environment, which obtains the best percentage for some brain disease. This objective was established in the Project Outline, carried out before development began. Through research, it was defined that the brain disease to be classified should be schizophrenia due to the great impact it has on society nowadays. The program to create the model has been obtained through research, scientific curiosity and perseverance, therefore, to complete the objective set, a model with a high percentage of success must be obtained. For this, a simple experiment has been established, which is based on executing the program using the different classification algorithms explained in Chapter 11. Through the different possibilities that the algorithms offer, we will evaluate which of them is the best.

Before executing the program with the different algorithms, it is necessary to establish the different variables that the code has. To start the dataset chosen for the training. After an analysis through different databases available on the internet, the one used in(Olejarczyk and Jernajczyk, 2017), was selected. Using (M.V.Lomonosov Moscow State University, no date), from the Laboratory for Neurophysiology and Neuro-Computer Interfaces (LNNCI), was also positively valued for the use in (Bougou *et al.*, 2020). Olejarczyk's dataset is in EDF format, and the LNNCI dataset is in EEA format. The code made allows us to use any of the two datasets, although in the end the chosen one was the first because of its universal format.

The total number of tests in the dataset is 28, being 14 healthy and 14 schizophrenics. The dataset has 19 channels, which are distributed using the International 10/20 Standard (Morley Andrew Morley Hons and Hill, 2016). The duration of these EEGs is five seconds each, with an average of approximately 1250 samples per channel. So, each EEG has an average of approximately 23,750 samples.

In the signal pre-processing, the enframing is carried out with an overlap of 50%. In later steps in which the different features are obtained, the values used in the experiment can be found in APPENDIX C: MATLAB Code. The code is commented, making it easy to read.

To generate the ML classification models the MATLAB tool, Classification Learner, will be used. This native App is useful at all levels for the large number of editable variables that it offers. The matrix entered must contain the features of each of the EEGs and the validation for those EEGs. The validation chosen is a binary validation which means that '0' means Healthy and '1' means Schizophrenia.

Something very important when creating a machine learning model is to establish the way to test the trained model. There are two ways. The first and oldest is the holdout, which is based on choosing a percentage of the training data, which is reserved for testing. The second is the cross validation, which performs folds with different training and testing data in each fold. With this technique it is possible to use all the training data in addition to making as many tests as the number of folds. In the project, the

second form, the cross validation, has been chosen, since we do not have a large amount of data and this makes it take full advantage of it. The number of folds chosen has been five, which is the recommended.

The algorithms to be used are those explained in Chapter 11. These algorithms have sub-branches due different calculation methods are applied, but with the same basis. For SVM, Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian and Coarse Gaussian variations are used. These SVM variations are the different ways to calculate the distances between the support vectors and the hyperplane. In Naive Bayes, Gaussian Naive and Kernel will be used. The difference between these two classifiers is the distribution used. In KNN Fine, Medium, Coarse, Cosine, Cubic and Weighted will be used. The difference between these models are the distinctions and the way to calculate the closest neighbours. Due to what is explained in section 11.3, a study of the value of k should be made, so different values will be used to obtain the value that produces the minimum error. Finally, in Decision Trees, Fine, Medium and Coarse are used. The difference between these models is the maximum number of splits allowed.

In addition to the analysis between the results obtained, Principal component analysis (PCA) will also be applied to the best results in order to check whether the result improves. This statistical method simplifies the amount of data by eliminating through the simplification of several variables in one that is a combination of these. The set of new variables, called the main component, are orthogonal to each other. Due to the limitation of computing time, PCA will only be applied in the best results obtained from the algorithms without any statistical transformation.

Due to the long calculation time of each algorithm, it was not possible to carry out tests with the proper misclassification costs. This parameter is very useful in a model like the one that is intended to be created in this project, since it is estimated how much it costs to detect a false positive compared to a false negative. In the case of this project, it is worse to diagnose a schizophrenic patient as healthy than a healthy patient as schizophrenic. Using the parameter of misclassification cost, we could improve the percentage of misdiagnosed schizophrenics, getting worse percentage of healthy patients misdiagnosed. Therefore, this is a variable to adjust in future research, which not only depends on a random number, but it is necessary to study how bad it is to give a false positive compared to a false negative for this specific purpose.

The computers used for the experiment, property of the University of Hertfordshire, have 16GB of RAM. A total of four computers have been used, working day and night, a total of 20 days. The total number of days is large due to technical errors such as the inability to enter the laboratory during weekends and automatic shutdowns of the computer network, among others. For the acceleration of this whole process, the MATLAB Parallel Pool Toolbox has been used. The total number of parallel workers for each computer is 4, making a total of 16 workers at the same time.

In addition, other ways to speed up the process have been investigated, such as data processing in the cloud through MATLAB Parallel Server, with which all processing can be performed in Amazon Web Services, preconfiguring an EC2 server. With this method, although it is paid, it could be done much faster and from home. In addition, there are models of EC2 servers for MATLAB, so the only difficulty is to create the connection. This way has been discarded due to the need of a specific license that we didn´t have.

From the best results we have obtained the code to execute so as not to have to depend on Classification Learner, and the best of them is found in the main, to have a complete program. In addition, the best models will be obtained and with the best of them a user interface will be programmed, while the computers are processing the data. The user interface, which is not in the initial objectives, will give greater usability to the results of the project, specifically non-technical users, such as doctors or users at home.

To use the model for future predictions, using the interface or not, the data to be entered must be in the same format and the same channels as the data used for the input, that is, in EDF format and 19 channels. Channel distribution must be in accordance with Table 16 in APPENDIX B.

13. RESULTS

The distribution of training data is a very visual measure to determine how accurate the results can be. Figure 8 shows the relationship between the values of channel one (Fp2) and two (F8). The orange dots correspond to EEGs that are schizophrenic and the blue ones to healthy. There is a relatively evident separation between the data, however, by expanding the predominant blue area, it can be seen that there are also orange dots (Figure 7). The presence of the mixed values will cause an error in the data.

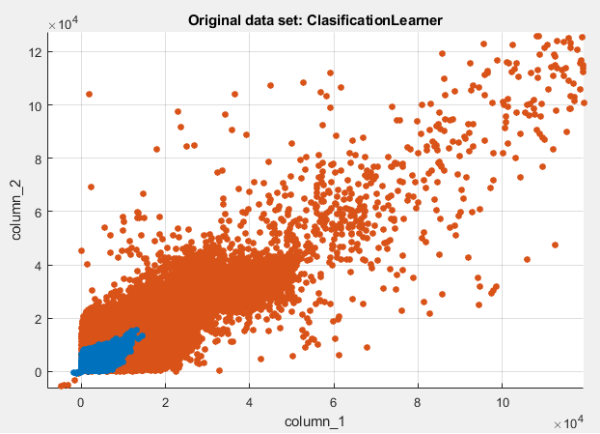


Figure 8: FP2 channel against F8 channel.

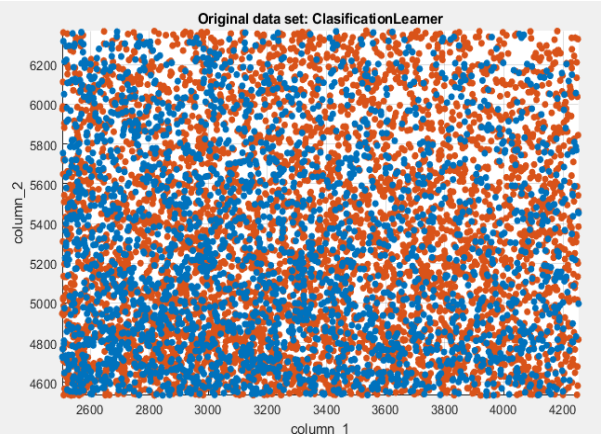


Figure 7: Zoomed dataset. FP2 against F8.

In the first phase of the experiment, that is, without applying PCA, and without analysing the value of k (in the case of KNN), a large difference is observed between the algorithms used. The best methods of each algorithm are shown in Table 7.

Algorithm	Method	% Accuracy
SVM	Quadratic	55.0
KNN	Weighted	82.7
Decision Tree	Fine	64.3
Naive Bayes	Kernel	59.7

Table 7: Best results for each algorithm.

Weighted KNN obtains the best accuracy with a difference of 18.4% on the Fine Decision Tree, which is the second best. Both SVM and Naive Bayes do not exceed 60%, so they can be considered bad results, due to the low effectiveness it has. Decision Tree with 64.3% could be improved through PCA and variations, but it is very unlikely that it will reach the values that are around the KNN methods. It is noteworthy that at this point of the experiment the best k value have not been studied, so KNN results could greatly improve.

For the next phase of the experiment, PCA is applied in the best results of each of the algorithms and their variation with respect to the same method without PCA. The results are shown in Table 8.

Algorithm	Method	% Accuracy	% Variation
SVM	Quadratic + PCA	51.7	-3.3
KNN	Weighted + PCA	80.0	-2.7
Decision Tree	Fine + PCA	61.8	-2.5
Naive Bayes	Kernel + PCA	58.9	-0.8

Table 8: Best results for previous algorithms applying PCA

When applying PCA, a greater error is obtained in all the algorithms in which it has been studied. The average variation when applying PCA is -2.3%. Therefore, it can be determined that, in the set of data studied, the application of PCA is not useful.

At the moment, the best results have been obtained by applying KNN. The average success of its six methods is 76.18%, with the best result being 82.7% using the Weighted KNN method. The next step of the experiment is to carry out a study, to obtain the optimal value for the value k. For this, Weighted KNN has been experimented and the number of neighbours has been varied starting from the default value and in increments of five. Figure 9 clearly shows that for k = 35, there is an absolute minimum.

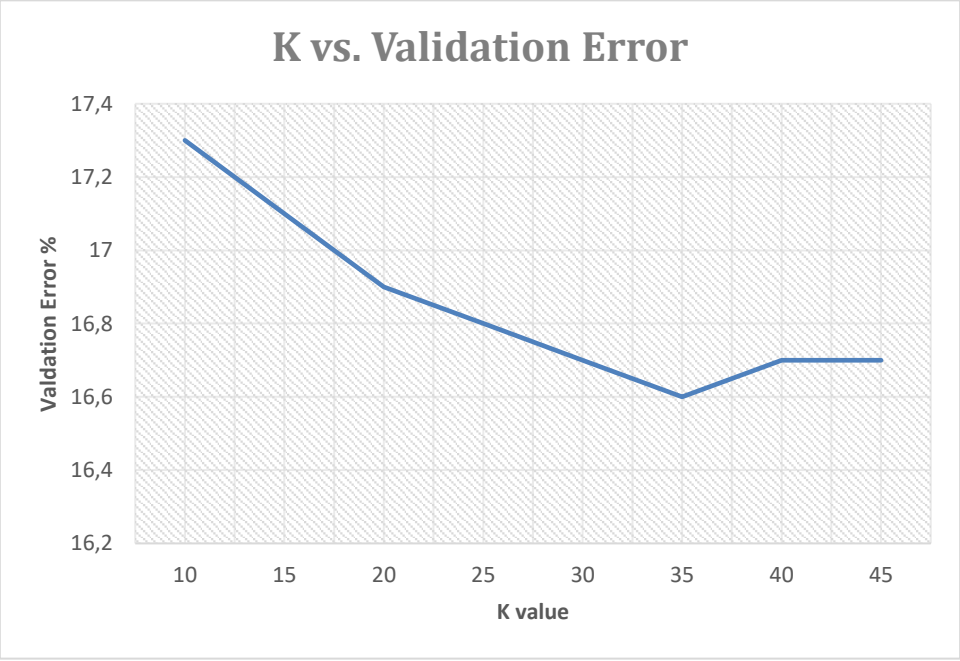


Figure 9: K value vs. Validation Error.

An error rate of 16.6% represents an accuracy of 83.4%, so that adjusting the value of k, an improvement of 0.7% is obtained. With the Weighted KNN result, with k adjusted, a new better result is established in the experiment. With this model we can now obtain different relevant data when it comes to knowing the relationship between false positives and false negatives. For this we are going to use the confusion matrix, shown in Figure 10.

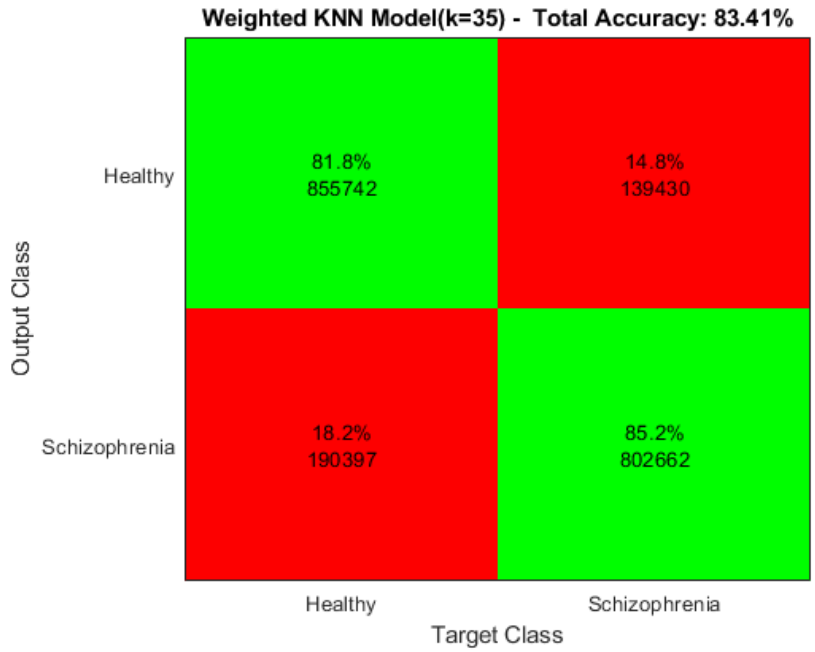


Figure 10: Confusion Matrix of Weighted KNN with k=35 without PCA.

The confusion matrix shows the number of data that have been classified in each category against the actual value of the data. On the x-axis it shows the validation classes and on the Y-axis, the predicted class. Obviously, the data on the main diagonal (in green) are data that have been correct, since they belong to the same class as predicted. In the case of the secondary diagonal (in red), they are the values that have been misclassified, that is, the data has been classified in a class that is not correct.

In the analysis of the data it has been established that the positive value is the result of schizophrenia, so the true negatives and true positives are arranged according to Figure 12 in APPENDIX A. The measures to evaluate are:

- **Accuracy**: It is the ratio of correct predictions to total predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.8341 = 83.41\% \quad \textbf{(Eq. 39)}$$

- **Recall**: It is the accuracy of finding true positives, i.e. correctly diagnose schizophrenia. This measurement is also known as sensitivity or true positive rate (TPR).

$$Recall = \frac{TP}{TP + FN} = 0.852 = 85.2\% \quad \textbf{(Eq. 40)}$$

- **Specifity**: it is the accuracy of finding true negatives, that is, of correctly diagnosing healthy patients. This measurement is also known as true negative rate (TNR). The false positive rate equals 1-Specifity.

$$Specifity = \frac{TN}{FP + TN} = 0.818 = 81.8\% \quad \textbf{(Eq. 41)}$$

In medical terminology, another series of measures is used that, instead of evaluating the test, evaluates the probability that the test is right about a predicate result. That is, in these measures the subject is evaluated and not the diagnostic capacity of the test.

- **Positive Predicted Value (PPV)**: It is the probability that the test is successful having given a positive result. In other words, if it comes out as a result schizophrenia, the probability that the test is correct.

$$PPV = \frac{TP}{FP + TP} = 0.808 = 80.8\% \quad \textbf{(Eq. 42)}$$

- **Negative Predicted Value (NPV)**: It is the probability that the test is correct having given a negative result. In other words, it comes off as a result Healthy the probability that the test is correct.

$$NPV = \frac{TN}{TN + FN} = 0.86 = 86\% \quad \textbf{(Eq. 43)}$$

The aim of the project is to properly classify schizophrenia. Therefore, the most important measure of those calculated is the Recall, since it measures the percentage of True Positives the model generates. If we look at the specificity and the recall, the second is higher, so that patients with schizophrenia are better predicted than healthy patients. This is what is desired, since, as previously mentioned, it is better to classify a healthy patient as schizophrenic than a schizophrenic as healthy. To improve the recall in future research, the misclassification costs can be used.

Medical measures are very important because with them we know the probabilities of the test being correct in each of the cases. A very good result has been obtained, which is 80.6% for a test resulted as positive, and 86% when is negative.

The designed interface is simple and intuitive. To use it, simply select the EEG to analyse and hit the "Predict Case" button. While it is processing, a wait message will appear, and when you are finished, the result will appear as shown in Figure 11. To obtain this result, an average is made between the predicted points, and if it is greater than 0.5, the EEG is classified as schizophrenic, and otherwise, as healthy.

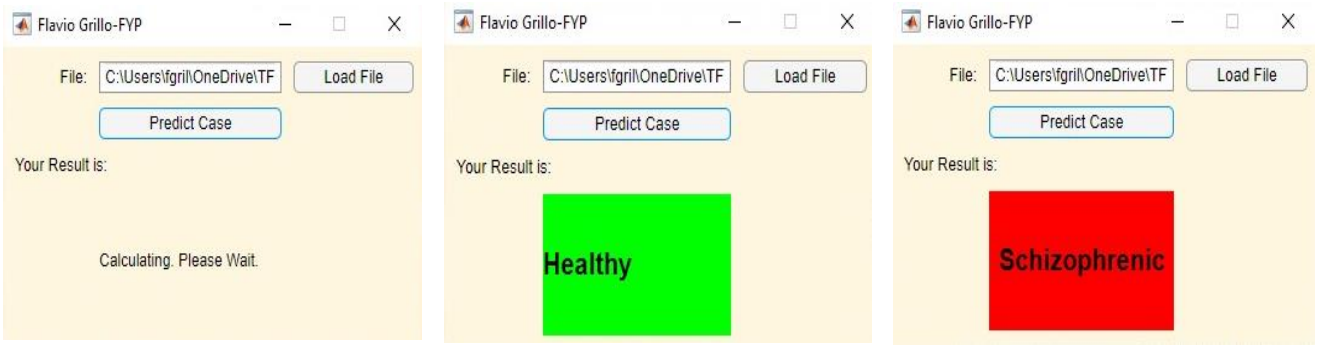


Figure 11: Developed user interface.

In summary, the best model obtained is through the Weighted KNN algorithm using 35 neighbours. The result of the model is an accuracy of 83.4% and a recall of 85.2%. This model is used in the user interface created specifically for the purpose of simplifying the use of the result of the project.

Gross results obtained in the experiment are found in APPENDIX A

14. TIME MANAGEMENT

This project has been carried out using a Gantt chart as a temporary guide for the tasks required for the completion of the project.

The Gantt Chart in Appendix D is the Gantt initially raised and is the one that has been followed. All the tasks have been carried out on their date, except for the Experiment Evaluation task, which took longer than normal, so while the results were being obtained, the report began to be drafted.

Due to the COVID-19 crisis, the delivery period has been extended by two months. The project was not affected, as the results had previously been collected using computers from the University of Hertfordshire. The memory and the slides have been made in the original period. It is also true that thanks to the extended deadline, the format of the slides has been adapted to be used in defence. This was because the original requirements changed, making that instead of a face-to-face defence of the project, a defence was made through a video.

In general, it can be considered that the time management of the project has been optimal. From the beginning, a division of tasks and the time required for each of them was made, and this has been carried out correctly, except in the case of the experiment evaluation due to the high computation time required. Anyway, given the technical problem, it was possible to continue according to the Gantt Chart, making the report at the same time as the experiment evaluation.

15. CONCLUSION

Artificial intelligence in its branch of machine learning is increasingly used in all areas and especially in medicine. Some diseases, such as brain diseases, are very difficult to diagnose due to the complexity of the brain and how confusing the results of some diagnostic tests are. With machine learning, it is possible to diagnose automatically with great precision in a simpler way, helping doctors in their daily work.

The project has focused on the diagnosis of schizophrenia, a disease that affects 1% of the world population. The objective is therefore to classify patients as healthy or schizophrenic, through one of the most useful and difficult tests to understand, the EEG. After processing the EGG, obtaining different features and the use of ML algorithms, different results have been obtained that have been compared with each other. In the experiment carried out, it has been studied how the use of PCA affects the data set and the optimal value of k has been obtained for the KNN algorithms.

The best model that has been obtained has been with the Weighted KNN algorithm with k = 35 and without PCA. This model classifies with an accuracy of 83.4% between the two categories. Regarding the classification of schizophrenia, that is, the recall, the percentage is 85.2%. The PPV is 80.6%, so a positive test has that probability of getting it right. The model thus exceeded all initial expectations, as the disease can be diagnosed through the EEG test. In addition, a user interface has been developed to help non-technical users to use the model.

In a future development, it should be investigated which features are most useful in order to reduce computing time. Another way to improve the model created is through the application of misclassification costs, for which it would require a study of the severity of misclassifying a schizophrenic patient.

REFERENCES

Andrzejak, R. G. *et al.* (2003) “Bivariate surrogate techniques: Necessity, strengths, and caveats,” *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*. Phys Rev E Stat Nonlin Soft Matter Phys, 68(6). doi: 10.1103/PhysRevE.68.066202.

Antoni, J. (2006) “The spectral kurtosis: A useful tool for characterising non-stationary signals,” *Mechanical Systems and Signal Processing*. Academic Press, 20(2), pp. 282–307. doi: 10.1016/j.ymssp.2004.09.001.

Arnhold, J. *et al.* (1999) “A robust method for detecting interdependences: Application to intracranially recorded EEG,” *Physica D: Nonlinear Phenomena*. Elsevier, 134(4), pp. 419–430. doi: 10.1016/S0167-2789(99)00140-2.

Bougou, V. *et al.* (2020) *Evaluation of EEG Connectivity Network Measures based Features in Schizophrenia Classification*.

Chicharro, D. and Andrzejak, R. G. (2009) “Reliable detection of directional couplings using rank statistics,” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*. American Physical Society, 80(2), p. 026217. doi: 10.1103/PhysRevE.80.026217.

Dixon, S. (2006) *ONSET DETECTION REVISITED*.

European Data Format (EDF) (no date). Available at: https://www.edfplus.info/index.html (Accessed: April 8, 2020).

Fallani, F. D. V. *et al.* (2010) “A graph-theoretical approach in brain functional networks. Possible implications in EEG studies,” *Nonlinear Biomedical Physics*. BioMed Central, 4(SUPPL. 1), p. S8. doi: 10.1186/1753-4631-4-S1-S8.

fMRI (2018). Available at: https://psychcentral.com/lib/what-is-functional-magnetic-resonance-imaging-fmri/ (Accessed: April 8, 2020).

Fujishima, T. (1999) “Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music,” *ICMC*.

Horwitz, B. (2003) “The elusive concept of brain connectivity,” *NeuroImage*. Academic Press Inc., 19(2), pp. 466–470. doi: 10.1016/S1053-8119(03)00112-5.

Johns Hopkins Medicine (2017) *Computed Tomography Scan of the Brain | Johns Hopkins Medicine*. Available at: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/computed-tomography-ct-or-cat-scan-of-the-brain (Accessed: April 8, 2020).

Keepers, G. A. *et al.* (2020) *THE AMERICAN PSYCHIATRIC ASSOCIATION PRACTICE GUIDELINE FOR THE TREATMENT OF PATIENTS WITH SCHIZOPHRENIA Guideline Writing Group Systematic Review Group Committee on Practice Guidelines APA Assembly Liaisons*.

Kemp, B. *et al.* (1992) "A simple format for exchange of digitized polygraphic recordings," *Electroencephalography and Clinical Neurophysiology*. Electroencephalogr Clin Neurophysiol, 82(5), pp. 391–393. doi: 10.1016/0013-4694(92)90009-7.

Koenig, T. *et al.* (2001) "Decreased functional connectivity of EEG theta-frequency activity in first-episode, neuroleptic-naïve patients with schizophrenia: Preliminary results," *Schizophrenia Research*. Schizophr Res, 50(1–2), pp. 55–60. doi: 10.1016/S0920-9964(00)00154-7.

Lazega, E. (1995) "Wasserman Stanley, Faust Katherine, Social network analysis: methods and applications.," in *Revue Française de Sociologie*. Cambridge ;;New York: Cambridge University Press, pp. 781–783. Available at: http://www.persee.fr/web/revues/home/prescript/article/rfsoc_0035-2969_1995_num_36_4_4431 (Accessed: May 14, 2020).

Lehner, B., Widmer, G. and Sonnleitner, R. (2014) *ON THE REDUCTION OF FALSE POSITIVES IN SINGING VOICE DETECTION*.

Lerch, Alexander. (2012) *An introduction to audio content analysis : applications in signal processing and music informatics*. Wiley.

Medina, R. J. (2012) "Communication-Systems.pdf." Rice University.

Morley Andrew Morley Hons, A. and Hill, L. (2016) *0-20 system EEG Placement Specialist Respiratory Clinical Physiologist, Royal Hospital for Sick Children, Edinburgh*.

Murthy, H. A. *et al.* (1999) "Robust text-independent speaker identification over telephone channels," *IEEE Transactions on Speech and Audio Processing*. IEEE, 7(5), pp. 554–568. doi: 10.1109/89.784108.

M.V.Lomonosov Moscow State University (no date) *EEG Database - Schizophrenia*. Available at: http://brain.bio.msu.ru/eeg_schizophrenia.htm (Accessed: April 8, 2020).

Naghavi, M. (2019) “Global, regional, and national burden of suicide mortality 1990 to 2016: systematic analysis for the Global Burden of Disease Study 2016,” *BMJ (Clinical research ed.)*. NLM (Medline), 364, p. l94. doi: 10.1136/bmj.l94.

National Health System, N. (2018) *Electroencephalogram (EEG) - NHS*. Available at: https://www.nhs.uk/conditions/electroencephalogram/ (Accessed: April 8, 2020).

Newson, J. J. and Thiagarajan, T. C. (2019) “EEG Frequency Bands in Psychiatric Disorders: A Review of Resting State Studies,” *Frontiers in Human Neuroscience*. Frontiers Media SA, 12, p. 521. doi: 10.3389/fnhum.2018.00521.

Olejarczyk, E. and Jernajczyk, W. (2017) “Graph-based analysis of brain connectivity in schizophrenia,” *PLOS ONE*. Edited by D. Yao. Public Library of Science, 12(11), p. e0188629. doi: 10.1371/journal.pone.0188629.

OpenLearn (2019) *How FMRI works - OpenLearn - Open University*. Available at: https://www.open.edu/openlearn/body-mind/health/health-sciences/how-fmri-works (Accessed: April 8, 2020).

Peeters, G. (2014) “A large set of audio features for sound description (similarity and classification) in the CUIDADO project.”

Plavchan, P. (2011) *CalTech Wiki*. Available at: http://coolwiki.ipac.caltech.edu/index.php/What_is_a_periodogram%3F (Accessed: April 8, 2020).

Protter, M. and Morrey, C. (1977) “College calculus with analytic geometry.”

Quian Quiroga, R. *et al.* (2002) “Performance of different synchronization measures in real data: A case study on electroencephalographic signals,” *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*. American Physical Society, 65(4), p. 14. doi: 10.1103/PhysRevE.65.041903.

Rosenblum, M. G. *et al.* (2002) “Identification of coupling direction: Application to cardiorespiratory interaction,” *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 65(4), p. 11. doi: 10.1103/PhysRevE.65.041909.

Rosenblum, M. G., Pikovsky, A. S. and Kurths, J. (1996) *Phase Synchronization of Chaotic Oscillators*.

Sakellariou, D. *et al.* (2016) “Connectivity Measures in EEG Microstructural Sleep Elements,” *Frontiers in Neuroinformatics*. Frontiers Media S.A., 10(FEB), p. 5. doi: 10.3389/fninf.2016.00005.

Sham, P. C., MacLean, C. J. and Kendler, K. S. (1994) “A typological model of schizophrenia based on age at onset, sex and familial morbidity,” *Acta Psychiatrica Scandinavica*. John Wiley & Sons, Ltd, 89(2), pp. 135–141. doi: 10.1111/j.1600-0447.1994.tb01501.x.

Sim, K. *et al.* (2006) “Psychiatric comorbidity in first episode schizophrenia: A 2 year, longitudinal outcome study,” *Journal of Psychiatric Research*. Pergamon, 40(7), pp. 656–663. doi: 10.1016/j.jpsychires.2006.06.008.

Smith, J. O. (2011) *Spectral Audio Signal Processing*, *Center for Computer Research in Music and Acoustics (CCRMA)*. Available at: https://ccrma.stanford.edu/~jos/sasp/ (Accessed: April 8, 2020).

Spectral Descriptors (2014) *The Mathworks Inc.* Available at: https://www.mathworks.com/help/audio/ug/spectral-descriptors.html#SpectralDescriptorsExample-12 (Accessed: April 8, 2020).

Sporns, O. and Kötter, R. (2004) “Motifs in Brain Networks,” *PLoS Biology*, 2(11). doi: 10.1371/journal.pbio.0020369.

Stuart, G. and Laraira, M. (2006) “Enfermería psiquiátrica: principios y práctica.” Available at: http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=SUV.xis&method=post&formato=2&cantidad=1&expresion=mfn=005887 (Accessed: April 8, 2020).

Tzanetakis, G. and Cook, P. (1999) “Multifeature audio segmentation for browsing and annotation,” in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, pp. 103–106. doi: 10.1109/aspaa.1999.810860.

Vergin, R., O’Shaughnessy, D. and Farhat, A. (1999) “Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition,” *IEEE Transactions on Speech and Audio Processing*. IEEE, 7(5), pp. 525–532. doi: 10.1109/89.784104.

Watts, D. J. and Strogatz, S. H. (1998) “Collective dynamics of ‘small-world’ networks,” *Nature*. Princeton University Press, 393(6684), pp. 440–442. doi: 10.1038/30918.

WHO (2019) *Esquizofrenia*. Available at: https://www.who.int/es/news-room/fact-sheets/detail/schizophrenia (Accessed: April 8, 2020).

WHO, W. H. O. (2019) *Schizophrenia*. Available at: https://www.who.int/news-room/fact-sheets/detail/schizophrenia (Accessed: April 8, 2020).

APPENDIX A: Experiment Results

SVM	% Success
Linear	53.6
Quadratic	55.0
Cubic	51.8
Fine Gaussian	52.7
Medium Gaussian	51.9
Coarse Gaussian	52.3

Table 9: SVM Results

KNN	% Success
Fine	81.2
Medium	71.5
Coarse	70.8
Cosine	70.5
Cubic	80.4
Weighted	82.7

Table 10: KNN Results

Decision Tree	% Success
Fine	64.3
Medium	62.2
Coarse	59.2

Table 11: Decision Tree Results

Naïve Bayes	% Success
Gausian Naive	58.0
Kernel	59.7

Table 12: Naive Bayes Results

USING PCA	
Algorithm	% Success
Weighted KNN k=20	80.5%
Weighted KNN k=10	80%
Fine KNN PCA k=15	66.9%
Linear SVM	53.7%
Quadratic SVM	51.7%

Table 13: PCA Results

Weighted KNN	
K= 10	82.7%
K= 20	83.1%
K=25	83.2%
K=30	83.3%
K=35	83.4%
K=40	83.3%
K=45	83.3%

Table 14: Weighted KNN with different values of K

Weighted KNN		
K	% Success	%Error
10	82,7	17,3
15	82,9	17,1
20	83,1	16,9
25	83,2	16,8
30	83,3	16,7
35	83,4	16,6
40	83,3	16,7
45	83,3	16,7

Table 15: Different Values of K and its % of Error and Success

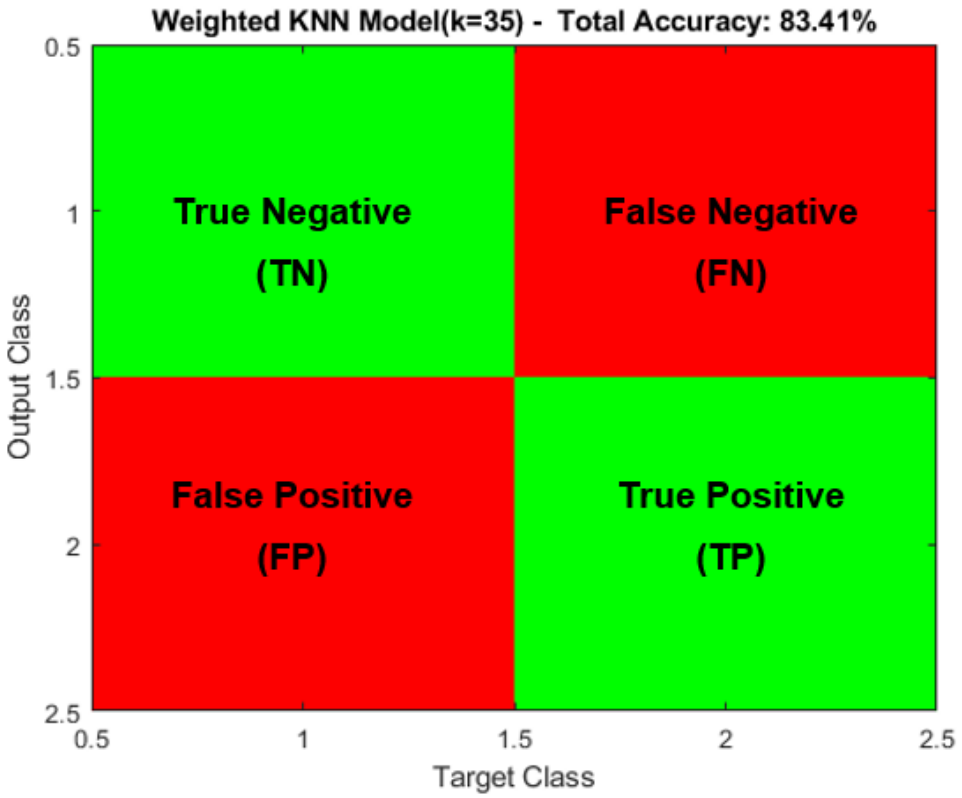


Figure 12: Confusion Matrix Explanation

APPENDIX B: EEG Channels and Locations

Channel	Location
1	'Fp2'
2	'F8'
3	'T4'
4	'T6'
5	'O2'
6	'Fp1'
7	'F7'
8	'T3'
9	'T5'
10	'O1'
11	'F4'
12	'C4'
13	'P4'
14	'F3'
15	'C3'
16	'P3'
17	'Fz'
18	'Cz'
19	'Pz'

Table 16: Display of electrodes.

APPENDIX C: MATLAB Code

Main

```
        'myFunctions'
        'Toolboxes/voicebox/voicebox'
        'Toolboxes/ACA/ACA'
        'Toolboxes/HERMES/HERMES Toolbox/lib/tim-matlab-1.2.0'
        'Toolboxes/HERMES/HERMES Toolbox/H_minifunctions'
        'Toolboxes/BCT'
        'Toolboxes\HERMES\HERMES Toolbox\H_connectivity'
        'EDF_Files'

%Obtain the digital signals
    'h01.edf' 'h02.edf'
'h03.edf' 'h04.edf' 'h05.edf' 'h06.edf' 'h07.edf' 'h08.edf' 'h09.edf' 'h10.
edf'

'h11.edf' 'h12.edf' 'h13.edf' 'h14.edf' 's01.edf' 's02.edf' 's03.edf' 's04.
edf' 's05.edf'
  's06.edf' 's07.edf' 's08.edf' 's09.edf' 's10.edf' 's11.edf' 's12.edf'
's13.edf'
  's14.edf'
% this data is from: https://repod.pon.edu.pl/dataset/eeg-in-schizophrenia

for
 %%IF EDF FILES ARE USED
    %Convert data from edf to Matlab
                                %f=header.frequency;
            %Nyquist Frequency

    %%
    %IF EEA FILES ARE USED
    %If this section is used, the code for the creation of the model has to
    %change (put correct number of channels, responses...etc
% %
% %     data=importdata(files(i,:));
% %     channels=16; %As stated in the paper where .eea files were obtained.
% %     N=7680; %As stated in the paper where .eea files were obtained
% %     dataeeg=reshape(data,[channels,N]);
% %
% %     fs=128; %f=header.frequency;
% %     fn=fs/2; %Nyquist Frequency

    %%
    %PREPROCESSING
    %A=data(1,:); %select only first eeg channel
                %order of filter
                    %low cut frequency in Hz %COrresponds with the min
delta freq
```

```
[redacted] %high cut frequency in Hz %Corresponds with the max
Beta freq
[redacted] 'bandpass' [redacted]
[redacted] %filtered signal
    %%
    %Enframe
[redacted] %frame length = to 1sec
[redacted] %(signal,
duration of the signal, overlap). overlap=header.duration*overlap%
    %[W,M]=size(data_frames); %W length of each frame, M data_frames

    %%
    %Feature Extraction
[redacted] %obtain the
feature vector

[redacted]

[redacted]



    %%
    %Training and Validation
[redacted]
    if [redacted] 'h' [redacted] %if the first letter of the file is H will add
zeros to the healthy row, and ones to the scqizofrenia row

[redacted]
    else
[redacted]
    end
[redacted]
[redacted] 'Calculated Feature vector number %d  \n' [redacted]

[redacted] 'feature_vector' 'i' 'outputHermes' 'ClassicFeats' '-
v7.3' [redacted]
end
[redacted]
%%
%Create the model: KNN, k=35, No PCA

[redacted] 'VariableNames' [redacted] 'Fp2'
'F8' 'T4' 'T6' 'O2' 'Fp1' 'F7' 'T3' 'T5' 'O1' 'F4' 'C4' 'P4'
'F3' 'C3' 'C3' 'P3' 'Fz' 'Cz' 'labels' [redacted]

[redacted] 'Fp2' 'F8' 'T4' 'T6' 'O2' 'Fp1' 'F7' 'T3' 'T5'
'O1' 'F4' 'C4' 'P4' 'F3' 'C3' 'C3' 'P3' 'Fz' 'Cz' [redacted]

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
```

```
                                ...
                    ...
                ...
    'Distance'  'Euclidean'  ...
    'Exponent'       ...
    'NumNeighbors'       ...
    'DistanceWeight'  'SquaredInverse'  ...
    'Standardize'        ...
    'ClassNames'

% Create the result struct with predict function
                                                'VariableNames'


% Add additional fields to the result struct
                                                'To make predictions on a new
predictor column matrix, X, use: \n  yfit = c.predictFcn(X) \nreplacing
''c'' with the name of the variable that is this struct, e.g.
''trainedModel''. \n \nX must contain exactly 19 columns because this model
was trained using 19 predictors. \nX must contain only predictor columns in
exactly the same order and format as your training \ndata. Do not include
the response column or any columns you did not import into the app. \n
\nFor more information, see <a href="matlab:helpview(fullfile(docroot,
''stats'', ''stats.map''),
''appclassification exportmodeltoworkspace'')">How to predict using an
exported model</a>.'

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
% Convert input to table
                                                'VariableNames'   'Fp2'
'F8'  'T4'  'T6'  'O2'  'Fp1'  'F7'  'T3'  'T5'  'O1'  'F4'  'C4'  'P4'
'F3'  'C3'  'C3'  'P3'  'Fz'  'Cz'  'labels'

                  'Fp2'  'F8'  'T4'  'T6'  'O2'  'Fp1'  'F7'  'T3'  'T5'
'O1'  'F4'  'C4'  'P4'  'F3'  'C3'  'C3'  'P3'  'Fz'  'Cz'


% Perform cross-validation
                                                                'KFold'


% Compute validation predictions


% Compute validation accuracy
                                                        'LossFun'
'ClassifError'

%The End.
```

TimeAndFreqFeatures

```
function
```

```
%TIME DOMAIN FEATURES
                    %minimun
                    %maximun
                    %standard deviation
                      %mean
                                                %quartles to 25% 50% and
75%
                                              %percentiles to 25% 50% and
75%
                    %zero Crossing rate
% Energy = sum(abs(data_frames).^2); %Energy
% BandPower_Delta = bandpower(data_frames,fs,[0,4]);
% BandPower_Theta = bandpower(data_frames,fs,[4,7]);
% BandPower_Alpha = bandpower(data_frames,fs,[8,15]);
%BandPower_Beta = bandpower(data_frames,fs,[16,31]);
% [Energy_Delta,Energy_Theta, Energy_Alpha,
Energy_Beta]=BandFilterDesing(data_frames,fn);
```

```
%FREQUENCY DOMAIN FEATURES
% SpectralAcf= PitchSpectralAcf(data_frames,fs);
```

```
%Creation of the feature vector
```

```
    % Energy;
    % BandPower_Delta;
    % BandPower_Theta;
    % BandPower_Alpha;
    % BandPower_Beta;
    % Energy_Delta;
    % Energy_Theta;
    % Energy_Alpha;
    % Energy_Beta;

    % SpectralAcf;
```

```
end
```

FunctionalAndEffective

```
function

% Gets the size of the analysis data.

%%HermesCM Funciona

%%HermesGS FUnciona

%%HermesPS  funciona

%%HermesIT  Funciona


end
```

FadnE_CM

```
function
```

```
%Esto se utiliza para calcular COH y iCOH
```

```
% Calculates the indexes for each window and trial.
    for
    for
                                                                'coeff'
                                                                'coeff'
    end
    end
```

```
   % for trial = 1: trials
                                    'coherence'
                                    'icoherence'
   % end
```

```
    %if is NaN put it as 0
   % output.iCOH.rawdata ( isnan ( output.iCOH.rawdata ) ) = 0;
```

```
    %Mean of the data
```

```
    %make COH and ICOH single
```

```
    %Delete columns with only 1
```

[redacted]

[redacted]

```
end
```

```
FadnE_GS
```

```
function [redacted]
```

[redacted]

[redacted]

```
%DATA TO CONFIRM AND SEARCH ABOUT
```

[redacted]

```
% Reserves the needed memory.
```

[redacted]

```
% Calculates the indexes for each trial and pair of sensors.
```

[redacted]

```
                                % Number of vectors
```

[redacted]

```
% Effective number of distances for each reference vector
```

[redacted]

```
 for
```

[redacted]

```
 end
```

```
                'single'            % This is used as a matrix to save
calc time
                    'single'
                'single'
                            'uint32'


    % Matrixes of rank distances
                            'uint32'



 %%%%%%% Calculation of the indexes %%%%%%%%
 for

    for

        %  First we sort the distances for all the channels in each trial
once
        % to save time, as sorting is a time-consuming process

        for

            for

            end


            for
                for

                end
            end

            % And now for the elements of the diagonal


        end

        % Prepare the distance matrix X and Hsums1


        % Now perform the calculations for each channel (fixed window and
        % trial

        for

        % Calculate the matrices of ranks.
```

```
        if
        else

            for
            end
        end

        for

            % Creating the embbeding vectors
            for
            end


            % Now we set to zero the distances to vectors closer to each
            % reference than the Theiler window

            for
                for
                end
            end


            %% Calculate the matrices of ranks.


                % Matrixes of rank distances
                for
                end


            for

            end


            for
```

[redacted]

[redacted]

```
%S(X|Y)=Rkl/Rcond(X|Y)
%H(X|Y)=log(RNX/Rcond(X|Y))
%N(X|Y)=(RNX-Rcond(X|Y))/(RNX)
%M(X|Y)=(RNX-Rcond(X|Y))/(RNX-RkX)
%L(X|Y)=(GN-G(X|Y))/(GN-Gk)
```

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

```
end
```

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]
```
            end
        end
```

[redacted]
```
    end
end
```

```
% Averages across trials. WARNINGS!!!!
```

[redacted]
```
    if
%         warndlg ('Negative values of GS index S were obtained. It is
advisable to increase the number of neighbours', 'GS warning');
    end
```

[redacted]
```
    if
%         warndlg ('Negative values of GS index H were obtained. It is
advisable to increase the number of neighbours', 'GS warning');
```

```
    end


    if
%         warndlg ('Negative values of GS index M were obtained. It is
advisable to increase the number of neighbours', 'GS warning');
    end


    if
%         warndlg ('Negative values of GS index N were obtained. It is
advisable to increase the number of neighbours', 'GS warning');
    end


    if
%         warndlg ('Negative values of GS index L were obtained. It is
advisable to increase the number of neighbours', 'GS warning');
    end


end
```

FadnE_IT

```
function
```

```
% if channels<15
%     Nlags = channels-1;
% else
                % 1, to save time but...
% end

                        % para la parcialization
% tWindRad  = 10; % para la TMI

% % Reservamos memoria para cada indice.
%   output.MI.data_frames  = zeros ( channels, channels, windows );
%  output.PMI.data_frames = zeros ( channels, channels, windows );
%  output.TE.data_frames  = zeros ( channels, channels, windows );
% output.PTE.data_frames = zeros ( channels, channels, windows );


%  We go over all windows and all pairs of sensors.
for
    for

        % Prepare the data_frames in the first channel (of each pair ) for
analysis

        if
        else
        end

        %%%%
        %%%%

% Now run for channels greater than ch1
        for

            % Places each trial in a separate cell field.
            if
            else
            end
            %%%%
```

```
[redacted] %%%%

[redacted] 'yLag' [redacted] 'k' [redacted]

%                if H_check ( config.measures, 'TMI' )
%                    output.TMI.data_frames ( i, j, window ) =
max(mutual_information_t ( celldata_i, celldata_j, tWindRad,'yLag',0:Nlags-
1,'k',config.Nneighbours));
%                    % asimetrica? output.TMI.data_frames ( j, i, window ) =
output.MI.data_frames ( i, j, window );
%                end

[redacted]

[redacted] for [redacted]

[redacted]

[redacted] if
[redacted]
[redacted] else
[redacted]
[redacted] end
[redacted] %%%%
[redacted]
[redacted] %%%%
[redacted]

[redacted] for
[redacted] 'yLag' [redacted] 'zLag' [redacted]
 'k' [redacted]
[redacted] end
[redacted]
[redacted] end

[redacted] %     for dt = config.TimeDelay

[redacted]
```

```
[redacted]

[redacted] for [redacted]
[redacted]
[redacted] 'yLag' [redacted] 'k' [redacted]
[redacted]
[redacted] 'yLag' [redacted] 'k' [redacted]
[redacted]
[redacted] end

[redacted]


%                    if H_check ( config.measures, 'TTE' )
%                        output.TTE.data_frames ( i, j, window ) =
max(transfer_entropy_t ( celldata_i, celldata_j, w,
tWindRad,'yLag',0:Nlags-1,'k',config.Nneighbours));
%                        output.TTE.data_frames ( j, i, window ) =
max(transfer_entropy_t ( celldata_j, celldata_i, w,
tWindRad,'yLag',0:Nlags-1,'k',config.Nneighbours));
%                    end


[redacted]

[redacted] for [redacted]
[redacted]

[redacted] % Places each trial in a separate cell field.
[redacted] if [redacted]
[redacted] else [redacted]
[redacted] end
[redacted] %%%%
[redacted] %%%%

[redacted] for [redacted]
[redacted] 'yLag'
[redacted] 'zLag' [redacted] 'k' [redacted]
[redacted] 'yLag'
[redacted] 'zLag' [redacted] 'k' [redacted]
[redacted] end
```

[redacted]
end

[redacted]
end
end

[redacted]
end

[redacted]
end

FadnE_PS

[redacted]

[redacted]
'ema'
[redacted]

% Goes throough all bands
[redacted]

% Performs a narrow band filtering
[redacted]

% filtered = H_filtfilt (fir, 1, data);
[redacted]

```
    % Calculates the wPLI, if required (all sensors at once)

        % Windows the filtered data.

        % Calculates the index for each window
        for
                                                              'cross'

        end

        % Calculates Hilbert transform
%         hilbertedata = reshape ( hilbert ( filtered ( :, : ) ), [],
channels, trials );

        % Concatenates the trials for each window

        % Gets the phase angles of the signals

        % Calculates the indexes for each pair of sensors
        for
            for

                % Gets the difference of phases

                % Calculates the indexes for each window
                for

                    % Number of bins according to Otnes & Enochson

                    % Sets the diference of angles between 0 and 2pi.

                    % Calculates Rho as 1-SE of the histogram
```

[redacted]

[redacted]

[redacted]

[redacted]

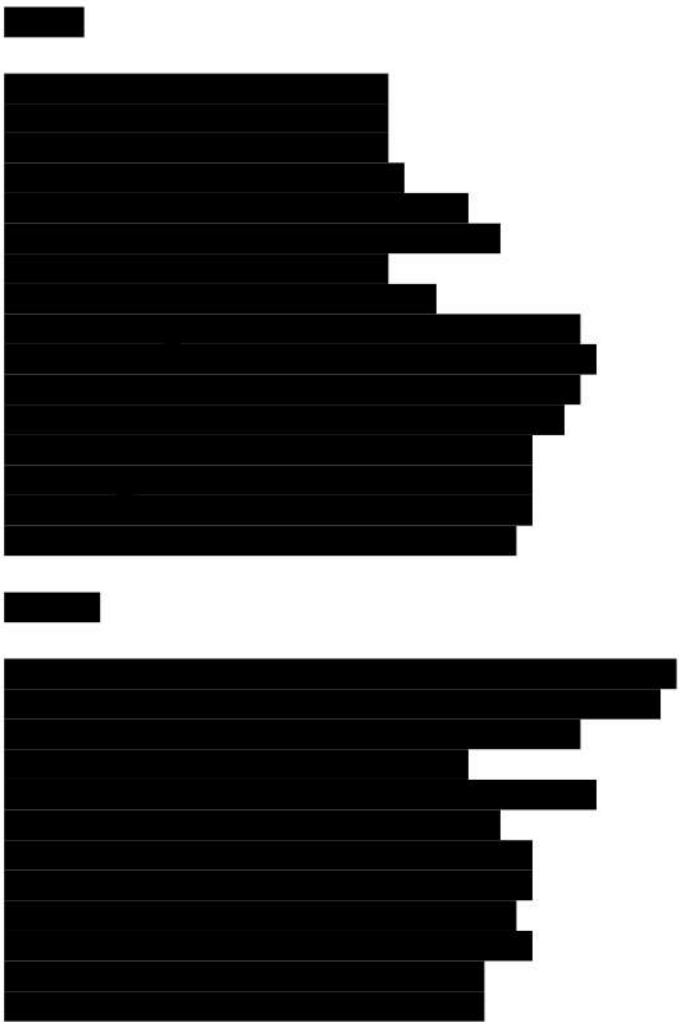
[redacted]

```
                end

            end

            % Updates the waitbar

        end

end

% Averages across trials
[redacted]

if [redacted]
end
[redacted]

[redacted]

end
```

FandE_stat

```
function
[redacted]

end
```

Features Functions and Dimensions



[redacted]

[redacted]

BandFilterDesing

```
function
%Common Parameters
                                                    % Passband
Ripple
                                                    % Stopband
Ripple

%Delta Band Filter
                                                  % Delta Passband
                                                    % Bandstop
Frequencies

                                                    % Determine
Optimal Order
                                                    % Transfer
Function Coefficients
    % [sos_delta,g_delta] = tf2sos(b_delta,a_delta);          % Second-Order-
Section For Stability

    % figure(1)
    % freqz(sos_delta, 4096, fs);                             % Filter Bode
Plot

                                                %Energy Delta Band

%Theta Band Filter
                                                    % Theta
Passband
                                                    % Bandstop
Frequencies
                                                    % Determine
Optimal Order
                                                    % Transfer
Function Coefficients
    % [sos_theta,g_theta] = tf2sos(b,a);                      % Second-Order-
Section For Stability

    % figure(2)
    % freqz(sos_theta, 4096, fs);                             % Filter Bode
Plot
                                                %Energy Theta Band

%Alpha Band Filter
```

```
                                                                % ALpha
Passband
                                                                % Bandstop
Frequencies
                                                                % Determine
Optimal Order
                                                                % Transfer
Function Coefficients
    % [sos_alpha,g_alpha] = tf2sos(b,a);                       % Second-Order-
Section For Stability

    % figure(3)
    % freqz(sos_alpha, 4096, fs);                               % Filter Bode
Plot

                                                    %Energy Alpha Band

%Beta Band Filter
                                                                % Beta Passband
                                                                % Bandstop
Frequencies
                                                                % Determine
Optimal Order
                                                                % Transfer
Function Coefficients
    % [sos_beta,g_beta] = tf2sos(b,a);                          % Second-Order-
Section For Stability

    % figure(4)
    % freqz(sos_beta, 4096, fs);                                % Filter Bode
Plot
                                                    %Energy Beta Band
end
```

```
App

classdef

    % Properties that correspond to app components
    properties
                                matlab.ui.Figure
                                matlab.ui.container.GridLayout
                                matlab.ui.control.Button
                                matlab.ui.control.Button
                                matlab.ui.control.Label
                                matlab.ui.control.Label
                                matlab.ui.control.EditField
                                matlab.ui.control.Label
    end

    % Callbacks that handle component events
    methods
```

```
        % Callback function
        function

        end

        % Button pushed function: LoadFileButton
        function
                                            '*.edf'
            if
                    'User selected Cancel'
            else
              %  disp(fullfile(path, file))

                                                                    % Set the
file name here
            end

        end

        % Callback function
        function

        end

        % Button pushed function: PredictCaseButton
        function

            %%Type next state
                              'Calculating... Please Wait. Importing File'

                                    'normal'
                                      'none'


            %Type next state
                             'Calculating... Please Wait. Obtaining Features'
                                    'normal'
                                      'none'


          %Type next state
                             'Calculating... Please Wait. Predicting Result'
                                    'normal'
                                      'none'
```

```
                                        'bold'
            if          "Healthy"
                                                    '0.00,1.00,0.00'
            else
                                                    '1.00,0.00,0.00'
            end
        end

        % Callback function
        function


        end
    end

    % Component initialization
    methods

        % Create UIFigure and components
        function

            % Create FlavioGrilloFYPUIFigure and hide until all components
are created
                                                  'Visible'  'off'
                                                'Flavio Grillo-FYP'
                                                 'off'

            % Create GridLayout
                                                'fit'              '1x'
'fit'
                                              'fit'     'fit'     'fit'  'fit'
'fit'  '1x'

            % Create LoadFileButton
                                                          'push'
                                          'Load File'

            % Create PredictCaseButton
                                                        'push'
                                          'Predict Case'

            % Create YourResultisLabel
```

```
                                                    'Your Result is: '

            % Create FileEditFieldLabel
                                                                        'right'
                                                    'File:'

            % Create FileEditField
                                                                    'text'
                                                  'Please Load a File'

            % Create Label
                                                    'center'
                                  ''

            % Show the figure after all components are created
                                                    'on'
        end
    end

    % App creation and deletion
    methods

        % Construct app
        function

            % Create UIFigure and components

            % Register the app with App Designer

            if
                    app
            end
        end

        % Code that executes before app deletion
        function

            % Delete UIFigure when app is deleted

        end
    end
end
```

ImportFileGU

```
function
```

```
        './myFunctions'
        './Toolboxes/voicebox/voicebox'
        './Toolboxes/ACA/ACA'
        './Toolboxes/HERMES/HERMES Toolbox/lib/tim-matlab-1.2.0'
        './Toolboxes/HERMES/HERMES Toolbox/H_minifunctions'
        './Toolboxes/BCT'
        './Toolboxes\HERMES\HERMES Toolbox\H_connectivity'
        './EDF_Files'

%Obtain EEG Signal;
    %Convert data from edf to Matlab
                                        %f=header.frequency;
            %Nyquist Frequency
              %order of filter
                  %low cut frequency in Hz %COrresponds with the min delta
freq
                      %high cut frequency in Hz %Corresponds with the max
Beta freq
                                                  'bandpass'
                                    %filtered signal

end
```

CalculateVectorGui

```
function

        './myFunctions'
        './Toolboxes/voicebox/voicebox'
        './Toolboxes/ACA/ACA'
        './Toolboxes/HERMES/HERMES Toolbox/lib/tim-matlab-1.2.0'
        './Toolboxes/HERMES/HERMES Toolbox/H_minifunctions'
        './Toolboxes/BCT'
        './Toolboxes\HERMES\HERMES Toolbox\H_connectivity'

        './EDF_Files'

%Obtain EEG Signal;

    %Enframe
                      %frame length = to 1sec
                                                                %(signal,
duration of the signal, overlap). overlap=header.duration*overlap%
    %[W,M]=size(data_frames); %W length of each frame, M data_frames

    %Feature Extraction
                                                          %obtain the feature
vector

end
```

PredictResult

```
function
        'weightedKNN35.mat'
    if
                    "Healthy"
    else
                    "Schizophrenic"
    end
end
```

APPENDIX D: Gantt Chart

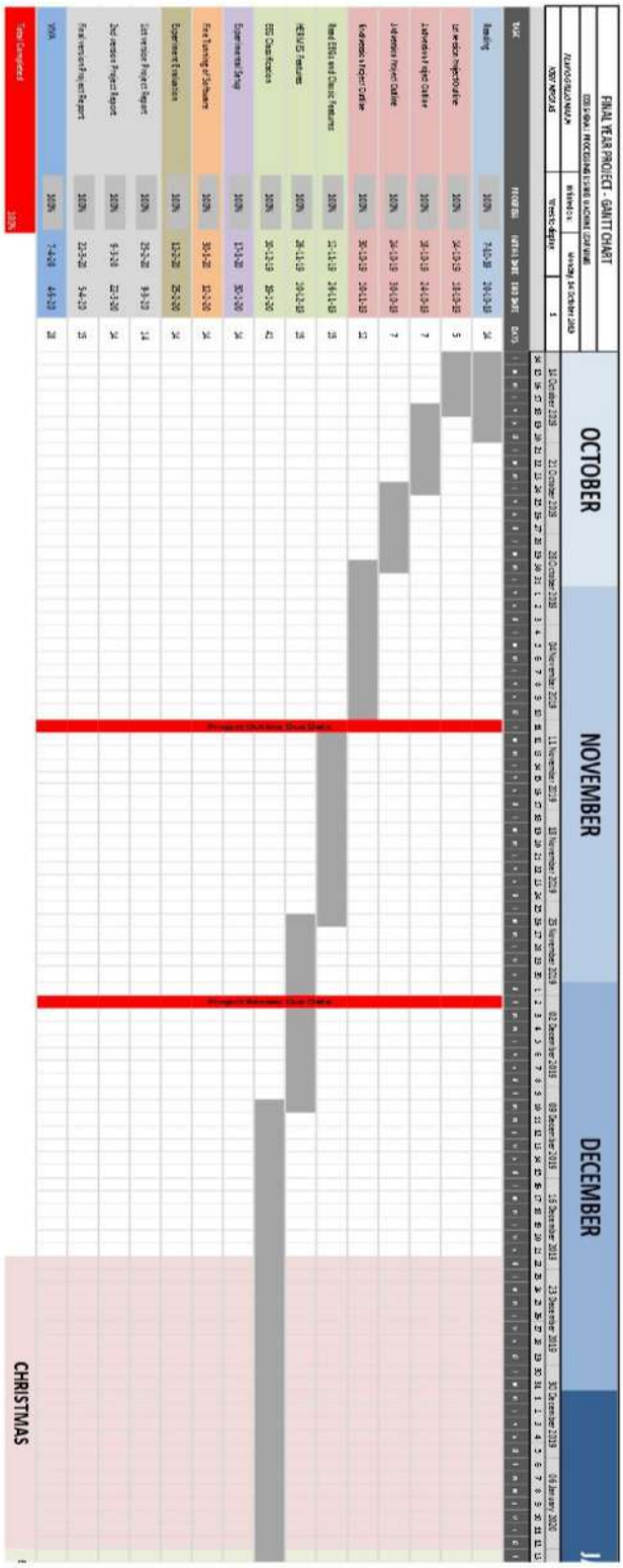


Figure 13: Gantt Chart 1/3

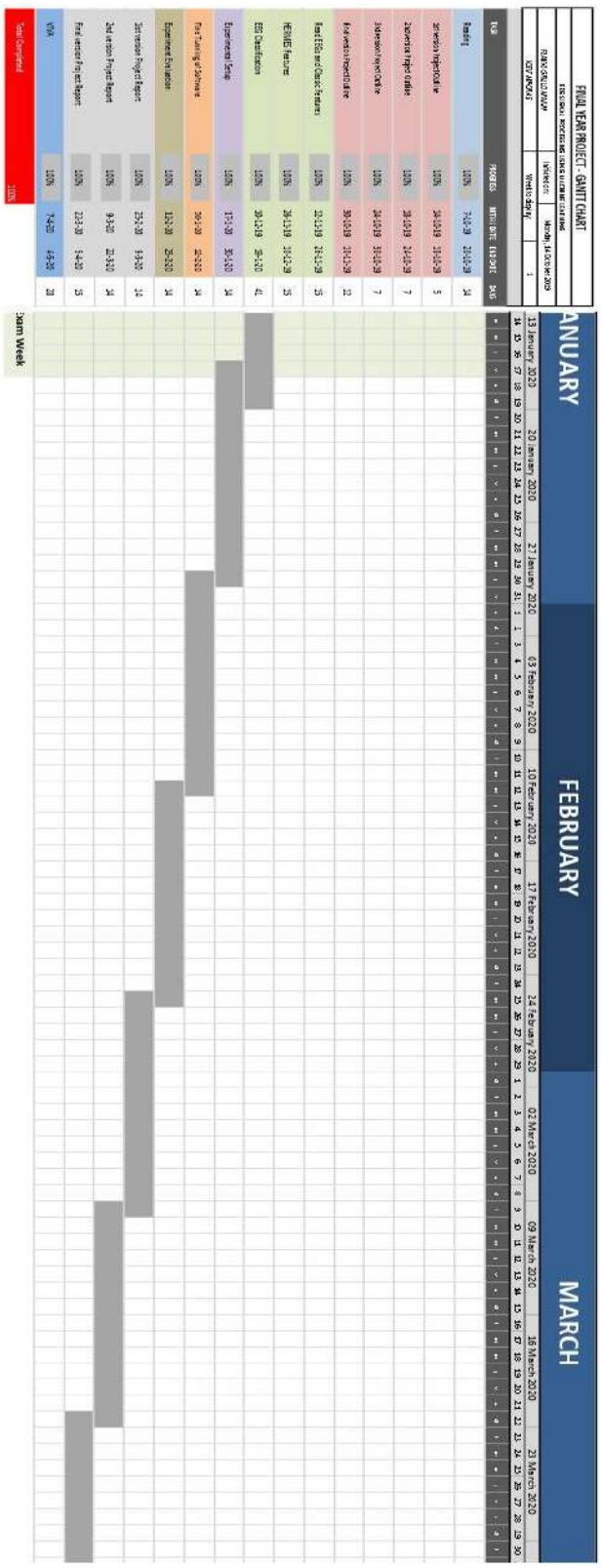


Figure 14: Gantt Chart 2/3

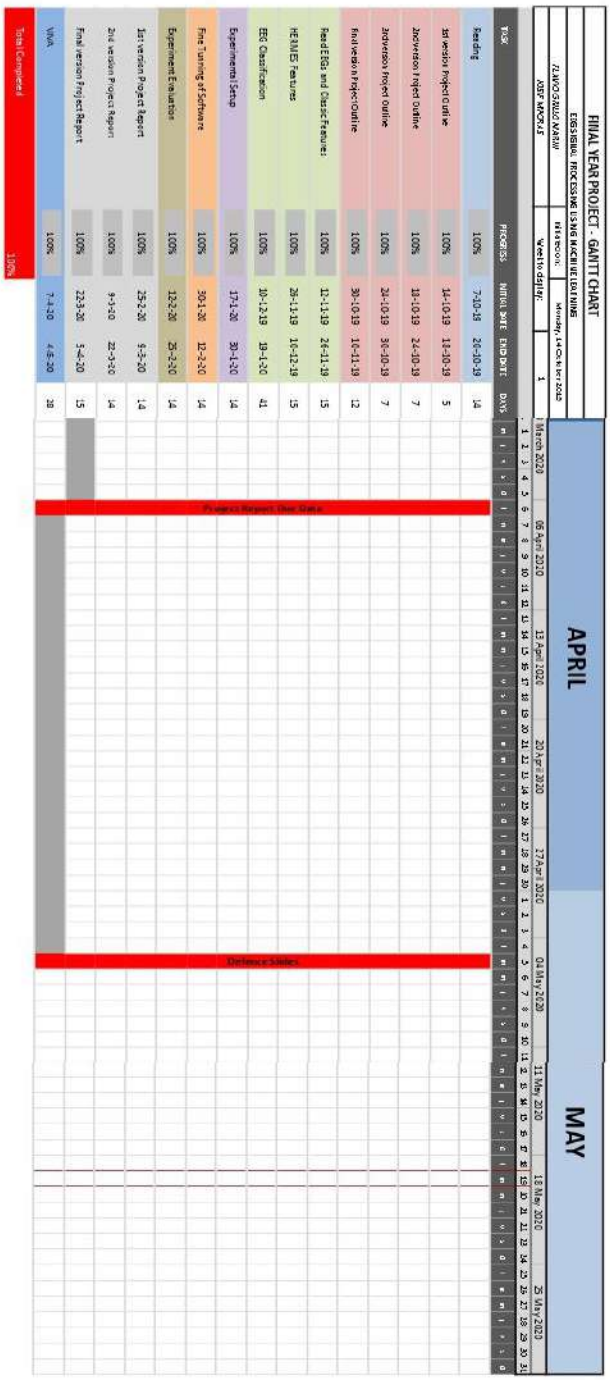


Figure 15: Gantt Chart 3/3