T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

CAPSTONE PROJECT FINAL REPORT

EEG DATA ANALYSIS FOR DETECTING RELAXATION AND CONCENTRATION MOODS

Ilhan Ozan Aral – Biomedical Engineering

Muharrem Asan – Software Engineering

Bensu Sengel – Biomedical Engineering

Can Yılmaz – Software Engineering

Advisors:

Assist. Prof. Burcu Tunç Çamlıbel

Assist. Prof. Betül Erdoğdu Şakar

ISTANBUL, June 2021

STUDENT DECLARATION

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that

- they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
- they have not received unpermitted aid for the project design, construction, report or presentation;
- they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

ABSTRACT

In the future, we can learn something about how to make AI solves the problems in science and in this specific area. On the other hand, this study could reduce the time scientists spend on any research because this study will show how to predict whether a person is focused or relaxed using EEG Signals from Machine Learning. In particular, this paper examines the possible techniques for preprocessing the signals and what techniques to use while building an ML system. Moreover, this work will provide an interface application to build an ML model and use the model to predict it. Finally, the system has many ML classifiers inside of it and any person can use them to analyze the classifier's performance with the data. After analyzing, the selected model will automatically upload into the project's folder for further use. In the end, using any saved model with the unpredicted data will show you the responses of the brain signals.

Key Words: Machine Learning, Brain Signals, EEG, Detection, Relaxation and Concentration, Moods

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SEN	Software Engineering
BME	Biomedical Engineering
EEG	Electroencephalogram
ML	Machine Learning
LDA	Linear Discriminant Analysis
SVM	Suppor Vector Machine
RF	Random Forest
DT	Decision Tree
NB	Naive Bayes
AI	Artificial Intelligence
ERP	Event Related Potential
ICA	Independent Component Analysis
EU	European Union
EDF	European Data Format

1. OVERVIEW

Electroencephalogram (EEG) is a method used to measure the electrical activity at the peripheral part of brain. These activities are measured by placing number of electrodes across the scalp of the subject. However, these days during the covid, the data has found by the internet and the preprocessing methods will be applied by the Biomedical Engineering students. The aim is to predict whether signals between certain seconds are generated by the person's moments of relaxation or moments of concentration. Thus, Can Yilmaz, will build a ML model for to understand the signals and try model to predict data. Also, Muharrem Asan, will build an interface application to increase usability of the system by any person easly.

1.1. Identification of the need

Due to the limitations of the EEG technique we can obtain very limited information about the brain. However by utilizing several signal processing techniques we can obtain a clean data and by using machine learning techniques, we can make the most sense from the signals we gathered. This is an under-development research topic around the globe. Many researches have been made and many are needed to be made for a better understanding of the human brain. We saw an opportunity in the literature to make our contribution to the field.

1.2. Definition of the problem

Scientists have been doing many types of researches to understand how does the human brain work. That is still not a fully understood concept by science. EEG have been around since decades, but making sense of the signals gathered with this technique gave us little clue about the activities of the brain. Thanks to the latest developments in the field of computer science also new types of techniques have been developed for processing signals. One of the most important development of the field is the machine learning algorithms. Thanks to this method we can make hundreds of thousands of measurements and run a computer program to make sense of these signals in a very short period of time. Without this technology we would have to look at the plotted signals by eye and try to understand each signal one by one to come to a conclusion. Figure 1 shows the different steps of our work.

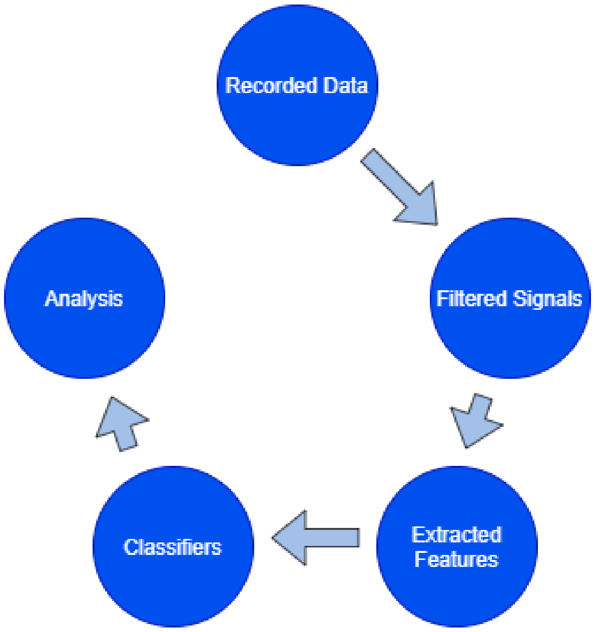


Figure 1. Schematic View of the System

In addition, Figure 2 shows the recorded data that have not been filtered yet. Displayes only the first 10 seconds of data with 14 different channels.

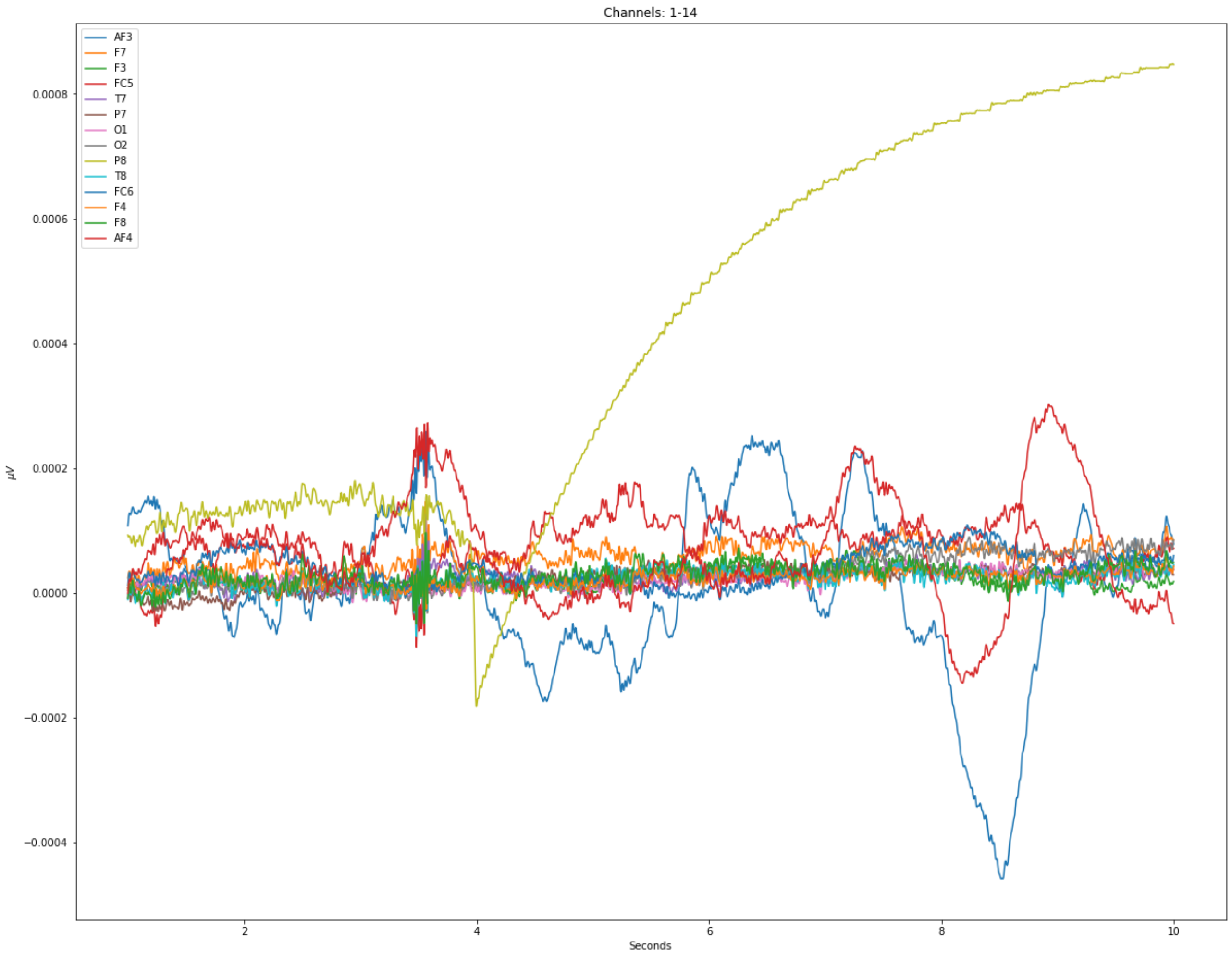


Figure 2. The Graph of Frequency by Seconds

1.2.1. Functional requirements

The system has two structures. One of them is filtering raw data and making it suitable for ML. Then, using ML methods on the data, also looking at their performance to train the most appropriate model. All models are automatically save by the system with their log files into the Results folder. The other one is to use the saved model for detection. At the detection process, the system predicts the data and visualizes it with graphs. Finally the main subject will be answered by looking at the red and green dots on the graph.

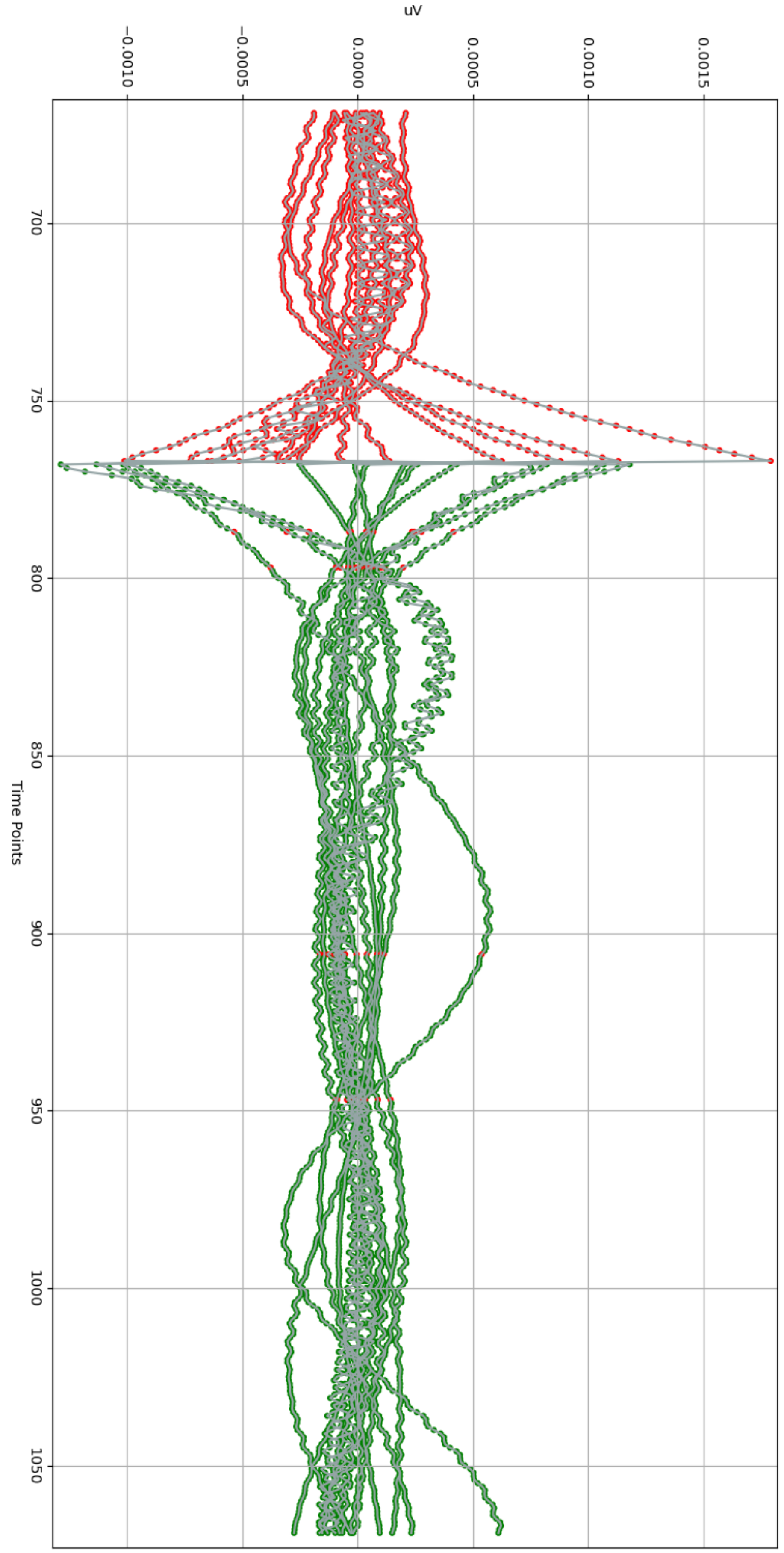


Figure 3. The Visualization of Test Data by Using Model

1.2.2. Performance requirements

In this project, more than 5 ML algorithms are used and the scientists can measure of their performances by trying all of them one by one. An example of trained model's result from the Results folder:

```
Model Name = Decision Tree

Date = May 24 2021
Time = 13:51:09

---SCORES---
Accuracy: 0.982646420824295
Precision: 0.9851235450275679
Recall: 0.9779210589844693
F1: 0.9813163654048797

---Confusion Matrix---
TP: 165
FP: 1
FN: 7
TN: 288
```

Figure 4. Result Logs

This figure shows, this model has 98% accuracy with the data. The accuracy of the model can be increased by good preprocessing and proper preparation of the data.

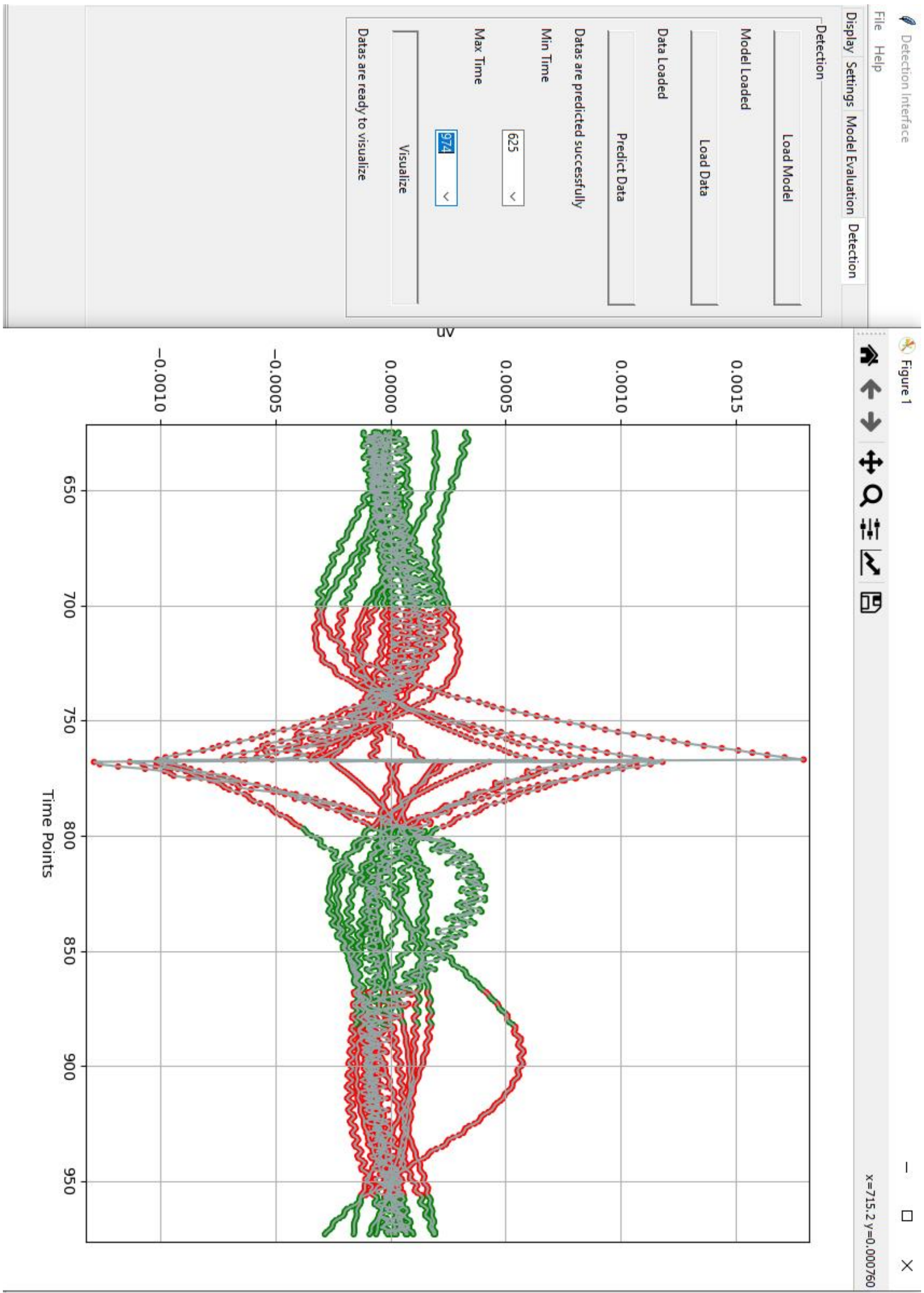


Figure 5. The Visualization of Predicted Data by Using Model

This figure shows, the predictions result is not good rather than the test prediction. This may be because ML can not make sense of the input data and it needs more data for building.

1.2.3. Constraints

On the physical side of the project, It is needed to use an EEG sensor for collecting sample data from different subjects for several sessions. Unfortunately, due to the pandemic conditions it is not possible to work with lots of subjects and do tests on them. For this reason, collected and shared data from the internet is used. Therefore, physical side of the project will not be evaluated by its effects.

On the software side of the project, it is expected that a program will be created and this program will analyze the data and it will reveal results. This program will only run on computers and its effect on the environment are examined:

- **Environmental Effects:** Computers work with electricity power.
- **Social Effects:** People can waste time dealing with a lot of data or there may be parts that get overlooked because there is too much data. Since the system can complete the given job within a few seconds, it positively affects the social lives of people.
- **Economic Effects:** Since the system's uptime will shorten and people will not spend too much time analyzing the data, it is anticipated that the budget allocated to research will be used less. Also, some advanced expensive programs will be less needed for EEG analysis.
- **Ethical Effects:** The system will provide the first rule of the code of ethics that is published by IEEE because it aims to decrease the time to spend and increase the rate of success of the human work.
- **Health and Safety:** The project is created to analyze and reveal an answer with an accuracy. It is necessary to find a method with more accuracy than others to decrease the error rate. But the whole system is designed to find the secret inside of the human brain and therefore it is possible to understand that the system increase human health and safety.

1.3. Conceptual solutions

In this project, using ML with EEG, a system will try to predict people's emotional states. In ML, appropriate algorithms should be chosen depending on the projects. But this system has many classifiers in it and ready to use according to their performances with the data. Therefore classification algorithms used in this project because the desired result in the project is to understand whether the person is in a focused state or in a relaxed state. Finally, the predicted data will be displayed to the user with an understandable graph.

1.3.1. Literature Review

Humans have been doing research about the brain for a long time, but still do not fully understand the brain. In these studies, an EEG device is generally used to examine the brain, but although this process has a cost and margin of error, the research can take a long time. But during the Corona EEG device is not necessary and related the has found by the internet. However the preprocessing process and building a ML model is necessary and attention should be paid to how to do it.

[13] For example, in order to understand whether a person is happy, angry, tired or relaxed, the ratios of arousal and valence values are examined. LDA and SVM methods were used for these two values. [1] In another study, people were watched some videos and asked to fill out a questionnaire containin arousal, valence, familiarity and dominance values for the situations they experienced while watching and used 5 different classifiers,it was tried to predict exactly which emotions the person

felt. [6] In another study, people were asked 6 questions. Answers to these questions will be answered positively or negatively and each answer represents a feeling about the person. As a result of this proces, they want to guess what the heart rate is by looking at the emotion. Researchers use the classifier methods of classification together with the deep learning methods to test. As a resultm they choose the most accurate result. [14] In another study, people are shown a video and at the end of the video they are asked to give a value between 1 and 9 for the arousal, valence, dominance and liking values they feel. The methods used are SVM and RF because it is stated that these methods can learn quickly and correct predictions are made with small data.

In this proposal, [2] 4 sessions of 180 seconds each were held with 30 people. Each session was held at different times. During the first minute of the session, the person is asked to be confortable, their eyes closed and seated on a chair. On the second minute, the person encounters random pictures or questions on the screen. At the last minute, the person is asked to relax again and his/her eyes are closed. Later, these datas are cleaned and extracted features by BME students. In this way, it is ready before ML is implemented. Ready data is received by the SEN students and goes to the cleaning phase. Then, a total of 21600 seconds of data is divided into test and train parts. The methods to be used are SVM, RF, DT, LDA and NB methods. Because some of them make quite accurate results and some of them are very popular. Then the results are compared with each other and the appropriate method is selected. Finally, the project will be ready to detection.

1.3.2. Concepts

There are two different ways for the project to work and some scopes need to be kept in mind for it to finish successfully. To explain them; first way, after preprocessing process of the Biomedical Engineering students, the ML model can be build by Software Engineering students and the best proper model can be present as a product for the detection process. But second way, is giving this opportunity to the users and let them build their own ML models. Because they may need to improve the accuracy of the model and that can be possible if the given dataset is large and well prepared. Finally, a suitable model created by software engineering students will be given as a product.

Table 1 compares different conceptual solutions with respect to the four most important requirements; Concept 2 is chosen for this project due to it’s high features because in order for the project to continue to be used in the future, it was intended to be flexible and Concept 1 would be a simple product for the users.

	Concept 1	Concept 2
Cost	Low	Medium
Complexity	Medium	High

Performance	Medium	Medium
Features	Medium	High

Table 1. Comparison of the Two Conceptual Solutions

1.4. Physical architecture

Figure 6. Emotiv Epoc Flex 32-Channel

On the physical side of the project, we use an EEG device manufactured by EMOTIV. The model we are going to use is Emotiv FLEX 32-Channel. This device is carefully prepared and each electrode is placed to the right position on the Emotiv EasyCap for neglecting false placement errors. Color coded wires gets connected to the controller on its' each sides. Then the cap is placed to the subject carefully and conductive gel is applied to each electrode one by one. This process usually takes 3 to 5 mins. After checking everything is wired up correctly, we can connect the controller to the computer. By using the EmotivPRO software we can check if the device is working as expected. Now we can start our session depending on the research and log the data of the session in ".edf" file format.

On the software side of the project, it starts with a preparation process after receiving the data with extracted features by Biomedical Engineering students. In this process, the data are changed and fragmented to work on a model. Then, the models to be worked on are put into the training process one by one. A prediction is provided with the trained data and all models go through the same process. As a result, a model with best accuracy is selected and the system becomes ready for prediction.

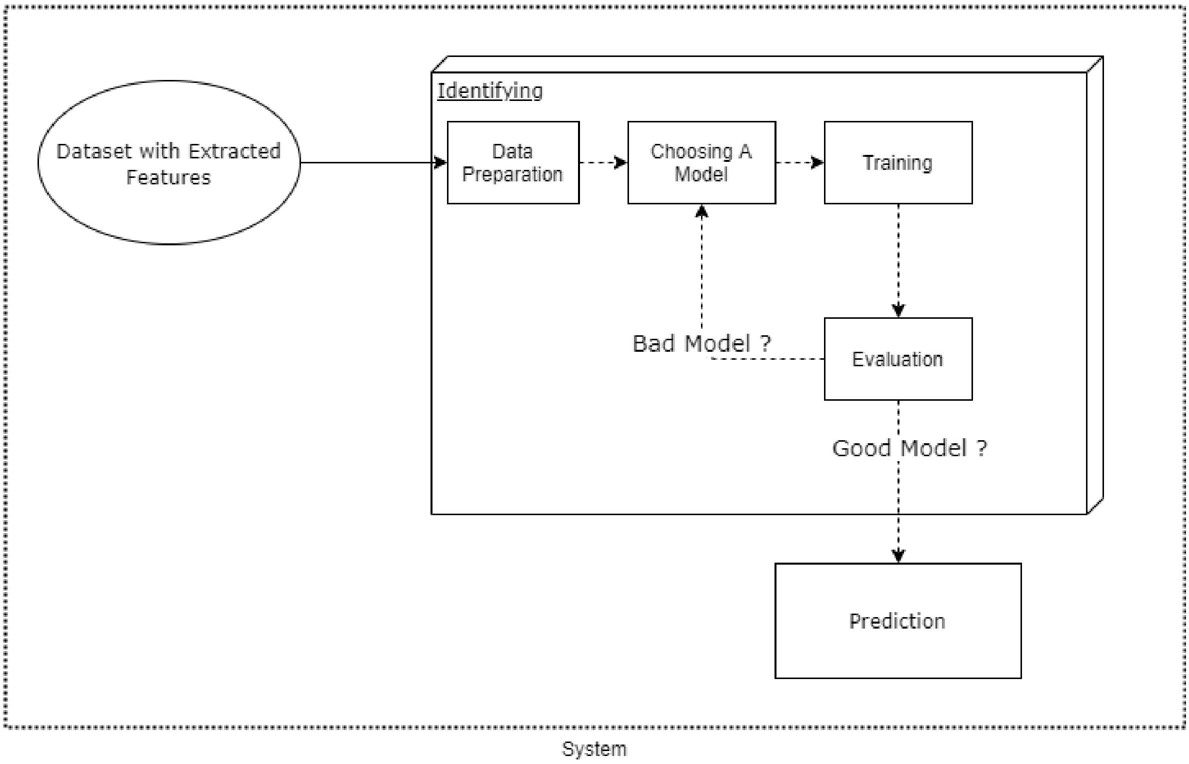


Figure 7. Physical Structure of the Software

2. WORK PLAN

2.1. Work Breakdown Structure (WBS)

1. System
    - 1.1. Data
        - 1.1.1. Data Gathering
        - 1.1.2. BME-Preprocessing
            - 1.1.2.1. Event Markers
            - 1.1.2.2. Plotting the Raw Data
            - 1.1.2.3. Filtering the Signal
            - 1.1.2.4. ICA Decomposing
            - 1.1.2.5. Examination by Eye
            - 1.1.2.6. Feature Extraction
    - 1.2. Software
        - 1.2.1. Dataset Preprocessing
            - 1.2.1.1. Data Preprocessing
                - 1.2.1.1.1. Converting EDF to Array
                - 1.2.1.1.2. Data Splitting
            - 1.2.1.2. Data Visualization
        - 1.2.2. Modeling
            - 1.2.2.1. Searching Models
            - 1.2.2.2. Assembling and Coding
            - 1.2.2.3. Evaluating
        - 1.2.3. Storing
            - 1.2.3.1. Saving Models
            - 1.2.3.2. Saving Log Files
    - 1.3. Interface
        - 1.3.1. Design
            - 1.3.1.1. Layouts
            - 1.3.1.2. Screens
        - 1.3.2. Implementation of Coding
        - 1.3.3. Testing

Figure 8. Work Breakdown Structure

2.2. Responsibility Matrix (RM)

Tasks	Ozan	Bensu	Can	Muharrem
Data Gathering	C	C	-	-
Event Markers	S	R	-	-
Plotting the Raw Data	R	S	-	-
Filtering the Signal	R	S	-	-
ICA Decomposing	S	R	-	-
Examination by Eye	R	S	-	-
Feature Extraction	S	R	-	-
Data Preprocessing	-	-	R	-
Data Visualization	-	-	R	-
Modeling	-	-	R	S
Storing	-	-	S	R
Design	-	-	S	R
Implementation	-	-	R	S
Testing	-	-	S	R
Planning	S	R	R	S
Reporting	S	R	R	S
Integration	-	S	R	-

Table 2. Responsibility Matrix

2.3. Project Network (PN)

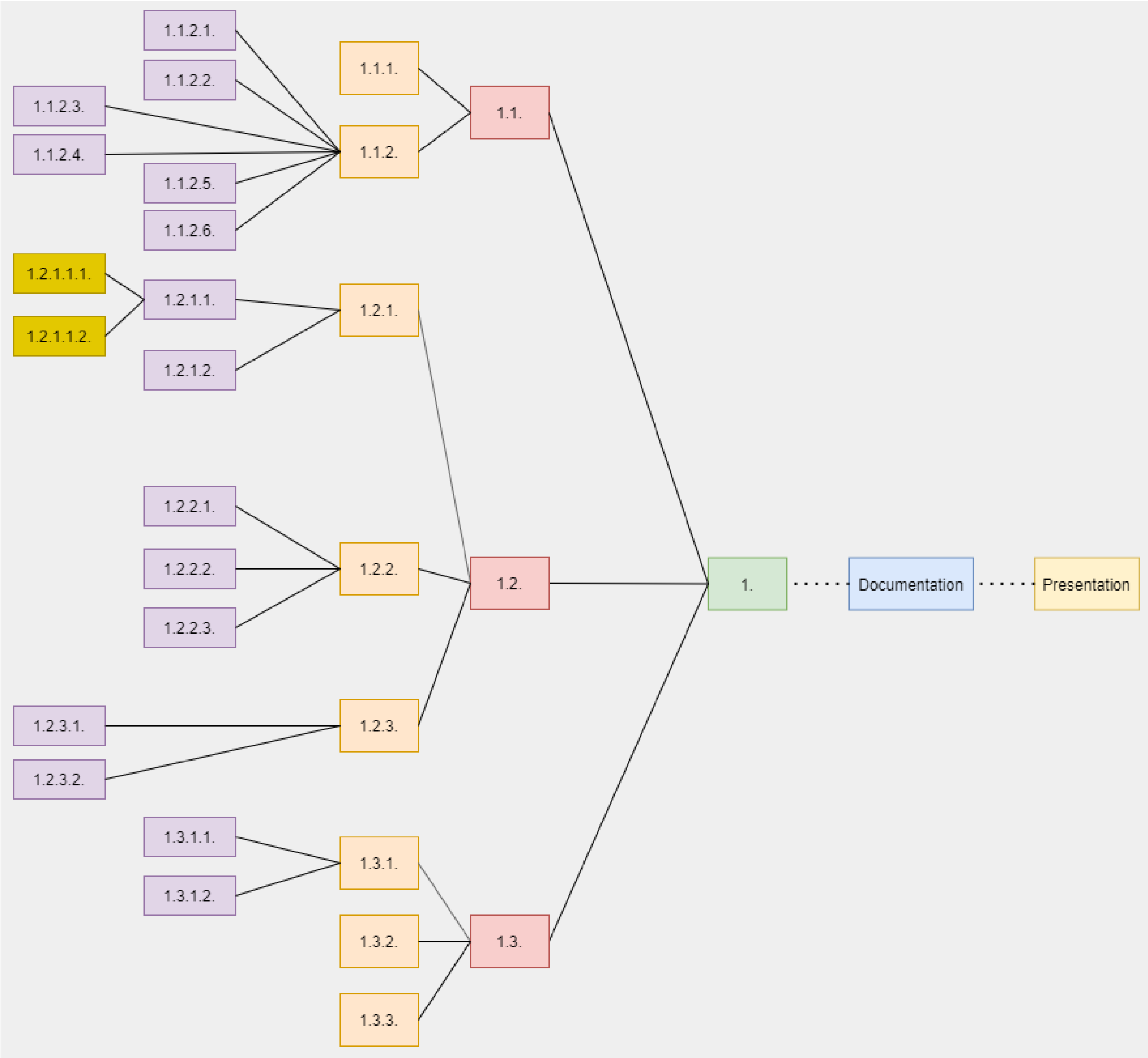


Figure 9. The Project Network.

2.4. Gantt chart

<table>
<tr><th>Week</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>Ozan</td><td colspan="2">1.1.1.</td><td></td><td colspan="2">1.1.2.2.</td><td colspan="2">1.1.2.3.</td><td></td></tr>
<tr><td>Bensu</td><td></td><td></td><td colspan="2">1.1.2.1.</td><td></td><td></td><td colspan="2">1.1.2.4.</td></tr>
<tr><td>Can</td><td>1.2.1.1.</td><td>1.2.1.2.</td><td colspan="2">1.2.2.</td><td colspan="2">1.2.3.2.</td><td colspan="2">1.3.2.</td></tr>
<tr><td>Muharrem</td><td colspan="2">1.3.1.1.</td><td colspan="2">1.3.1.2.</td><td>1.2.3.1.</td><td colspan="3">1.3.2.</td></tr>
<tr><td>Integration</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Verification</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Documentation</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Presentation</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th>Week</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>Ozan</td><td colspan="2">1.1.2.5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Bensu</td><td></td><td>1.1.2.5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Can</td><td>1.3.2.</td><td>1.3.3.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Muharrem</td><td>1.3.2.</td><td>1.3.3.</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Integration</td><td></td><td></td><td colspan="2"></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Verification</td><td></td><td></td><td></td><td></td><td colspan="2">1</td><td></td><td></td></tr>
<tr><td>Documentation</td><td></td><td></td><td></td><td></td><td></td><td>Draft</td><td>Report</td><td></td></tr>
<tr><td>Presentation</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Draft</td><td>Presentation</td></tr>
</table>

Table 3. Gantt Chart

2.6. Risk assessment

RISK LEVEL		Severity of the event on the project success		
		Minor	Moderate	Major
Probability of the event occuring	Unlikely	VERY LOW	LOW	MEDIUM
	Possible	LOW	MEDIUM	HIGH
	Likely	MEDIUM	HIGH	VERY HIGH

VERY LOW	This event is very low risk and so does not require any plan for mitigation. In the unlikely event that it does occur there will be only a minor effect on the project.
LOW	This event is low-risk; a preliminary study on a plan of action to recover from the event can be performed and noted.
MEDIUM	This event presents a signficant risk; a plan of action to recover from it should be made and resources sourced in advance.
HIGH	This event presents a very signficant risk. Consider changing the product design/project plan to reduce the risk; else a plan of action for recovery should be made and resources sourced in advance.
VERY HIGH	**This is an unacceptable risk. The product design/project plan must be changed to reduce the risk to an acceptable level.**

Table 4. Risk matrix

Failure Event	Probability	Severity	Risk Level	Plan of Action
Low accuracy	Likely	Major	VERY HIGH	Selecting another model or changing splitted data
Lack of data	Possible	Moderate	MEDIUM	Finding new dataset
Improper or incomplate training	Unlikely	Major	MEDIUM	Wrong coding, fixing the issues, Wrong data splitting

Table 5. Risk Assessment

3. SUB-SYSTEMS

3.1. Biomedical Engineering

3.1.1. Definition of the Problem

During the sessions, our goal is to keep the sensor connected and ensure that each electrode is in contact with the subject's sculp. To avoid any kind of disturbances that might affect the signals, we need to keep the session room clean, quiet, and dark (depending on the research goal).

3.1.2. Overview of Technologies and Method of Analysis

According to the research we have done, preprocessing has been performed many times in MATLAB using the EEGLAB toolbox, but EEGLAB was not preferred to extract feature extractions. In this project, we wanted to do all the operations from a single toolbox. When we realized the research we made and the plan we designed, we realized that a single toolbox was not enough for all this operation. Mentioned on the official tutorial page of EEGLAB; We completed our work more easily by adding ERPLAB, Cleanline, ICLabel, Biosig extensions.

3.1.3. Limitations

One of the challenges was the lack of time as the project had to be completed before the deadline. For the biomedical engineering department; having minimal coding experience was also an issue as my bandmate and me were only taught the basics of the computer language used. In order to practice, the group had to look at a large number of tutorials online.

Due to the pandemic conditions, we had to work at home and the computers we used did not run some necessary systems very slowly or at all, which left us in a difficult situation.

3.1.4. Description of Methods and Materials

The product will not have any direct physical interaction with the user because it is a software that uses image processing in EEG data to detect emotional state. Errors that can happen during EEG recording still exist.

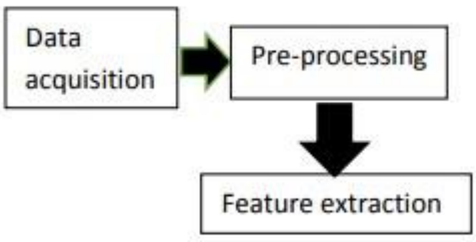


Figure 10. Physical Structure

3.1.4.1. Data Gathering

Following dataset has been collected and shared online with the public for scientific purposes. This dataset gave us the opportunity to test our signal processing methods before working with any real life subjects. Information about the dataset are as follows:

Title: EEG dataset of Fusion relaxation and concentration moods
DOI: 10.17632/8c26dn6c7w.1
By: Ahmed Albasri
URL: https://data.mendeley.com/datasets/8c26dn6c7w/1
Data type: EDF
Device: EMOTIV EPOC+

Description of this data

1. **Aim:**
This dataset aims to provide open access of raw EEG signal to the general public. We believe that such fusion of human moods (Relaxation & concentration) shall increase scientific transparency and efficiency, promote the validation of published methods, and foster the development of new algorithms. In addition, publishing research data is becoming more important as public funding agencies are moving towards open research data requirements.
2. **Scenario:**
The proposed scenario adapted to acquire the brain EEG signals in two different mental status. First while subjects in a relaxed mood, and second in concentration mood. Both of these cognitive stimuli considers as self-induced motivation. The recording period continues till three minutes for each session, as follows:
-In the first minute, the subject is asked to relax and sit on a handed chair with eye open looking at a black screen computer of about 40cm far. Until hearing beep sound.
-In the second minute, a random picture appear on the screen contain a question or some different objects. The subject is asked to solve the problem or to find common relation links all these objects together.
-In last minute, the subject is asked to close his/her eyes and relax again until the beep sound.
3. **Sessions:**
Fore sessions were recorded for each subject. Such that, first two sessions are done on the same day with 1-2 hours interval, and remaining sessions are done after 2-3 days in the same way. The reason behind this separation is to avoid medium term influences that may subjects have. Each session continues for three minutes. The total recording time for each subject equal to 720 seconds. A small program designed to control the timing and recording procedure of the sessions.
4. **Numbering system:**
The numbering system is formatted to include both subject enrollment number and trials. First four characters represent the subject number, where last three characters represent the session record number. For example (S001E03) indicate 1st subject and 3rd recording session.
5. **Artifacts:**
In this experiment, we notice that some subjects accidentally generated internal artifacts. Therefore we intentionally continue recording their brain signals to provide more realistic condition to the experiment and also provide a role for the artifact removal techniques in the pre-processing phase.
6. **Data recording:**
EEG raw data recorded using EMOTIV EPOC+ device with 14 channels (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF42), plus References in the CMS/DRL noise cancellation configuration P3/P4 locations. The signals were sampled with 250 SPS.
7. **Sample space:**
The sample space consists of 30 participants (56.6% male and 43.3% female) with ages of 18-40 years. The subjects do not suffer(ing/ed) from any brain problems (mentally or physiologically). 33% of the subjects were smokers and 3% of them were alcoholics. All the subjects are well educated and have at least B.S degree.

Figure 11. Data Gathering Informations

3.1.4.2. Preprocessing

For processing the gathered data we used a Matlab plugin software specifically developed for EEG signal processing called EEGLAB. We used the software versions "Matlab - R2021a" and

“EEGLAB - 2020.0”.

Many functions are included in EEGLAB that are required for signal analysis, processing, and visualization. In the image, you can see a screenshot of the plugin in action. When you import a dataset into EEGLAB, the interface displays important information about the dataset. The "Tools" section can be used to perform various operations on the signals, and the "Plot" section can be used to explore various visualization options.

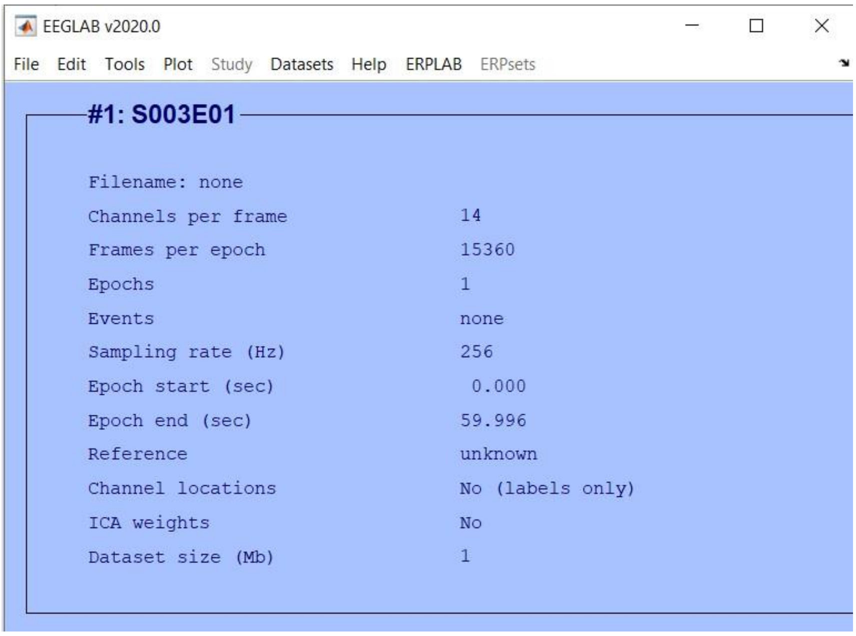


Figure 12. Screenshot of the Interface After Importing the Raw Dataset

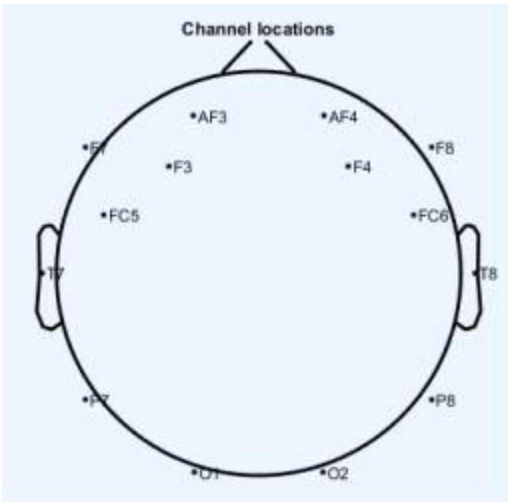


Figure 13. Plot Channel Locations

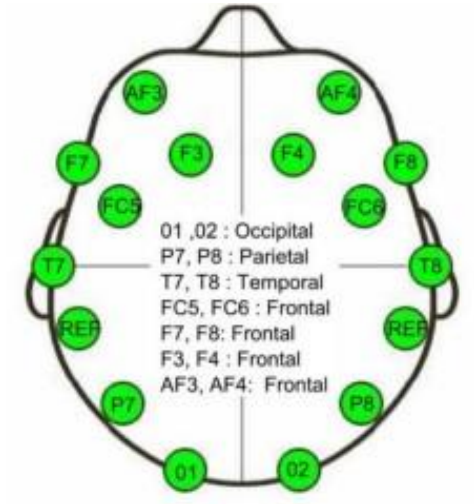


Figure 14. Channel Locations of Emotiv Sensor

In order to use the topographic visualization option, we must also import the channel locations. After importing channel locations, the interface's "Channel locations status" will change to "Yes." By comparing the topographic plot to the device manual, we can ensure that the channel locations have been imported correctly.

3.1.4.3. Filtering the Signal

We used high-pass and low-pass filters, but first we applied a low-pass filter and recorded the filtered data. Then we applied a high-pass filter and re-recorded the data. The order is not important, the high-pass filter can also be applied first. The reason we do this is because the filtering we enter separately into the system provides more precise and more reliable filtering. We used 40 Hz for the low-pass filter. We used 0.5 Hz for the high-pass filter.

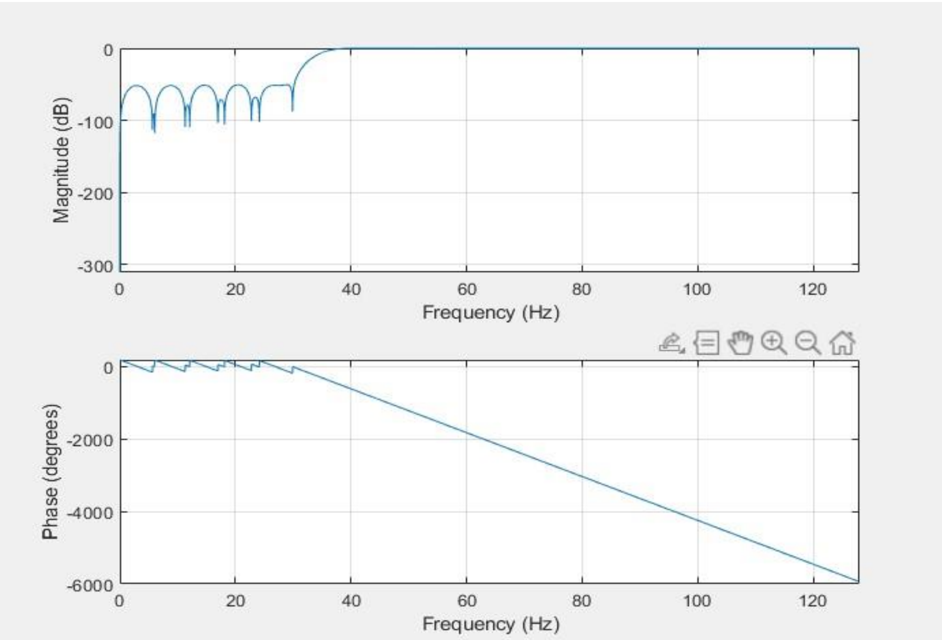


Figure 15. High-pass Filtered Data

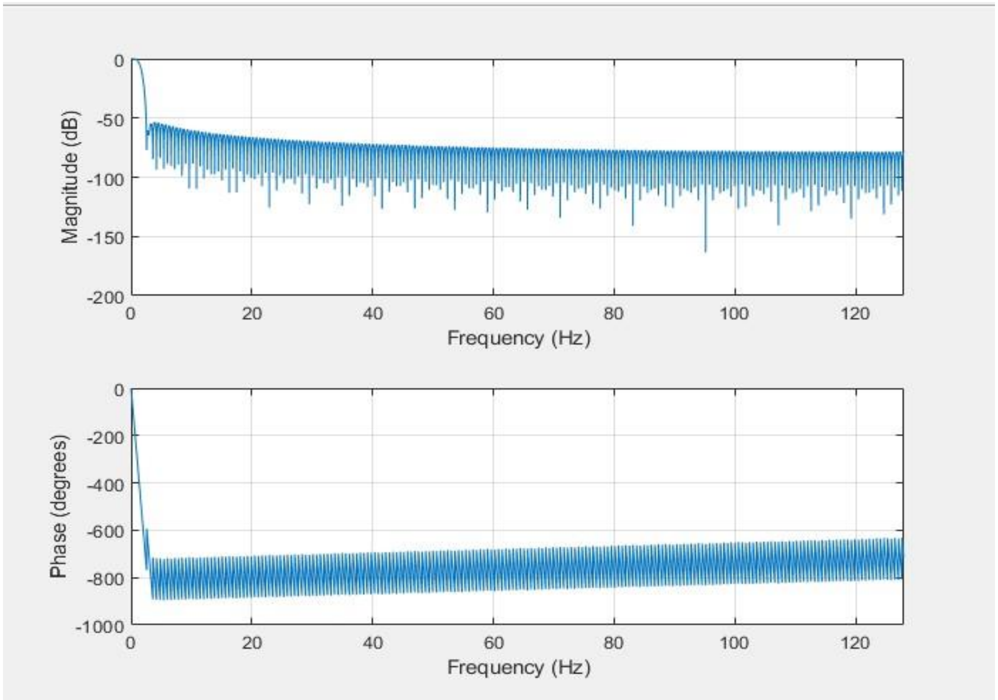


Figure 16. Low-pass Filtered Data

3.1.4.4. ICA Decomposing

To minimize the statistical dependency between certain components that are subsequently added to the signal, ICA offers a linear transformation. After applying ICA to EEG signals and comparing it with old EEG data, it has been shown that it significantly cleans the background noise and separates the signals as desired.

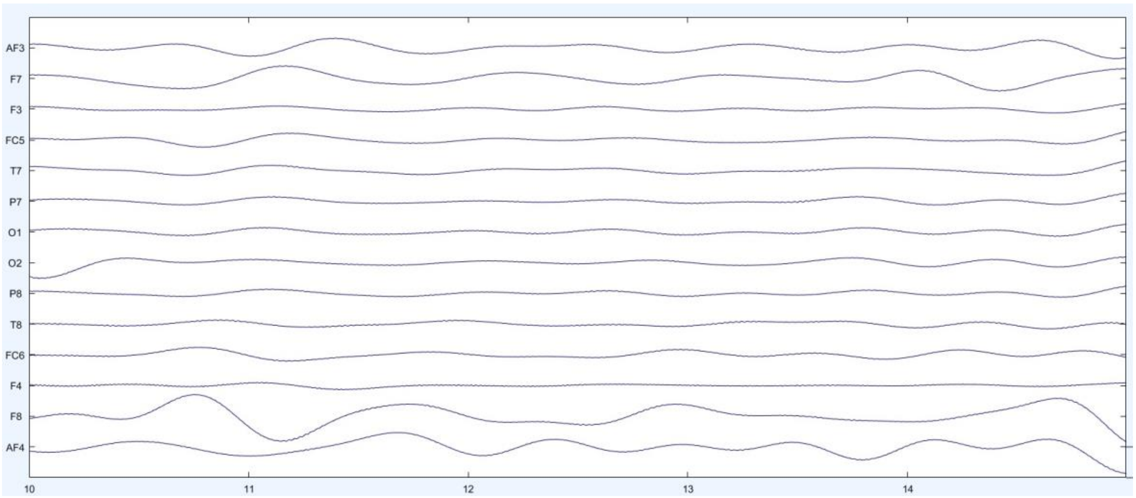
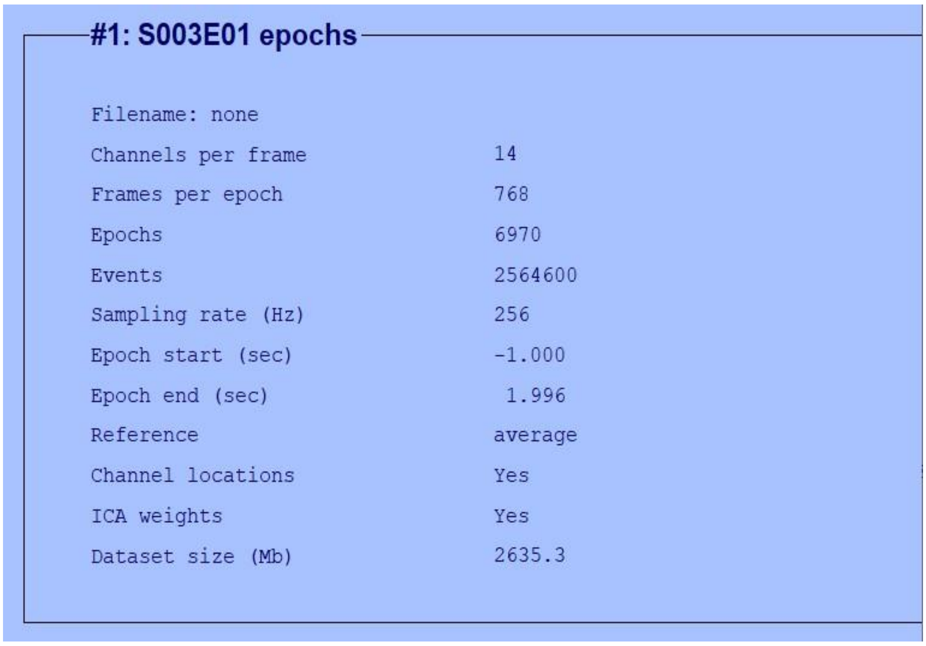


Figure 17. ICA Decomposed Data

3.1.4.5. Extracted Epoch

To analyze the event-related EEG dynamics of continuously recorded data, data epochs time-locked to events of interest must be extracted. We used the Extract epoch while keeping the default epoch limits. (between 1 and 2 seconds before and after the time lock event).



#1: S003E01 epochs

Filename: none	
Channels per frame	14
Frames per epoch	768
Epochs	6970
Events	2564600
Sampling rate (Hz)	256
Epoch start (sec)	-1.000
Epoch end (sec)	1.996
Reference	average
Channel locations	Yes
ICA weights	Yes
Dataset size (Mb)	2635.3

Figure 18. Extracted Epoch Dataset

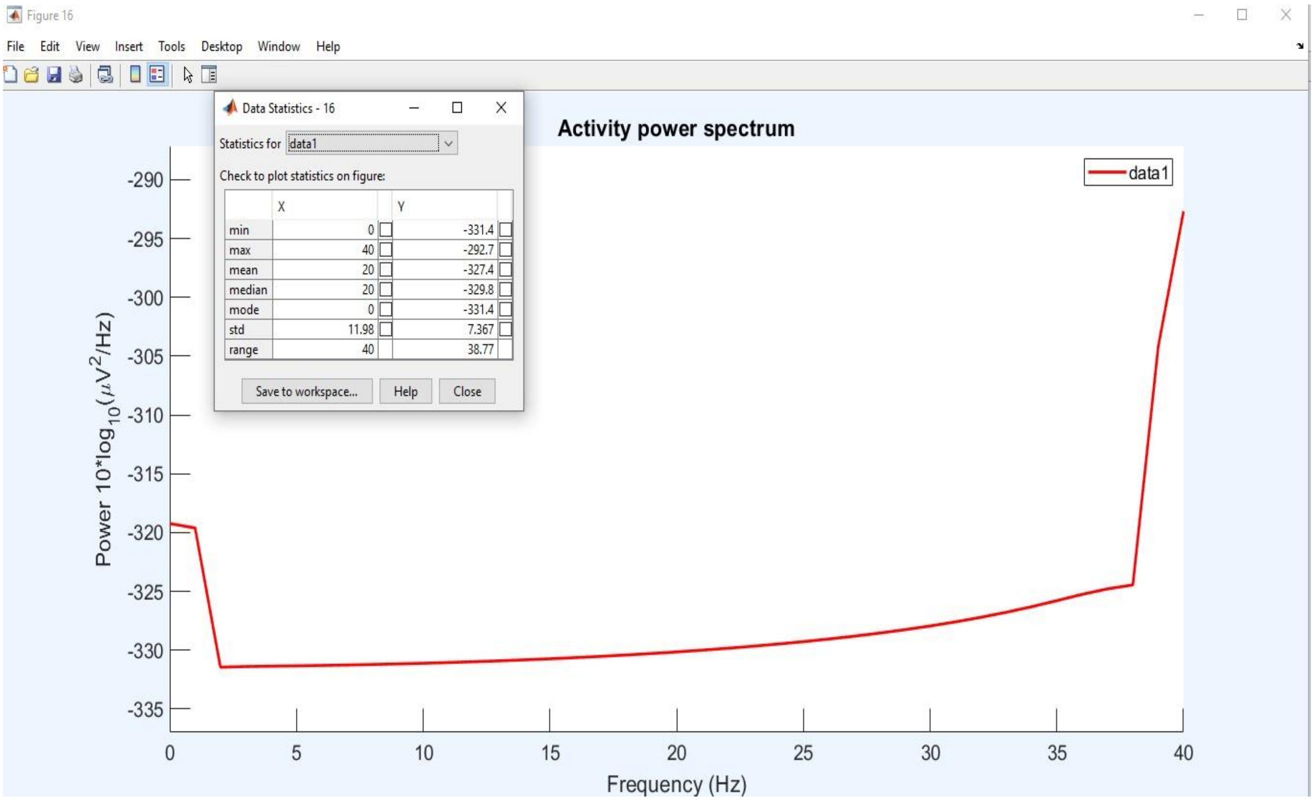
3.1.4.6. Feature Extraction

In order to extract feature extraction in a recording of 180 seconds, the recordings were analyzed by dividing them into three intervals of 60 seconds. In the first 60 seconds, the subject is in a state of relaxation. In the second 60 seconds, the subject is in concentration mode. In the last 60 seconds, he went into relaxation mode again. Therefore, it is impossible to take all 180 seconds and extract features for alpha(8-12 Hz) and beta(12.5-30 Hz) waves. For feature extraction, the FFT is an important and effective tool. From simple real and complex number arithmetic to group theory, the FFT algorithm is involved in a wide range of mathematical operations. The data becomes very clean at this point. Since we are looking for Alpha and Beta signals; to make them detectable, in the next step FFT is applied and Time series analysis is done and the result is channel by channel mode, median, mean and standard deviation. However, what matters is the Y-axis statistics.

	Delta Wave	*Theta Wave*	*Beta Wave*	*Alpha Wave*
Amplitude	*3.741*	*5.3226*	*8.2812*	*9.521*
Frequency	*4.6523*	*7.932*	*13.024*	*11.82*
Power	*0.52 uV^2*	*1.82 uV^2*	*12.3 uV^2*	*12.86 uV^2*
Mean	*1.324*	*3.961*	*7.6538*	*7.83*
Standard Deviation	*5.4721*	*3.7842*	*11.475*	*10.418*
Median	*3.587*	*5.26*	*23.15*	*32.278*
Mode	*-2.23*	*-3.147*	*-11.335*	*-15.11*

Table 6. Feature Extraction of EEG Signal

Different methods for extracting EEG signal features will be used. The brain's activity is separated into four frequency bands: Delta, Theta, Alpha, and Beta. In this project, the mean, median, standard deviation and mode were estimated by using FFT.



	X	Y
min	0	-331.4
max	40	-292.7
mean	20	-327.4
median	20	-329.8
mode	0	-331.4
std	11.98	7.367
range	40	38.77

Figure 19. Statistics of One Channel on the X and Y Axis

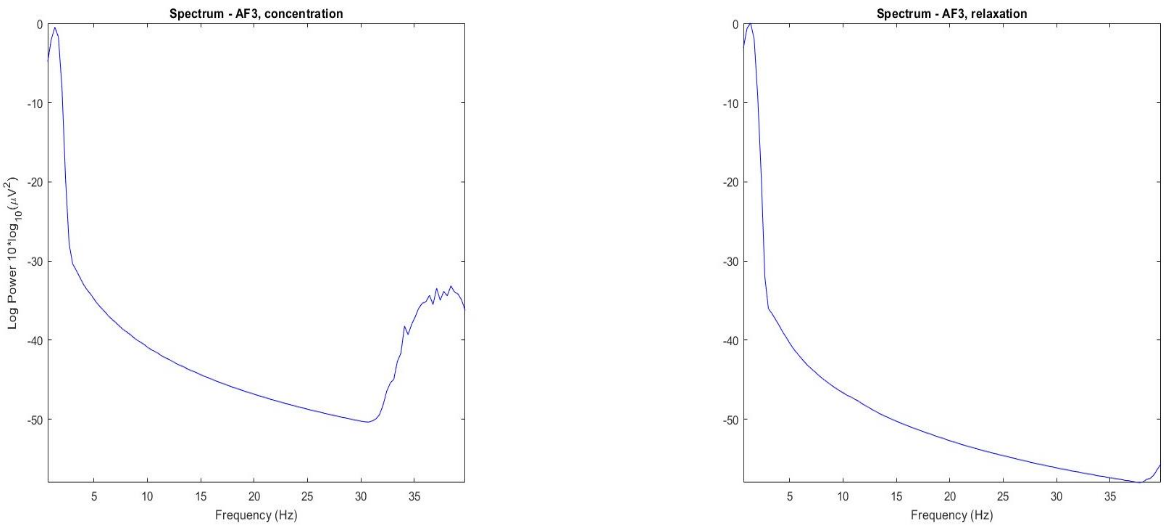


Figure 20. Analysis of Electrical Activity with Frequency

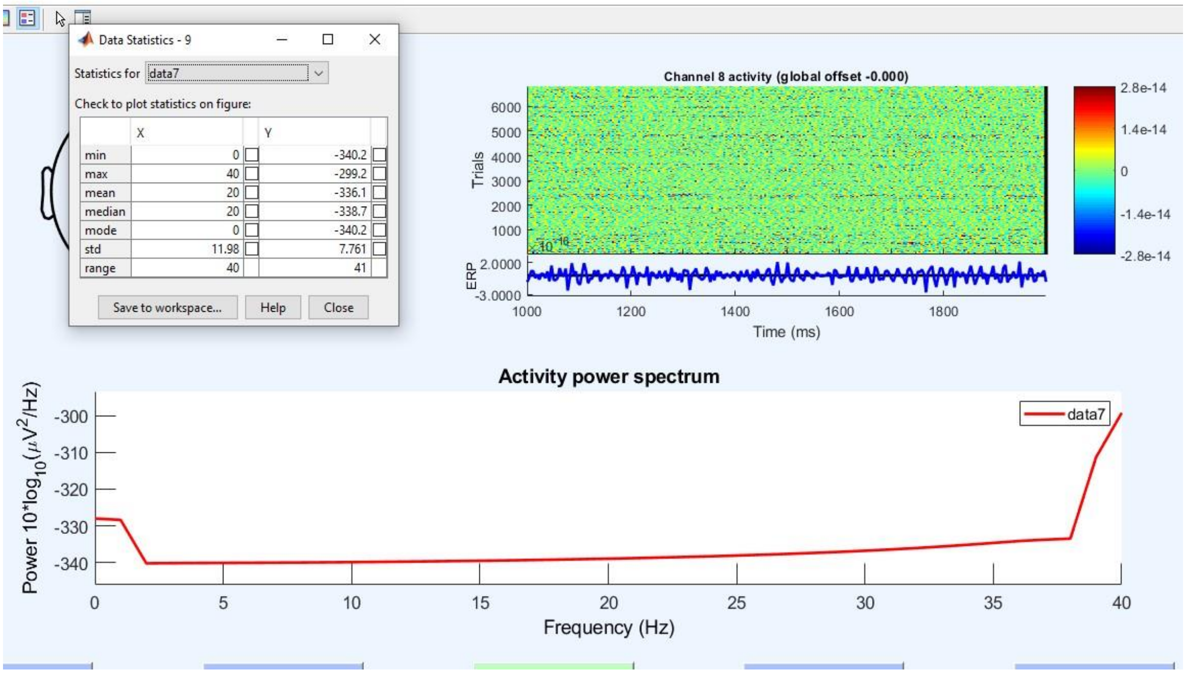


Figure 21. Statistics of Channel 8 and ERP Image

Unlike the draft of the project during the construction phase, the STUDY section and IClabel were used in the EEGLAB toolbox. The STUDY section allowed us to analyze and compare three 60-second data at the same time.

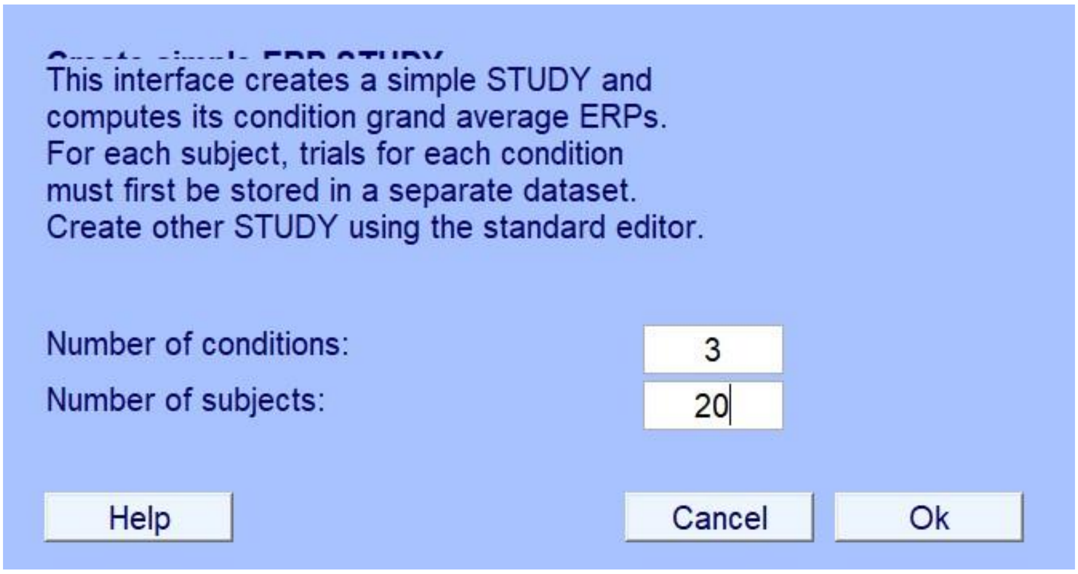


Figure 22. Creat Simple Study Pop-up Screen

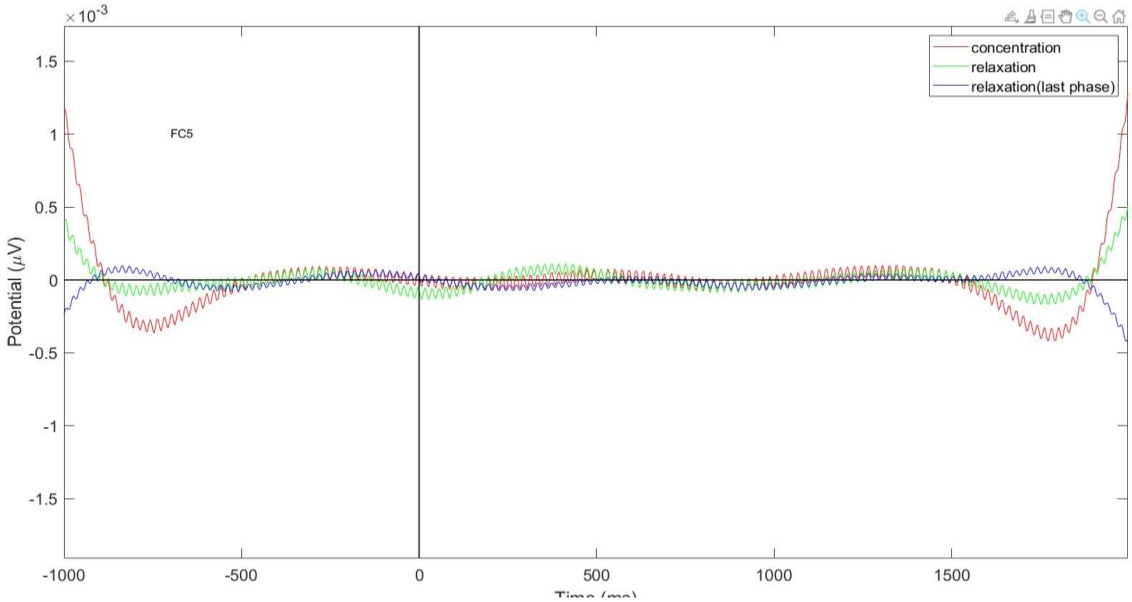


Figure 23. Electrical Graph of FC5 Channel Divided into Three Parts

We used the ICLabel tool to classify Label components. The ICLabel tool provides a topographic display of artifacts. It displays blinking, muscle movements and brain signals as a percentage.

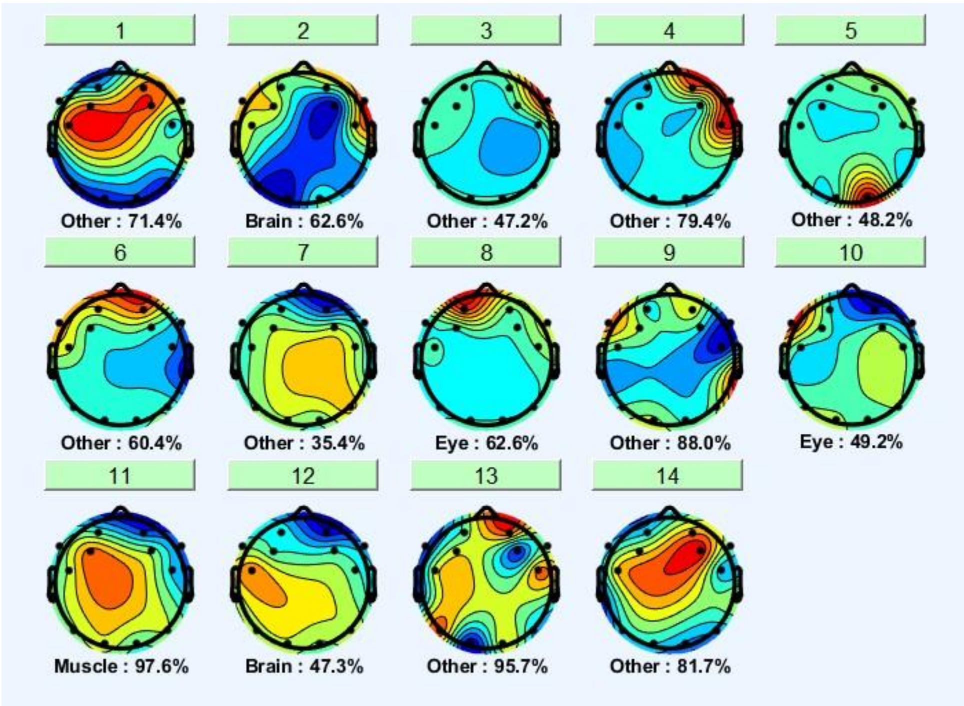


Figure 24. Topographical Image of Label Components

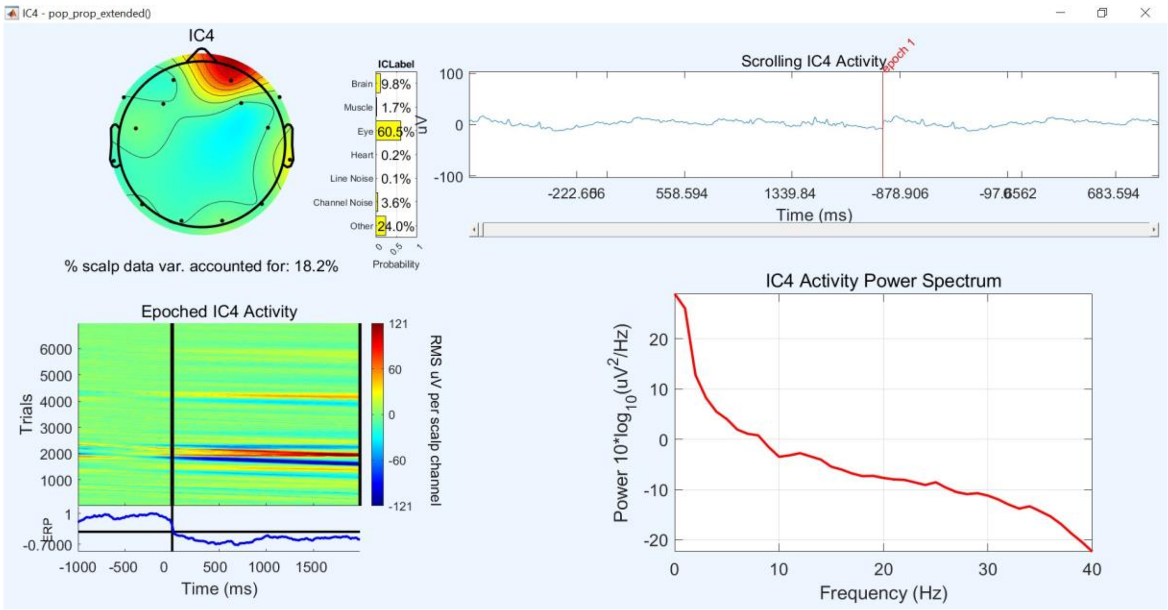


Figure 25. IClabel Applied, Percentage Representation of Brain Activities

3.1.5. Verification and Validation

The product is a program written through MATLAB and Phyton that can classify an EEG data into relaxation and concentration through signal processing and ML algorithms. The verification process will take place by testing the program with previously classified EEG signal data, and it will be verified whether the classification made by the program matches the previously known mood study.

3.1.6. Evaluation of Results

We conveyed the data we had passed through the processes described earlier to our group friends in the software engineering department, and the result definitely indicates that the test result was successful.

3.2. Software Engineering

3.2.1. Requirements

3.2.1.1. Functional Requirements

Funtional Requirement Name	**Description**
Importing Data	User should be able to enter Data
Adjusting Settings	User should be able to specify data test ratio
Specifying Classifier	User can select ML algorithm by Combobox Widget
Model Building	User can build the model from the Evaluation Screen

Saving Model	After Model building process, it can automatically saved by the System
Result Analyze	After Model building process, model's performance can be seen, also use can visualize it by a graph
Detection Process	User should be able to enter a data and a model, also prediction can be clickable and the results can be visualize by a graph

Table 7. Functional Requirements of Software

3.2.1.2. Nonfunctional Requirements

Performance And Scalability

There are some critical processes that will affect the speed of the application. These are when training the model and predicting the data. However, since the data is scaled and thread functions are used, the application experiences the minimum possible level of slowness. However, according to the ML algorithm, no precise information can be given since this time may change.

Usability

The application was created with simple interfaces for users to adapt quickly and separated the main functions of the application. However, in order for the person to use the program, the user basically needs to know about ML processes. If the conditions are done, the user can get the desired results quickly.

3.2.2. Technologies and methods

The project is made with Python 3.8 version, because this language has a huge library and made by community of Python therefore the libraries are several and easy to find. Tkinter library is used to create interfaces and layouts for a better user experiences. To analyze the data in the prediction part, a 30-person dataset consisting of 4 sessions for each was used and exported as mdf files after extracting different features of part of the project. To understand the edf files, "EDFbrowser" application is used. The Python libraries are installed on Pycharm virtually. These libraries used were mne, matplotlib, pandas, numpy, tkinter, sklearn to name a few. The example screens for the illustration process before creating the interface part was made by "Draw.io" application. The Project produces on Windows 10 with 2004 version using 7th generation intel core i7 processor with 6GB RAM.

3.2.3. Conceptualization

3.2.3.1 User Interface

The user will have the option to select a database in the file menu and then there will be four frames to adjust the data until it is detect process.

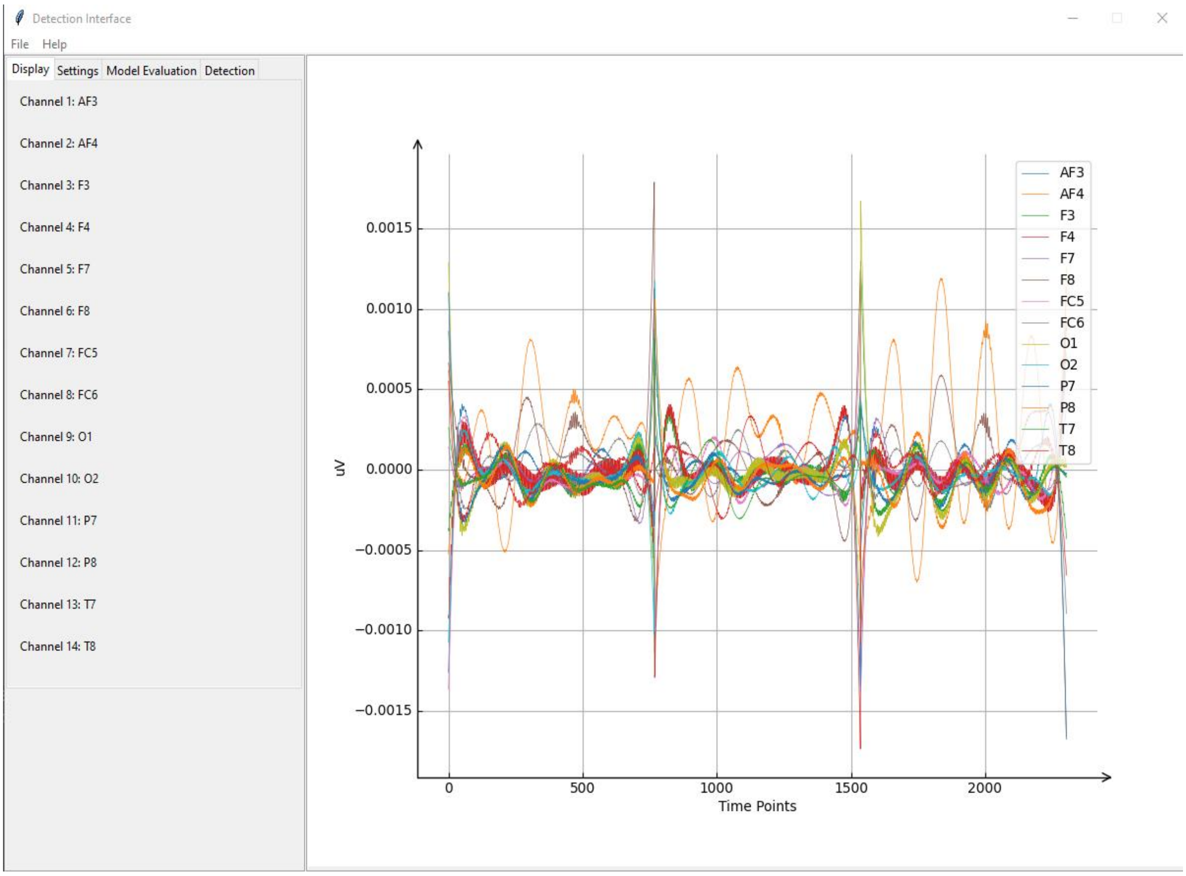


Figure 26. Display Layout

An example of the system's main interface in the figure. Users can display the related data's visual graph from the display frame.

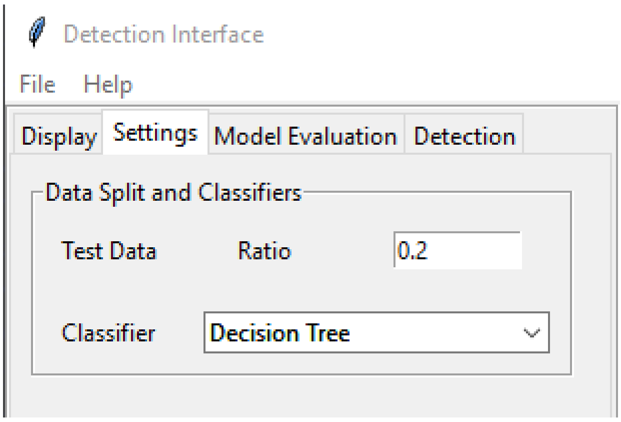


Figure 27. Settings Layout

Settings frame is to adjust the datas and filter methods with the apply button to save the changes.

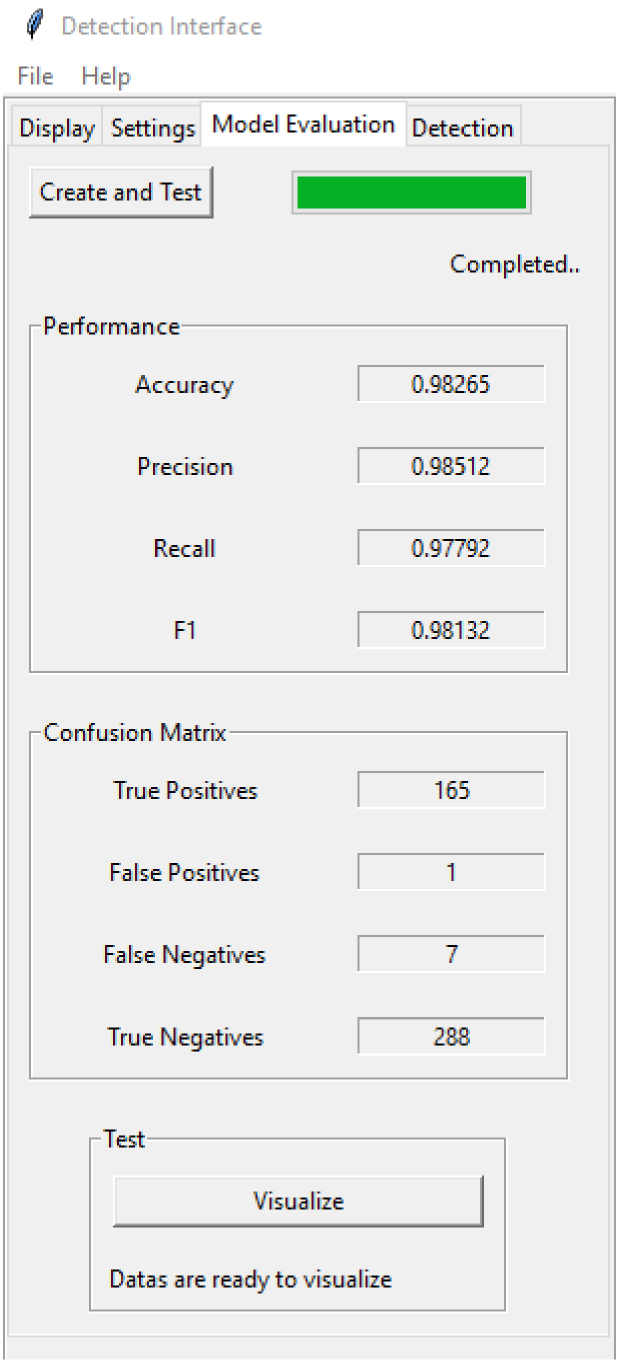


Figure 28. Model Evaluation Layout

In the model frame, users can select the preferred model type to create a new model. Also there are buttons to import another model or export to save the model.

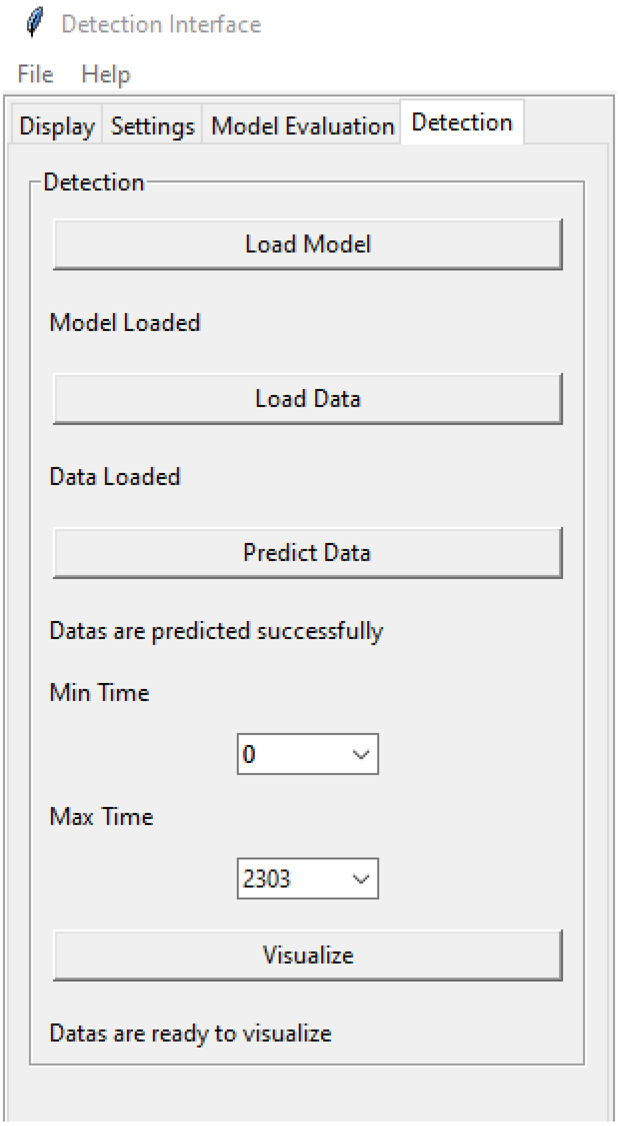


Figure 29. Detection Layout

After selecting a model, in the detection frame of the system, users can test it. The result will be displayed below the result container in the image.

3.2.3.2 Actor Glossary

User: A person who has access to the system, knowledge about EEG and machine learning basics.

System: Program that runs when requested by the user.

3.2.3.3 Use-case Glossary

Use-case Name	Description	Participating Actors

Importing Data	User imports dataset from specific location	User, System
Adjusting Settings	User can manage the data and selects classifier before creating a model.	User
Model Evaluation	User can create a model and system gives feedback after model created.	User, System
Detection Frame	User selects model and data to prediction process, and after prediction user can visualize results in a time rate.	User, System

Table 8. Use-case Glossary

3.2.3.4 Use-case Scenarious

Use-case Name	**Description**	**Participating Actors**
Importing Data	User imports dataset from specific location	User, System
Adjusting Display Settings	User sets the preferred settings	User
Adjusting Settings of the Data	User can manage the data before creating a model.	User
Adjusting Model	User can import, export or create a model and system giving feedback while model is creating.	User, System
Sending a Test Data to the Model	After user sends a request to the system to start the process,system gives a detailed feedback.	User, System

Table 9. Use-case Scenarios of Importing Data

Use-case Name	Adjusting Settings
Use-case Description	User can manage the data and selects classifier before creating a model.
Actors	User
Pre-Condition	User has to give a proper test ratio for the data splitting

Post-Condition	System will have test and train datas before modelling
Normal Flow	Step1: User clicks the settings frame Step2: User selects a ratio for the test sections Step3: User selects a model type rom combobox widget
Alternate Flow	Step2: If there is an error in setting the ratio values, the user must provide appropriate values.
Business Rules	Some settings must be valid for the system requirements

Table 10. Use-case Scenarious of Adjusting Settings of the Data

Use-case Name	Model Evaluation
Use-case Description	User can create a model and system gives feedback after model created.
Actors	User, System
Post-Condition	System will have a model to analyse
Normal Flow	Step1: User clicks the model evaluation frame Step2: User clicks to create and test button Step3: System creates a model and gives a feedback when creating process is finished Step4: System saves the results to Results folder in the applicaiton folder. Step4: User clicks visualize button to see graph
Business Rules	Imported model and data must be appropriate for the system requirements

Table 11. Use-case Scenarious of Sending Data to the Model Evaluation

Use-case Name	Adjusting Detection Frame
Use-case Description	User selects model and data to prediction process, and after prediction user can visualize results in a time rate.
Actors	User, System
Pre-Condition	There must be a valid model and data before starting prediction
Post-Condition	System will give a prediction graph of the process
Normal Flow	Step1: User clicks to load model button

	Step2: User clicks to load data button Step3: User clicks to prediction button to predict data with selected model Step4: User writes specific time rate to visualize Step5: User clicks to visualize button
Alternate Flow	Step1: If there is no a valid model than user go previous frames and creates a new one Step2: If there is no a valid data than user need to find a new one Step3: If time rate values are not valid than user should select them from the combobox values. Step4: User clicks to visualize button
Business Rules	There must be a valid model, data and time rate values

Table 12. Use-case Scenarious of Detection Frame

3.2.3.5 Use-case Diagram

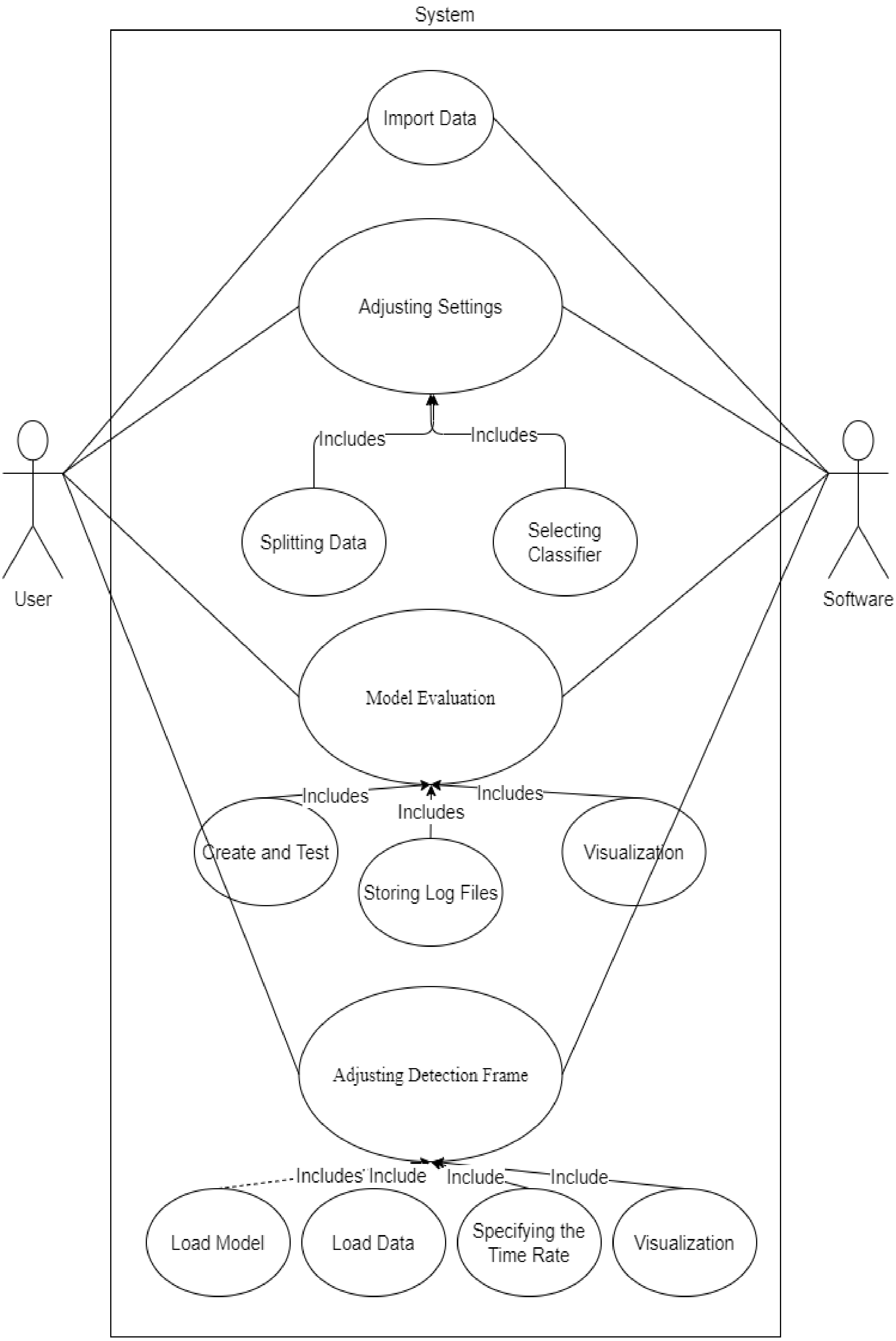


Figure 30. Use-case Diagram

3.2.3.6 Data Modeling

- What are the main tasks or functions that are performed by the actor?

The system has 5 functions for user actor to achieve a successful result. These are importing data, adjusting display settings, adjusting data settings, adjusting model settings and testing the model. System actor responds to some of the user actor's requirements for all the above functions.

- What system information will the actor acquire, produce or change?

The user can add a dataset to the system, make changes on the data attributes to be used in the dataset, add or remove this data, and create different models with model types.

- Will the actor have to inform the system about changes in the external environment?

If there is a later dataset change, it may need to be defined to the system again or new additions to some model or filter types may be required.

- What information does the actor desire from the system?

When the user adds data, it may be desirable to see the visual state of the data. Also, model creation can take a long time, and a loading bar symbolizing the formation may be requested by the user. Finally, after the data is predicted, a detailed return is requested.

- Does the actor wish to be informed about unexpected changes?

It is assumed that the user knows the basic information in the steps of machine learning. However, adding new unexpected data manipulation methods or adding new model types may affect user adaptation. The user must be informed about the changes made.

3.2.3.7 Activity Diagram

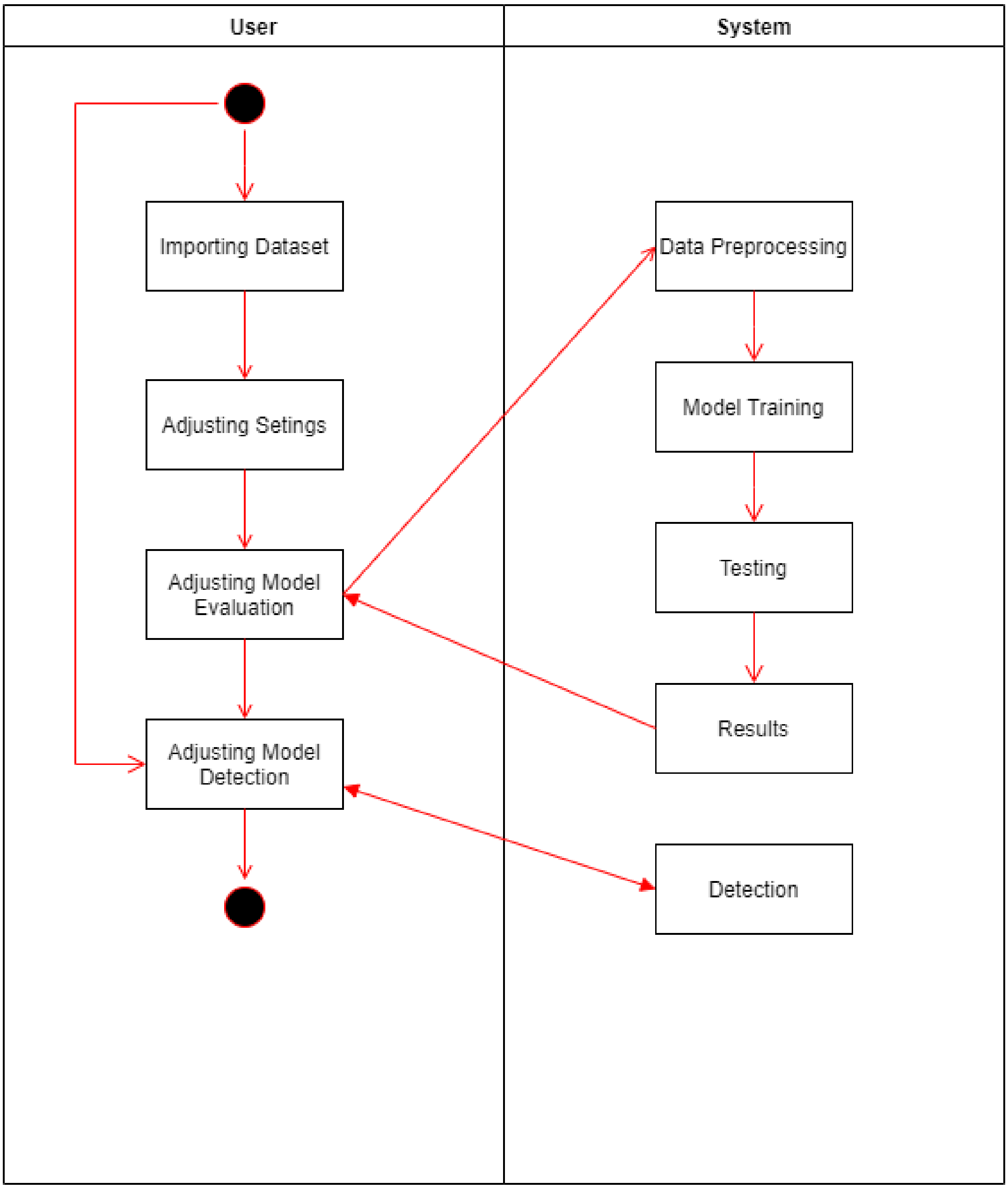


Figure 31. Activity Diagram

3.2.3.8 Data Analysis Process Diagram

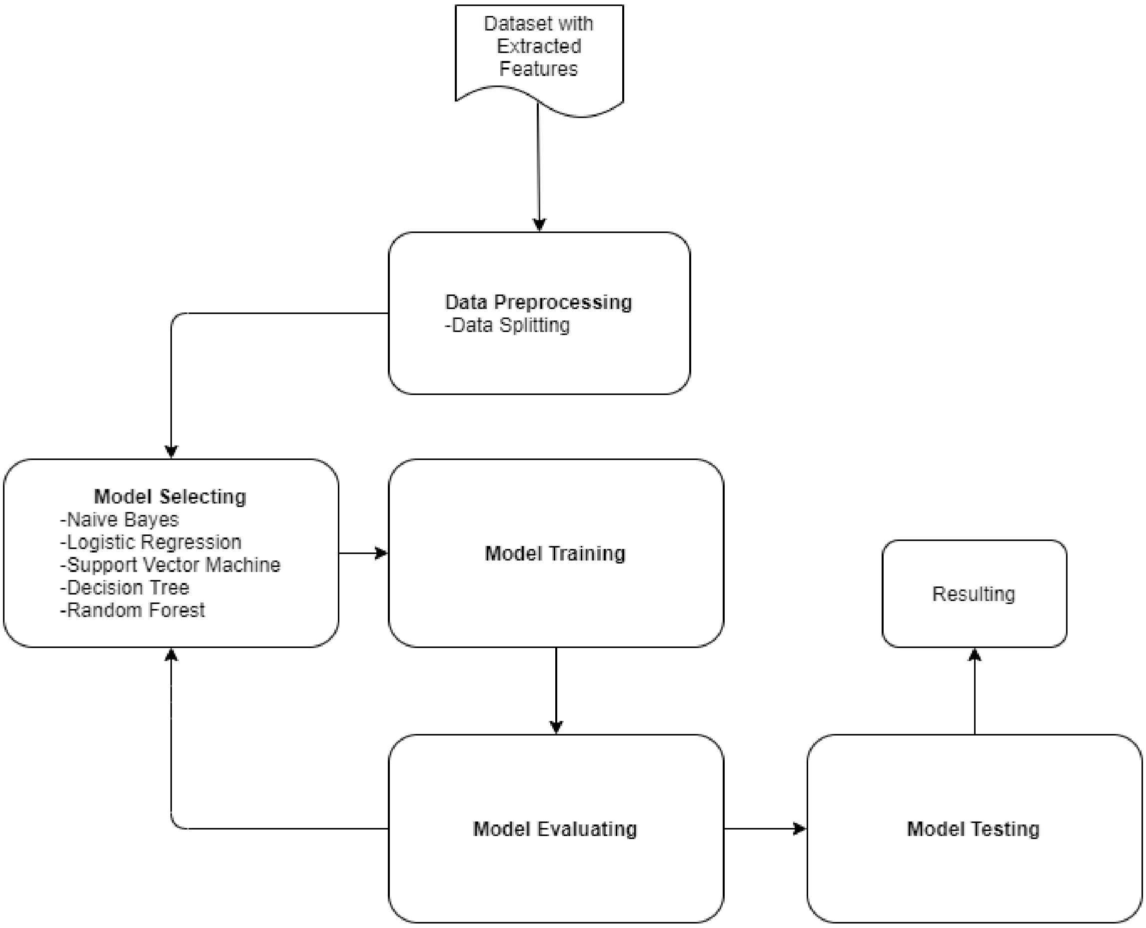


Figure 32. Data Analysis Process Diagram

Datas are taken after preprocessing process of Biomedical Engineering. Then the data is taken to a preliminary process and separation methods are applied here. Then, it is decided which machine learning method is to apply. After this process, the model is trained with the training data, tested with the test data and the results of the model are evaluated. If a high rate of success is achieved in the results, it means that the whole process was probably successful, if the success rate is not sufficient, the beginning is returned to make a change on the classifier.

3.2.4. Software Architecture

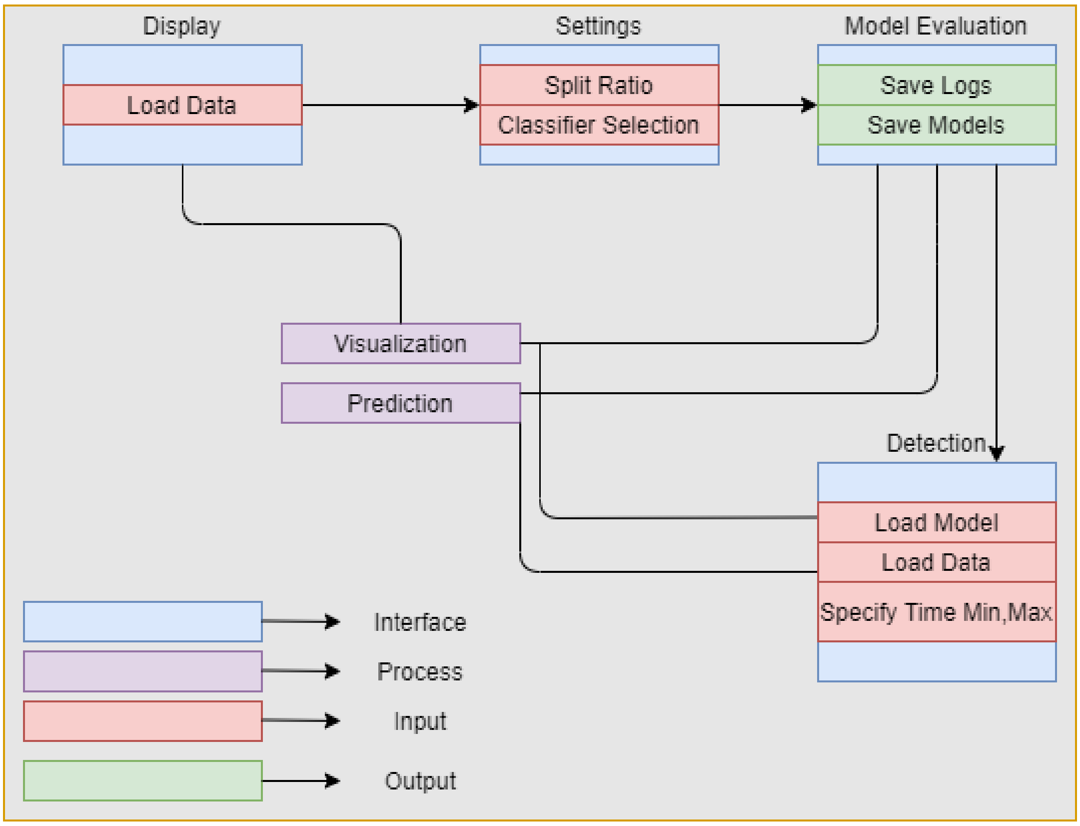


Figure 33. System Interface Graph

According to the graph, system has four interface layouts. Input, output and their processes are specified at each interfaceses. A user imports data on the Display interface, then specifies test ratio of the ML and selects a classifier algorithm. Then on the Model Evaluation interface, user trains the model and the result files are automatically saved by the system in the Results folder for future uses. Also user can visualize the test results. After that, on the Detection interface, user loads dataset and saved model. After the prediction process user visualize the graph of the results.

3.2.5. Implementation And Evaluation

3.2.5.1. Experiments

- We added a simple Machine Learning problem and checked the operability in order to understand that ML Works correctly in our application on the interface.
- Since our system will work using data provided by Biomedical Engineering students, we had to use fake data(EDF files) during the construction of the code part of the program.
- In the fake data to be used, we had to add the information that the person was focused and relaxad. For this, we had to use fake data again.

3.2.5.2. Changing Plan

There were some changes in the plans while building our system.

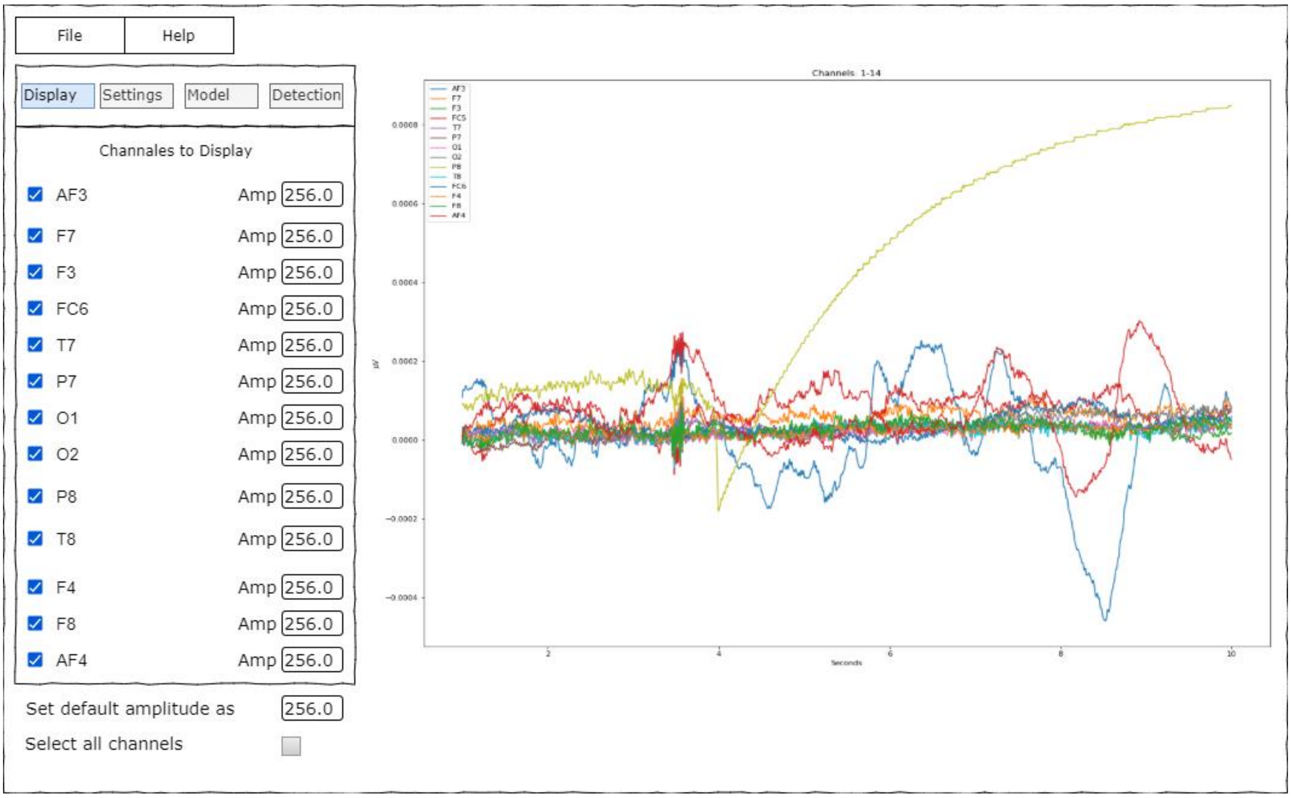


Figure 34. Previous Version of Software

One of the goals of our project was also preprocessing the brain signals on our application, first. But we gave up to do that because, there are many complicated aplications to preprocess the data, therefore we deleted this parts from our application.

3.2.5.3. Results of Analysis

Our system is running successfully. Test results are as in the picture:

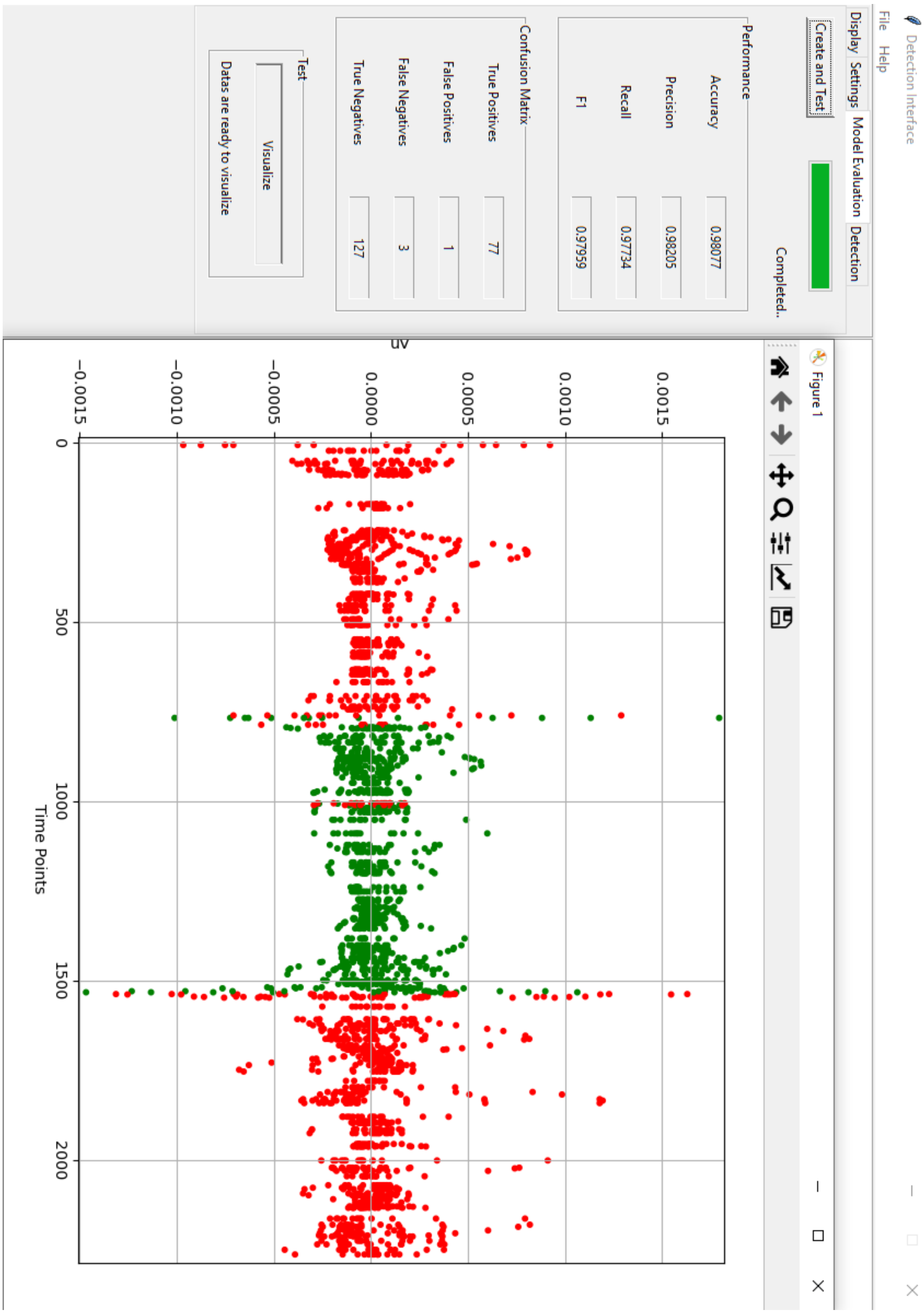


Figure 35. Test Result of Analyze

According to the picture, a model is builded and the results are on the left side of the application. User can compare the results of each classifier from this part. Also, if user clicks the Visualize button, graph of the test data can be seen. To speak about the graph, red dots are relaxation events and green dots are the concentration events of the data. Even we can see the wrong colors from the graph.

Prediction results are as in the picture:

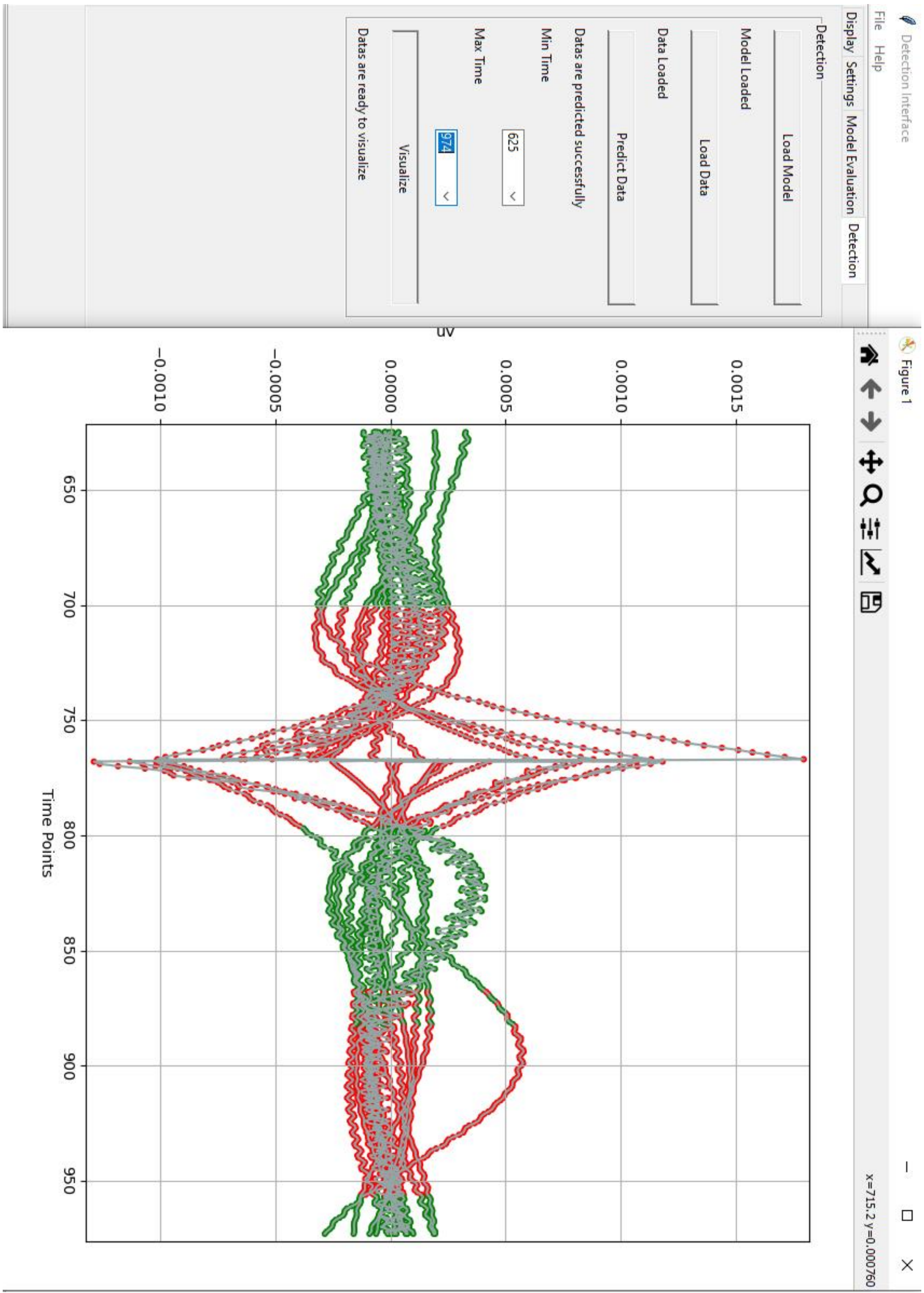


Figure 36. Detection Results

According to the picture, it can be seen that the model and the data have been added to the system. Then the model was run and the minimum and maximum values of the desired result were determined. The resulting graphic shows that the process was carried out successfully. But it can be seen that, prediction accuracy is not okey, and thats because, ML algorithm could not link the input datas with each other and it needs more data.

4. INTEGRATION AND EVALUATION

4.1. Integration

4.1.1. Biomedical Engineering

1. In our previous work, we preprocessed 180-second recordings. Our advisor stated that we need to divide 180 seconds into three, that we can carry out a more effective work in this way. In the new arrangement, we preprocess the EEG data for 60 seconds.

2. The most important change we made afterward was about filtering the signals. In the previous work, we applied the low-pass band as 0.5 Hz and the high-pass band as 50 Hz. We changed the low-pass band to 40 Hz and the high-pass band to 0.5 Hz, based on the literature we reviewed and the guidance of our advisor.

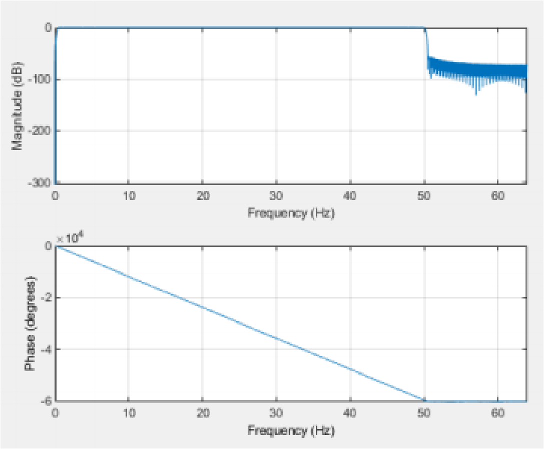


Figure 37. Previous Version of Filtered Signal

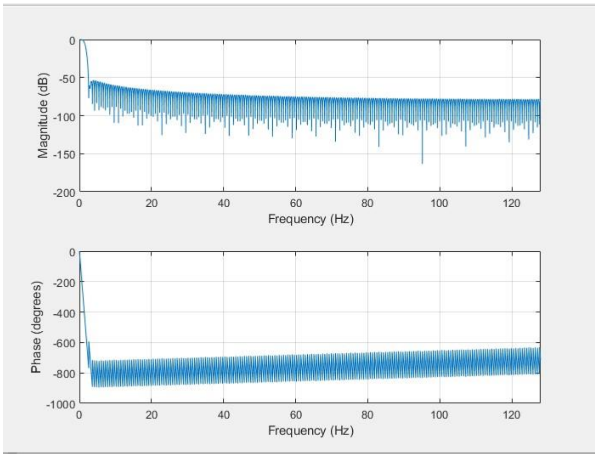


Figure 38. Last Version of Filtered Signal

3. No changes were made to the ICA and Examination by eye part because it was done correctly.

4. Unlike previous studies, we applied an extracted epoch. We got the epoch interval as -1

and 2 seconds. Immediately after, we automatically moved to the remove baseline step and subtracted the -1000 ms and 0 s interval.

5. After the epoch was applied, the ERP topographical image became visible in the EEGLAB toolbox.

6. To extract feature extractions properly and channel to channel, EEG data I preprocessed via EEGLAB (each subject has 180 seconds of recording. By dividing it into three, we preprocessed each 60 seconds and saved it as a .set file.) create I added it to the simple ERP study tab. There I applied FFT to the data and printed the statistical values of each channel on the Y axis in both moods one by one into the MATLAB workspace.

7. In the same study, we graphically printed the time domain frequency and electrical activities of the channels. Finally, we classify artifacts such as muscle movements, eye movements, and blinking in EEG data in the IClabel toolbox in EEGLAB. We used ICLabel's bottom tabs to clear the data from these artifacts once again.

8. We sent the clean data, which we preprocessed and extracted the feature extractions, in the form of files with .fig extension to our friends in the software engineering department for machine learning.

4.1.2. Software Engineering

1. For any Machine Learning problem, a data is absolutely needed. We did not have any data in the first weeks of the Project. For this reason, we worked on fake data to build a working system, so some parts of the project were built on these fake data. In the final stages of the project, when the real data was reached, the parts in the project were changed to the last version of the system.
2. In previous designs, the user had more control over the data while the model was in training process.

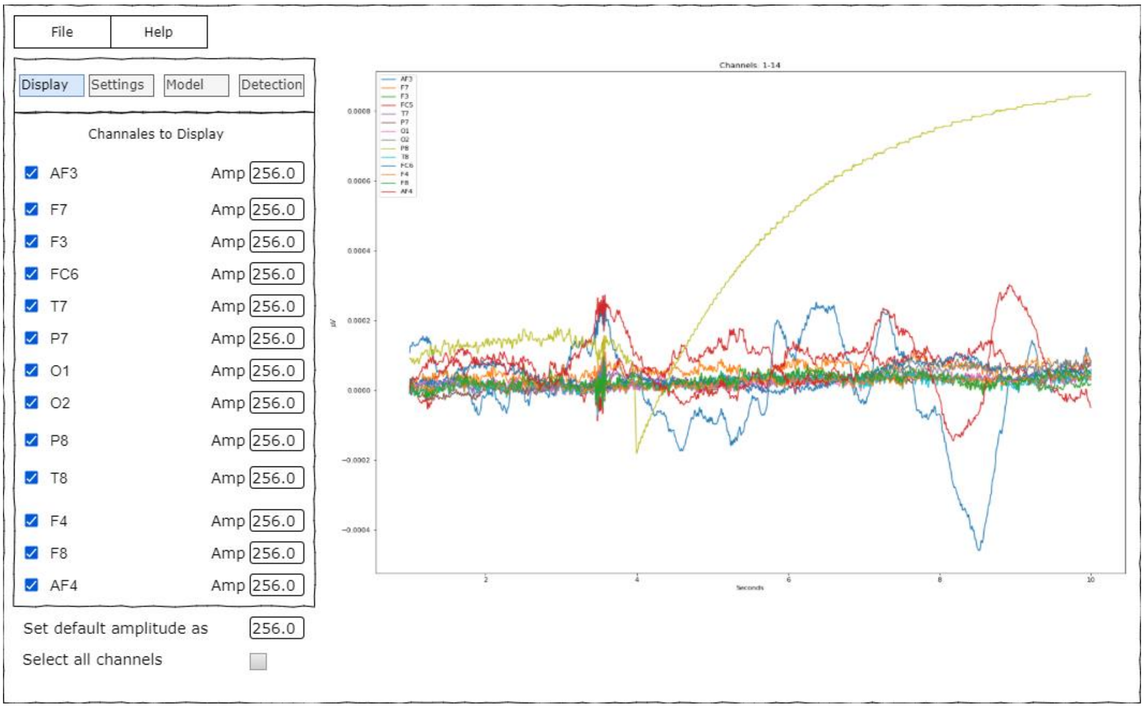


Figure 39. Previous Version of Display Frame

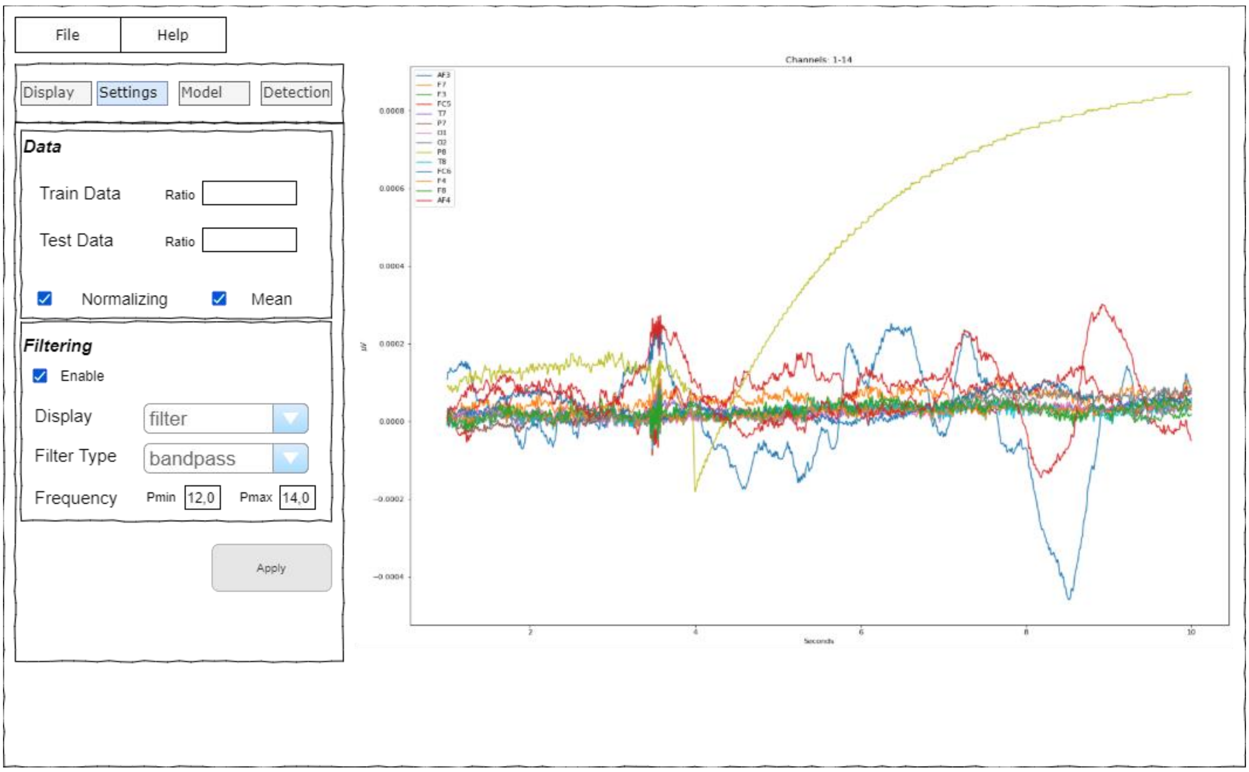


Figure 40. Previous Version of Settings Frame

These designs were only at the stage of thought. However, when we came to the programming stage, we realized that Python alone would not be sufficient and we needed to have the experience of other teams apart from software knowledge. Therefore, we have eliminated data filtering or similar works as in the figures and focused only on separating the data as test and train.

3. In the proposal, the system would track each time the user made changed in the frames. However, in order to provide convenience for the user, the user can make the desired settings until the user reaches the training stage and all changes are combined while creating the model. For this reason, the button in the settings frame has been removed and the codes of the Model frame have been adjusted accordingly.

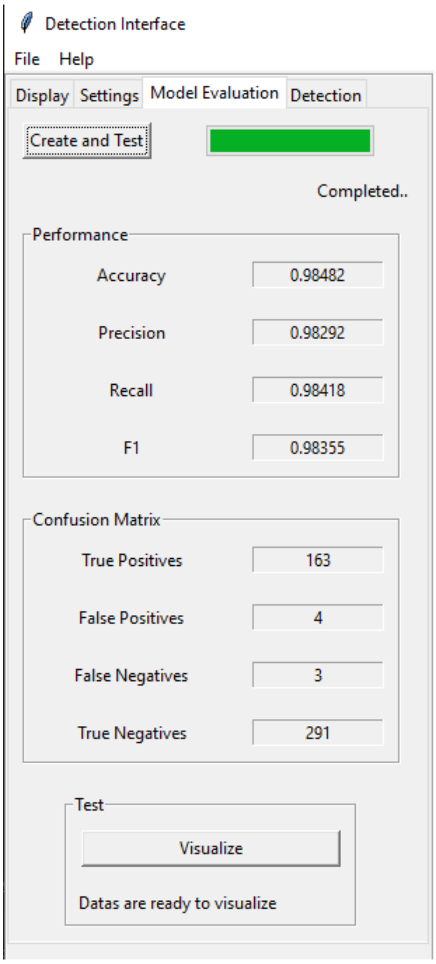


Figure 41. Current Version of Model Evaluation Frame

4. In the designs considered in the Proposal, the model creation and test results were in different frames. These sections have been combined in the Second Term so that model building and testing can be in one step. Thus, the user can now create and test a model with the click of a single button. In addition, the created model is automatically saved by the system into the Results folder which is in the Project's resource files. Also user can visualize the test results with the Visualize button as in the Figure 23.
5. There was no Detection Frame in the Proposal. This was a shortcoming and a new interface was created in the Report phase.

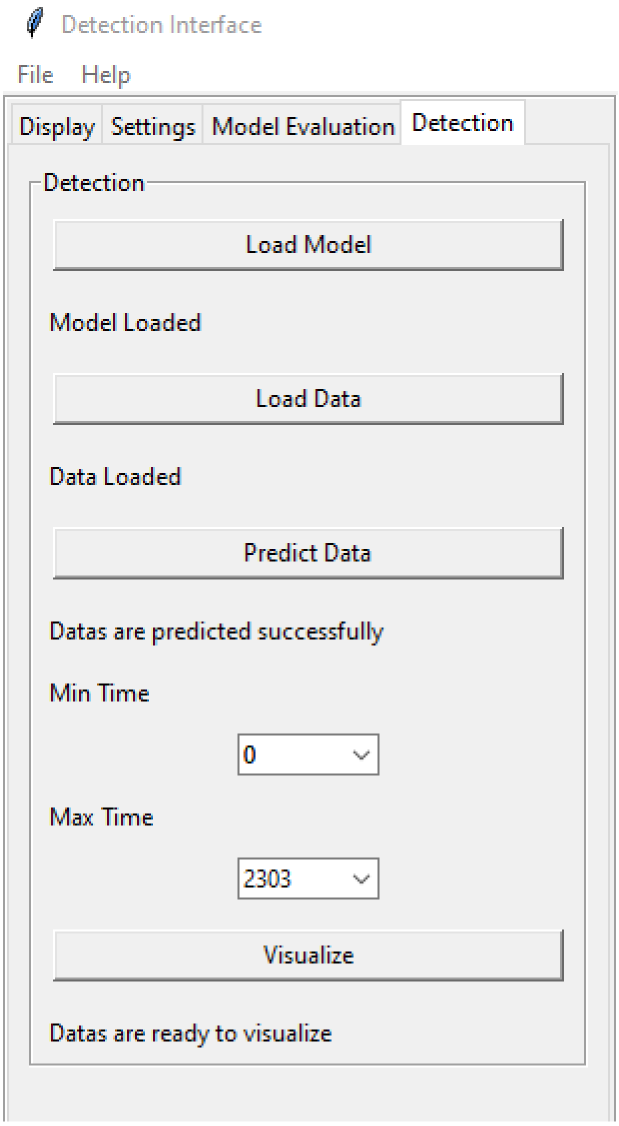


Figure 42. Current Version of Detection Frame

In this frame, the person must add the saved model and the desired data to the system. Then, by pressing the Predict button, it allows the model to predict. Then, when this process is finished, the person specifies the minimum and maximum time intervals they want to view and can see the results by pressing the Visualize button.

4.2. Evaluation

In order for the EEG data to be understandable, the data is preprocessed. The operations we apply to the data in the MATLAB/EEGLAB toolbox we use for the pre-process process are respectively: Channel locations, importing event fields, filtering, re-reference, ICA decomposition, examination by eye, extracted epoch, removing baseline.

The important changes in our plan are as follows; We changed the values of the high-pass and low-pass filters that we applied to the data, and enabled the clearer cleaning of artifacts such as background sound. The signals were not very understandable as the filtering values were wrong in our previous plan. Besides, we didn't know in our previous plan that dividing the signals into epochs would make such a significant difference. As we integrated our plan into real life, we decided that we needed to epoch the signals. After all these processes were completed in order, we had 60 seconds-60 seconds-60 seconds of preprocessed EEG data of a subject.

The reason we did feature extraction, which was not in our previous plan, was to classify the moods we wanted to detect. We tried to do feature extractions using only the EEGLAB extension, but we were unsuccessful. There was another problem, because the processes take a lot of time and there is a lot of data to be processed, so we cannot progress quickly. So at this point, we decided to do it in the STUDY tab in EEGLAB; We added as much preprocessed data as we wanted to the condition we wanted and analyzed all kinds of statistical data and graphics. This has made our work much easier.

On the software side, the system has two tasks. One of them is creating a model;

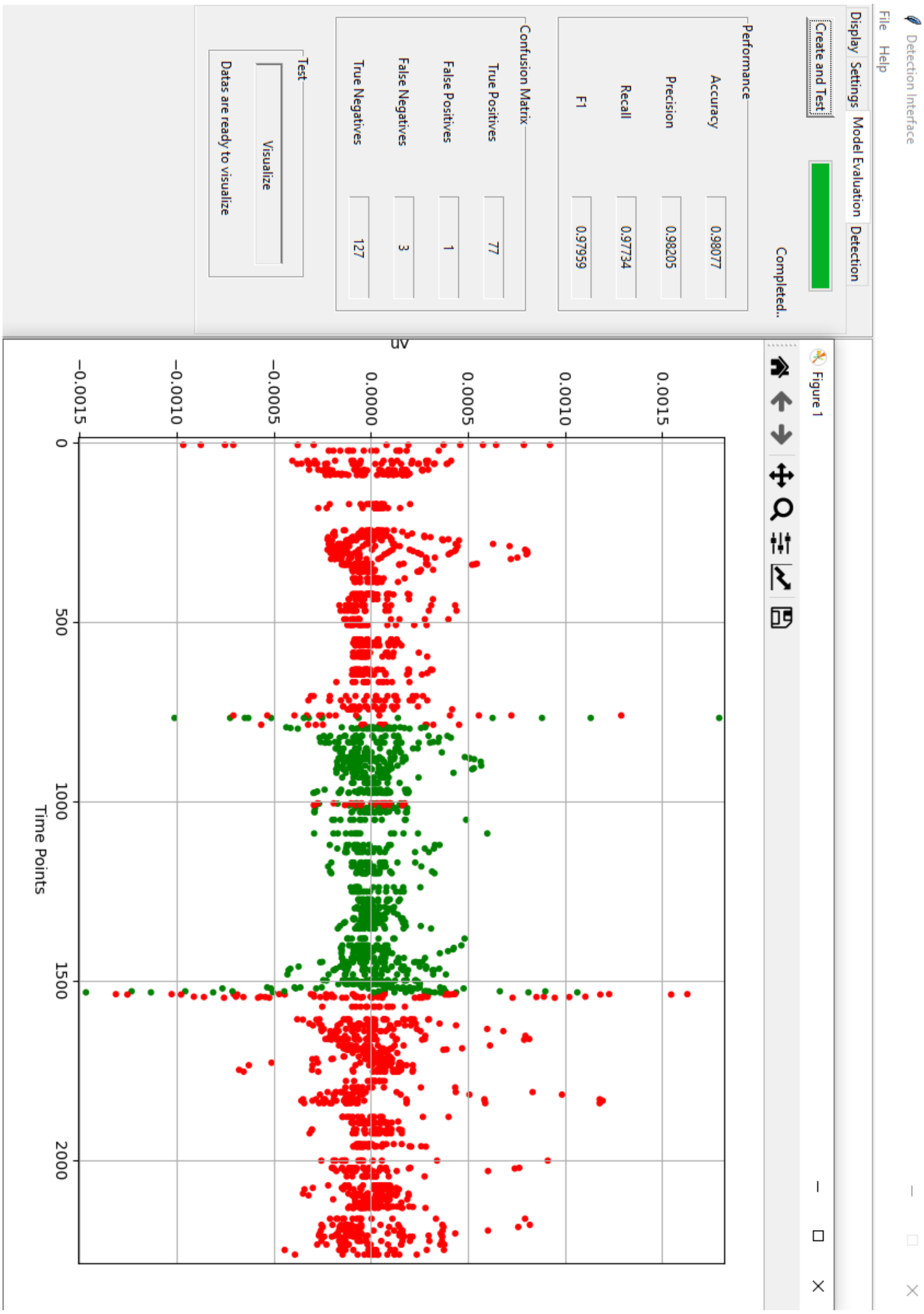


Figure 43. Test Results

The system successfully creates a model and displays the results of the model on the screen. It is also able to visualize test results, and when we look at the results, we can see that the system makes highly accurate predictions.

The other one is making detection;

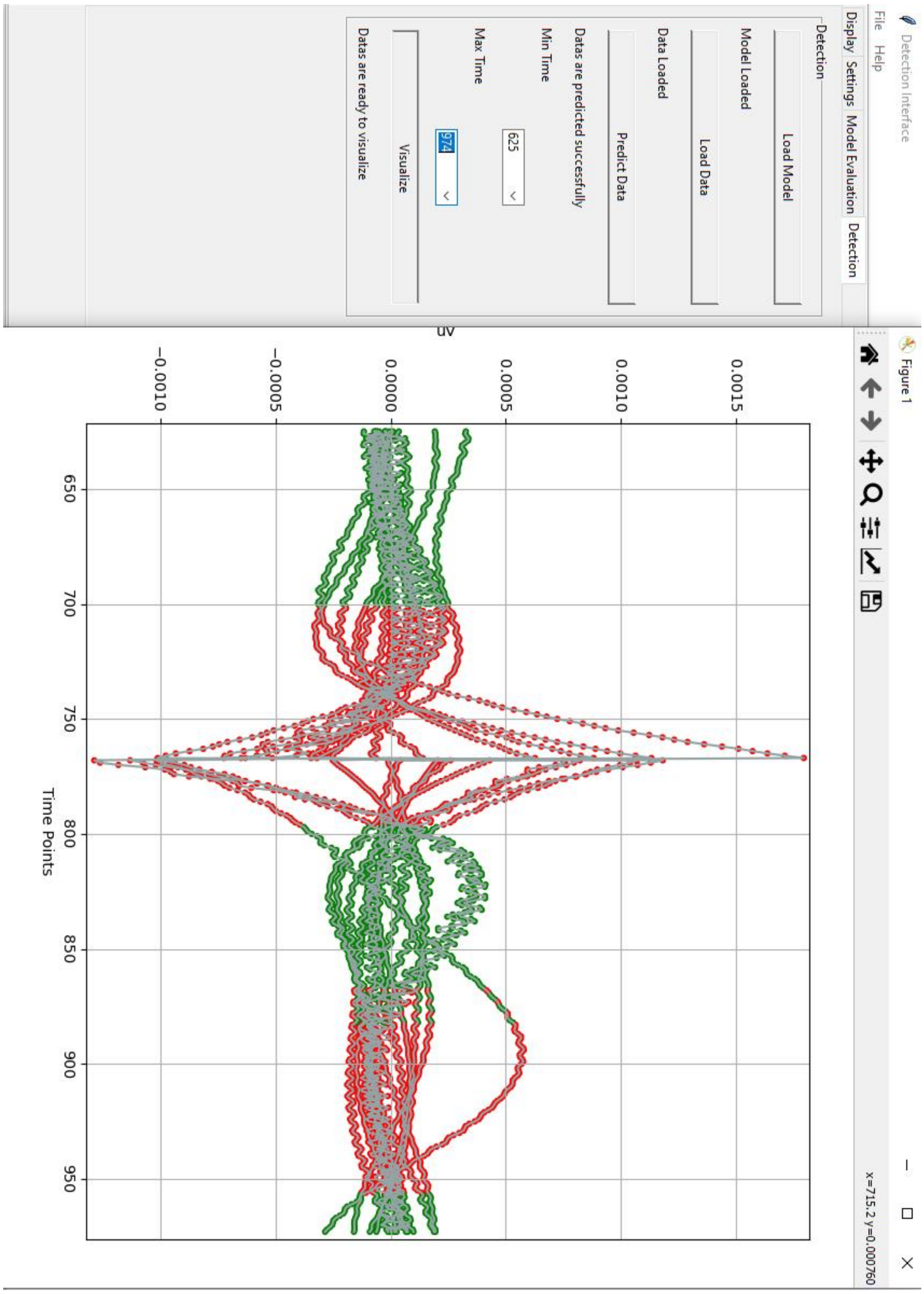


Figure 44. Prediction Results

There is a system error in this requirement. The model successfully terminate the prediction process, but the model makes the prediction wrong. The possible reason is that the model can not establish a link between the input data. To speak about this frame, model is trained with a person' s data but the prediction is made with another person's data. The possible solution would be to combine the data of many people in model training. In this way, the model will be able to establish a more consistent connection between the data.

5. SUMMARY AND CONCLUSION

In this project that we have been working on and completed, we have made the complex structure of EEG data analysable by going through many different processes. In order to detect relaxation and concentration moods in EEG data, we pre-processed the raw EEG data and removed artifacts such as blinking, muscle movements, and background noise. There were two brain waves we were looking for in this project. These were alpha and beta waves. We found the statistical properties of the processed data for feature extraction. We have classified the components of the data. We sent the data to software engineering students to do ML and the test results were successful.

We started by building the interface of the system, respectively. After the interfaces were prepared, we tried the system by adding a simple machine learning problem. When the test results were successful, we created a new ML problem by obtaining fake data related to the current problem and organized the system. Adhering to the designed plan, additions and arrangements were made in the project. When we got the real data, we changed the system again. Some bugs have been fixed. The final test stages were passed nearly successfully.

Some errors were encountered in the detection process. When the Model made prediction, there were too many errors in the results. Since the real data of the Model was reached in recent weeks, the Model remained more compatible with the fake data. In other words, the Model can not establish a connection between the data. However, if the data given to the Model is increased during the Model training phase, this problem will be solved. Still, the final version of the Project is working and errors can be fixed by making a few changes.

ACKNOWLEDGEMENTS

We wish to thank our advisers Assist. Prof. Burcu Tunç Çamlıbel and Assist. Prof. Betül Erdoğdu Şakar for their able guidance and support in completing our project.

REFERENCES

1. D. Vikrant, P. Matin, "A comparative analysis of machine learning methods for emotion recognition using EEG and peripheral physiological signals", *J Big Data* 7, 18 (2020). https://doi.org/10.1186/s40537-020-00289-7
2. Albasri, Ahmed (2019), "EEG dataset of Fusion relaxation and concentration moods", Mendeley Data, V1, doi: 10.17632/8c26dn6c7w.1. http://dx.doi.org/10.17632/8c26dn6c7w.1
3. MNEtools, URL: https://mne.tools/dev/auto_tutorials/intro/plot_10_overview.html
4. HackathonTutorial, URL: https://bigdatawg.nist.gov/HackathonTutorial.html
5. Youtube, URL: https://www.youtube.com/watch?v=sttf-Rgfl1Q
6. Z. H. Kilimci, A. Güven, M. Uysal, S. Akyokus, "Mood Detection from Physical and Neurophysical Data Using Deep Learning Models", Complexity, vol. 2019, Article ID 6434578, 15 pages, 2019. https://doi.org/10.1155/2019/6434578
7. Scott Makeig SCCN Director, Preprocessing Tools, 17 January 2020
8. Chuang C.-H., Ko L.-W., Lin Y.-P., Jung T.-P., Lin C.-T. (2014b). Independent component ensemble of EEG for brain-computer interface. IEEE Trans Neural Syst. Rehabil. Eng. pp. 22, 230–238. 10.1109/TNSRE.2013.2293139
9. Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. NeuroImage, 86, 446–460. Document id:10.1016/j.neuroimage.2013.10.027
10. International Journal of Computer Theory and Engineering, Vol. 7, No. 2, April 2015
11. Benjamin U. Cowley1,2, Jussi Korpela1, Jari Torniainen3, Computational testing for automated preprocessing: a Matlab toolbox to enable large scale electroencephalography data processing, March 6, 2017
12. Anderer, P., Roberts, S., Schlögl, A., Gruber, G., Klösch, G., Herrmann, W., … Saletu, B. (1999). Artifact Processing in Computerized Analysis of Sleep EEG – A Review. Neuropsychobiology, 40(3), 150–157. doi:10.1159/000026613
13. R. Ramirez, Z. Vamvakousis, "Detecting emotion of EEG signals using the emotive epoc device", Proceedings of the 2012 international conference on Brain Informatics, (2012).
14. M.L.R. Menezes, A. Samara, L.Galway, A. Santanna, "Towards emotion recognition for virtual environments an evaluation of EEG features on brenchmark dataset", Personal and Ubiguitous Computing 21, 6, (2017)
15. Kostas Michalopoulos, Nikolaos Bourbakis, "Application of multiscale entropy on EEG signals for emotion detection", Biomedical & Health Informatics (BHI) 2017 IEEE EMBS International Conference on, pp. 341-344, 2017.

16. Jingxin Liu, Hongying Meng, Asoke Nandi, Maozhen Li, "Emotion detection from EEG recordings", Natural Computation Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) 2016 12th International Conference on, pp. 1722-1727, 2016.
17. Özerdem, M.S., Polat, H. Emotion recognition based on EEG features in movie clips with channel selection. Brain Inf. 4, 241–252 (2017). https://doi.org/10.1007/s40708-017-0069-3