# EEG MECHANISM INTERACTION TO EVALUATE VEHICLE'S DRIVER MICROSLEEP

MUHAMMAD HAZIM BIN MOHD YUSOF

B.ENG (HONS.) ELECTRICAL ENGINEERING (ELECTRONICS)

UNIVERSITI MALAYSIA PAHANG

**UNIVERSITI MALAYSIA PAHANG**

**DECLARATION OF THESIS AND COPYRIGHT**

Author's Full Name : MUHAMMAD HAZIM BIN MOHD YUSOF

Date of Birth : 31/08/1998

Title : EEG MECHANISM INTERACTION TO EVALUATE DRIVER'S MICROSLEEP

Academic Session : SEMESTER I 2021/2022

I declare that this thesis is classified as:

| | | |
|---|---|---|
| ☐ | CONFIDENTIAL | (Contains confidential information under the Official Secret Act 1997)* |
| ☐ | RESTRICTED | (Contains restricted information as specified by the organization where research was done)* |
| ☑ | OPEN ACCESS | I agree that my thesis to be published as online open access (Full Text) |

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

980831-10-6559
New IC/Passport Number
Date:20/6/2022

(Supervisor's Signature)

DR. NORIZAM BIN SULAIMAN
Name of Supervisor
Date: 20/06/2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

## THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
*Perpustakaan Universiti Malaysia Pahang*,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

________________________________
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

## SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Electrical Engineering (Electronics).

_______________________________

(Supervisor's Signature)

Full Name : Ir. Ts. Dr. NORIZAM BIN SULAIMAN

Position : SENIOR LECTURER

Date : 20 JUNE 2022

_______________________________

(Co-supervisor's Signature)

Full Name :

Position :

Date :

**STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_______________________________

(Student's Signature)

Full Name : MUHAMMAD HAZIM BIN MOHD YUSOF

ID Number : EA18158

Date : 20/6/2022

# EEG MECHANISM INTERACTION TO EVALUATE VEHICLE'S DRIVER MICROSLEEP

MUHAMMAD HAZIM BIN MOHD YUSOF

Thesis submitted in fulfillment of the requirements
for the award of the
B.Eng (Hons.) Electrical Engineering (Electronics)

College of Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2022

# ACKNOWLEDGEMENTS

Praise to Allah (SWT) for His blessings and guidance in giving me strength, courgae, patience and perseverance to endure this final year project.

First and foremost, I'd want to express my gratitude to Ir. Ts.Dr. Norizam Bin Sulaiman, my final year project supervisor, for his important advice, knowledge sharing, continual mentoring, and support in making my project a success. I'd also like to thank him for his patience and courtesy in sharing useful information and observations with me as my project progressed. I appreciate his dedication and effort spent reviewing and exposing some errors in order to improve my work.

I am grateful to all lecturers at Universiti Malaysia Pahang (UMP), particularly those in the College of Engineering Department, for providing me with the opportunity to participate in research and projects.

# ABSTRAK

*Microsleep* atau lebih dikenali sebagai terlelap seketika tanpa dapat dikawal dalam tempoh yang sangat singkat biasanya berlaku di antara satu saat hingga lima belas saat. Di Malaysia, salah satu faktor yang menyumbang kepada kemalangan adalah disebabkan oleh faktor *microsleep* ketika pemandu sedang memandu tanpa mereka sedar. Faktor ini juga sering berlaku apabila memandu dalam keadaan penat dan jarak perjalanan yang terlalu jauh. Faktor cuaca juga boleh menyumbang kepada *mikrosleep*. Oleh itu, dalam penyelidikan ini, telah dibangunkan sebuah sistem untuk mengesan gelombang *frekuensi* daripada otak berdasarkan isyarat daripada *electroencephalogram* (EEG) elektrod bagi mengelak pemandu daripada mengalami *mikrosleep* dan terlibat dalam kemalangan. Bagi menjalankan penyelidikan ini, lima buah sampel daripada perbezaan umur dan jantina telah dipilih untuk mengutip data gelombang otak mereka menggunakan peranti NeuroSky Mindwave Mobile Headset dan aplikasi EegID Record dalam dua situasi yang berbeza, iaitu dengan memandu simulasi dalam situasi yang mencabar selama 30 minit dan yang kedua adalah dengan memandu simulasi dalam situasi yang tenang selama 30 minit. Di samping itu, penggunaan MATLAB dalam penyelidikan ini adalah untuk melakukan pra proses isyarat gelombang bagi membuang gangguan bunyi yang tidak diingini. Kemudian, penapis bandpass digunakan untuk mengklasifikasikan dan memisahkan isyarat kepada gelombang *Theta, Alpha,* dan *Beta.* Ketiga-tiga gelombang ini akan dianalisis dan dikaji berdasarkan perbezaan umur serta jantina subjek. Selepas spektogram daripada gelombang tersebut dilukis untuk mencetuskan sistem penggera dan motor getaran stereng jika *mikrosleep* dikesan untuk beberapa tempoh satu sehingga 3 saat.

# ABSTRACT

Microsleep or more commonly known as momentary uncontrollable fall asleep in a very short period of time usually occurs between one second to fifteen seconds. In Malaysia, one of the factors that contribute to accidents is due to the microsleep factor when the driver is driving without them being aware. This factor also often occurs when driving in a tired state and traveling too long distances. Weather factors can also contribute to microsleep. Therefore, in this research, a system has been developed to detect frequency waves from the brain based on signals from electroencephalogram (EEG) electrodes to prevent drivers from experiencing microsleep and getting involved in accidents. To conduct this research, five subjects of different ages and gender were selected to collect their brainwave data using the NeuroSky Mindwave Mobile Headset device and the EegID Record application in two different situations, namely by driving the simulation in a challenging condition for 30 minutes and the second situation is by driving the simulation in a relaxed condition for 30 minutes. In addition, the use of MATLAB in this research is to pre-process the wave signal to remove unwanted noise interference. Then, a bandpass filter is used to classify and separate the signal into Theta, Alpha, and Beta waves. These three waves will be analyzed and studied based on the age and gender differences of the subjects. After the spectrum of the wave is drawn to trigger the alarm system and the steering vibration motor if microsleep is detected for some period of one to 3 seconds.

**TABLE OF CONTENT**

## LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| EEG | Electroencephalogram |
| FFT | Fast Fourier Transform |
| PSD | Power Spectral Density |
| MSE | Microsleep Episodes |
| GUI | Graphical User Interface |
| BCI | Brain Controlled Interface |
| SD | Standard Deviation |
| FP1 | Frontal Parietal lobe 1 |
| KNN | K-Nearest Neighbour |
| SVM | Support Vector Machine |

# CHAPTER 1

# INTRODUCTION

## 1.1 Project Background

Microsleep, also known as a lapse, is one of the leading causes of fatal road accidents in Malaysia, with drivers experiencing a loss of attentiveness and brief eye closure while behind the wheel. Microsleep is described as a brief period of drowsiness during which an individual loses awareness and is unable to respond to arbitrary sensory input.

Electroencephalography (EEG) is a test that measures and monitors the various electrical processes in the brain. Different types of electrical activity correspond to various brain states. Every physical action a person takes is the result of brain activity, which generates an electrical signal. These signals can be recorded and analyzed to obtain useful data. The NeuroSky Mindwave EEG headset is used in this study to capture human EEG data or brain waves in two separate conditions: active and relaxed. Delta, Theta, Alpha, Beta, and Gamma are the five frequency bands that make up an EEG signal. The signal is then analyzed with MATLAB software to find the unique features that can be used to evaluate human microsleep.

## 1.2 Problem Statement

According to studies and research on EEG-controlled devices, the majority of EEG headsets used by researchers are either wet electrodes caps or commercial EEG headsets from the brand Emotiv with several electrode channels. Due to its branding and various channels, all of this EEG equipment is relatively expensive and hard to manipulate. Although it is claimed that having more channels allows for more precise extraction of brain signals, in reality, not everyone can afford such pricey gadgets.

Besides, some of the alert systems that preventing from microsleep does not suitable in some areas. An alert system using IoT where it sends messages and email to family members of the driver to alert the driver could become a problem where the driver might be going to some rural area that does not have a network coverage area.

## 1.3 Project Objective

The goal of this study is to create a microsleep detection system for drivers. To achieve the main purpose, numerous sub-objectives of this project must be implemented, as listed below:

1. To construct a microsleep detection using EEG mechanism by analyze the data collected at active state and relaxed state.

2. To develop an effective alert system by using vibration and alarm based on driver awareness.

## 1.4 Project Scope

In order to complete this project, there are numerous project scopes that cover the majority of the project objectives. It contains the following items:

1. Filter the Raw EEG data into Theta, Alpha, and Beta band.

2. Calculate the feature extraction (mean, standard deviation, and variance).

3. Feature classification using KNN Classification and deep learning method.

4. Build the GUI for monitoring driver state.

5. Develop the alert system to prevent the driver had microsleep.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

The literature overview and study cases on EEG signals are covered in this chapter. This strategy is critical for ensuring that the project stays on schedule and reducing the number of errors. The method utilized in earlier investigations will be explained in the sub-chapters that follow. The literature review is a vital phase in completing a project since students will get knowledge from the research paper that was published by reading the proof and claiming to get more precise results by doing a literature study.

## 2.2 EEG Signal Processing

The electrical activity generated by the brain was captured using electroencephalography (EEG). Our brain has millions of neurons. Millions of tiny electric voltage fields are generated as a result of these actions. Electrodes mounted on the scalp can detect the average of these voltage fields. Four simple periodic rhythms recorded in the EEG are alpha, beta, detlta, and theta [5]. These rhythms are identified by frequency (Hz or cycles/sec) and amplitude (μV) [5]. The absolute band power of the EEG signal was computed by taking the Fast Fourier Transform (FFT) of the time series signal [1]. The delta band, which is related with sleep posture, and the theta band, which defines tiredness, are both present in this signal. The alpha band describes relaxation, while the beta band describes alertness. Drowsiness is expressed by a decrease in the alpha band and an increase in the theta frequency. Table 2.1 shows the EEG pulses illustrated on certain location of human brain and their reason.

Table 2. 1: EEG Pulses

| Rhythms | Frequency Interval | Location | Reason |
|---|---|---|---|
| Delta | (0-4) Hz | Frontal Lobe | Deep sleep |
| Theta | (4-7) Hz | Median, temporal | Drowsiness and meditation |
| Alpha | (8-13) Hz | Frontal, occipital | Relaxation and closed eyes |
| Beta | (13-30) Hz | Frontal, central | Concectration and reflection |
| Gamma | (>30) Hz | - | Cognitive functions |

Source: Mejdi Dkhil (2015)

In this study, power spectral density (PSD) feature extraction was applied to the EEG signal. PSD is defined as the measure of signal’s power content versus frequency. PSD is used to extract the theta, alpha, and beta band.

Various data mining techniques used in feature classification are Artificial Neural Network (ANN), Support Vector Machines (SVM), Adaptive neuro-fuzzy inference system (ANFIS), k-nearest neighbor (kNN), naive Bayes classifier (NBC), Linear/Quadratic discriminate analysis (LDA/QDA) and Decision tree (DT) [5].

## 2.3 Journal Paper

The project's development was achieved by reading and researching related journal publications in order to develop ideas and information for the project's completion. The journal paper has defined the key theme that is related to the development in the development. Various approaches from each journal paper aid in the collection of various thoughts and information that aid in the development. The way through which journalists receive the provide determines the outcome of the journal paper collecting.

Table 2. 2: Highlighted research papers

| No | Title | Author | Method | Result |
|---|---|---|---|---|
| 1 | Drowsy driver detection by EEG analysis using Fast Fourier Transform (2015) | Mejdi Ben Dkhil,<br>Ali Wali,<br>Adel M. Alimi | ▪ Using Emotiv EPOC 14 electrodes Headwear<br>▪ FFT used to find the absolute band.<br>▪ Feature Extraction using Arousal, Valence, Dominance | Increase in alpha and theta power is observed when subject is in transition from alert state to drowsy state. |

| | | | | |
|---|---|---|---|---|
| | | | ▪ Clustering by Fuzzy C means. | |
| 2 | Recording and Characterization of EEGs by Using Wearable EEG Device (2019) | Ryoe Inue | ▪ EEG Device with electrode placement L1, L2, L3, L4, C, R1, R2, R3, R4 (9 placements).<br>▪ Ten adults (males, 22-23 y/o).<br>▪ Given set of instructions to the subject while recording. | Able to differentiate different waveform output from different instructions. |
| 3 | Automatic detection of microsleep episodes with feature based machine learning (2020) | Shorucak | ▪ EEG Recording and Pre-precessing.<br>▪ Quantitive Analysis method used.<br>▪ Power Spectral Analysis.<br>▪ Feature Engineering | Alpha activity was present before the microsleep but not during microsleep.<br><br>Drop on beta activity during MSE.<br><br>Theta activity is high during MSE. |
| 4 | Spectral Analysis of EEG During Microsleep Events Annotated via Driver Monitoring System to Characterize Drowsiness (2018) | Chunwu Wan | ▪ Spectral Analysis of EEG During<br>▪ Microsleep Events Annotated<br>▪ via Driver Monitoring System<br>▪ to Characterize Drowsiness | Results showed significant decrease in relative delta power, while the alpha power was significantly increased in all regions during MS |
| 5 | A Data Mining Approach for Sleep Wave and Sleep Stage Classification (2016) | Aleena Swetapadmaa<br><br>Brijesh Raj Swainb | Among all recordings, 5000 segments were selected as database. 70% is used for training and 30% is used for testing. | |
| 6 | Classification Consistency of Concentrated and Relaxed Mental States with EEG Signals (2019) | Shingchern D. You,<br><br>Yu-Cheng Wu | A subject is tested 40 times to collect 20 EEGs in concentrated state and the other in relaxed state | Performing normalization does not effectively improve the accuracy.<br><br>By using SVM as the classifier, the accuracy is over 78% |

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

This chapter will introduce the methodology to develop the mechanism interaction to analyze and extract the brain signals and to investigate the parameter and brain wave pattern of microsleep events for this project.

## 3.2 Overview of Research

It all began with determining the project's goal, problem statements, and scope of work. It is critical to guarantee that the study direction is correct and that any connected works or advances are on track. Electroencephalography (EEG) and microsleep event (MSE) understanding are required to complete this project. Thus, a literature review of EEG and MSE information and expertise is necessary to gain a better understanding of the project. Literature evaluations can help to uncover additional benefits or drawbacks in various handling ways for the project's software and hardware. The optimal method is decided based on the reviews so that no time is wasted in completing the job.

The next thing to be carried out is the EEG Device Recording Setup (NeuroSky Mindwave Mobile) and making sure the device is working properly. The next step is the Experimental Measuring Protocol where this step has defined the procedure to collect the EEG Dataset from selected subjects. After that, the process of signal processing of the EEG signal is taken by using MATLAB software to evaluate the brain wave signal followed by the development of hardware. To give a clear view, the Graphical User Interface (GUI) was constructed to display the band spectrum of Theta, Alpha, and Beta and also the microsleep occurrence. Then, the MATLAB programming will be compiled and system testing will be conducted. All these processes are summarized to be shown in Figure 3. 1 and Figure 3. 2 which will be explained in detail in this chapter.

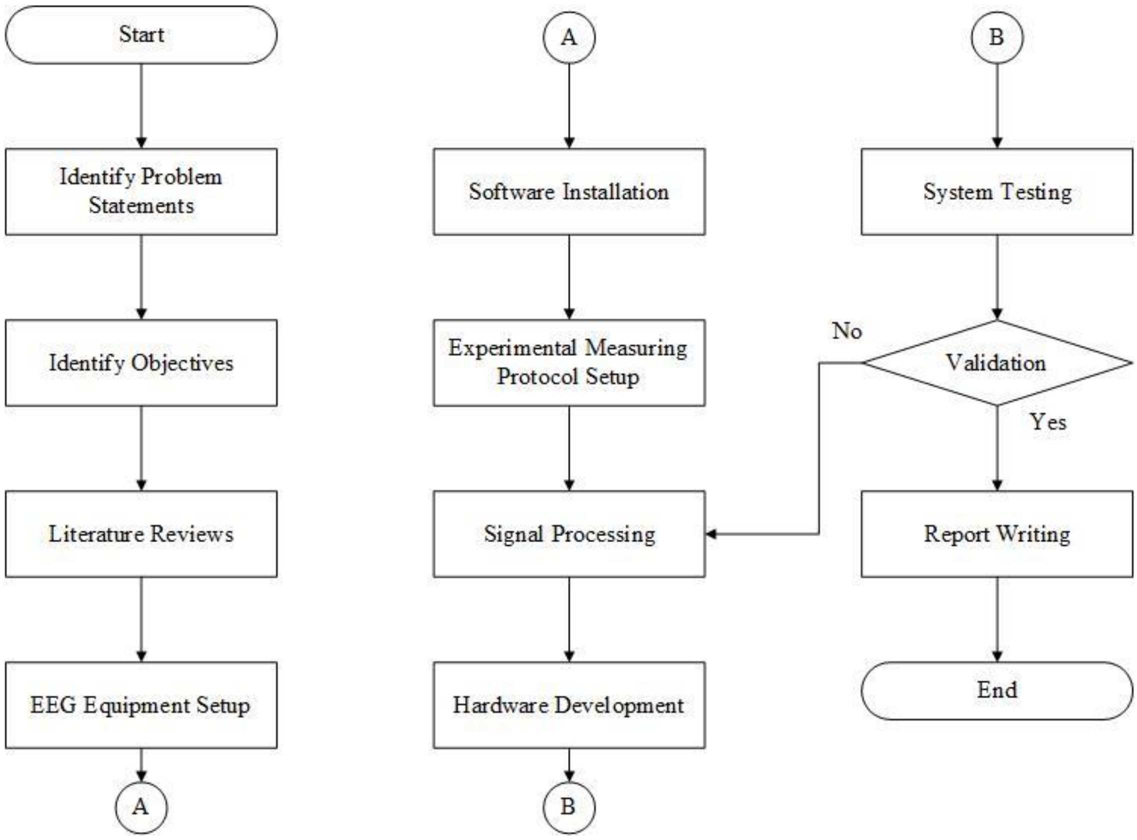



Figure 3. 1 Project Flowchart of EEG Mechanism Interaction to Evaluate Vehicle's Driver Microsleep.

The developed system is built to receive brain waves as input. The EEG signal will then be processed by the EEG Signal Acquisition Unit, which is the NeuroSky Mindwave Mobile Headset with a Bluetooth module. When the NeuroSky headset is connected, the mobile application EegID is launched to record and save the data into GoogleDrive as an Excel file (.csv format). After the EEG data is collected, it is sent to MATLAB for signal filtering, which removes the unwanted noise.

## 3.3 Methodology Flowchart and Block Diagram

Figure 3. 2 shows the methodology flowchart and Figure 3.3 shows the methodology block diagram of this project. It is started with subject selection and

there are 5 subjects chosen which consists of two different age categories which are mean of age 25 and 50 respectively. Human age is one of the most significant factors to consider when detecting microsleep because it has an impact on the results and outcome.

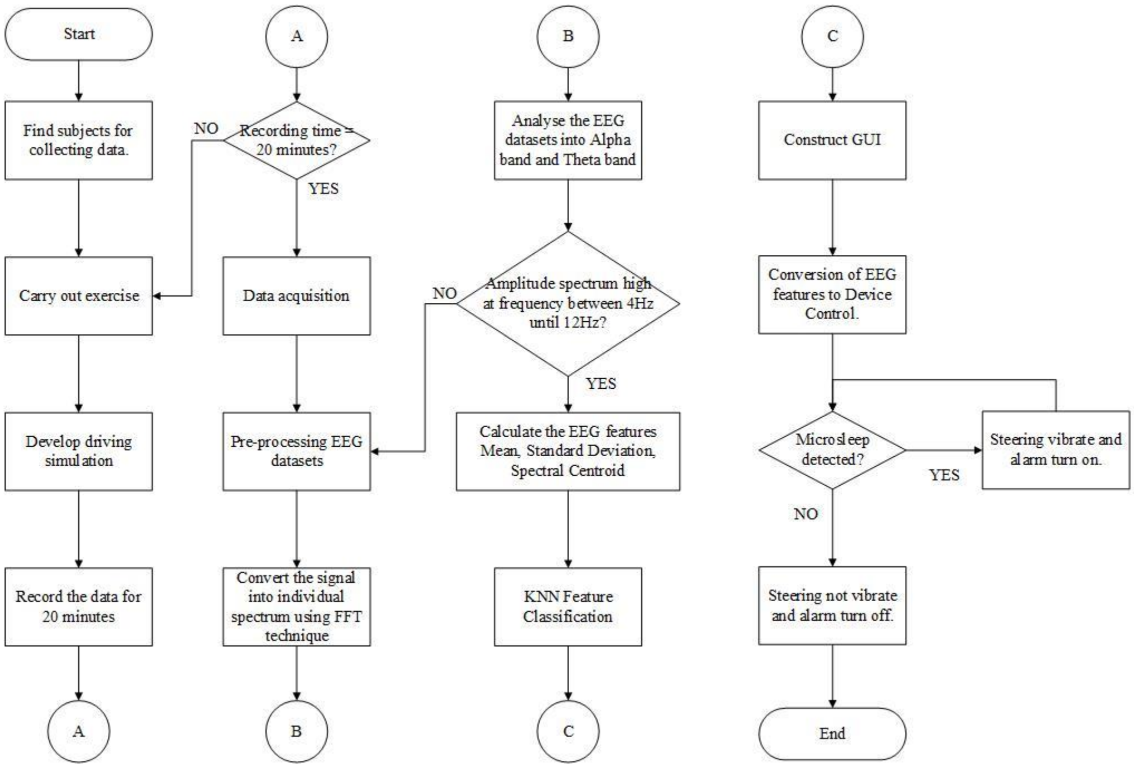



Figure 3. 2 Methodology Flowchart of the project which is explained in detail.

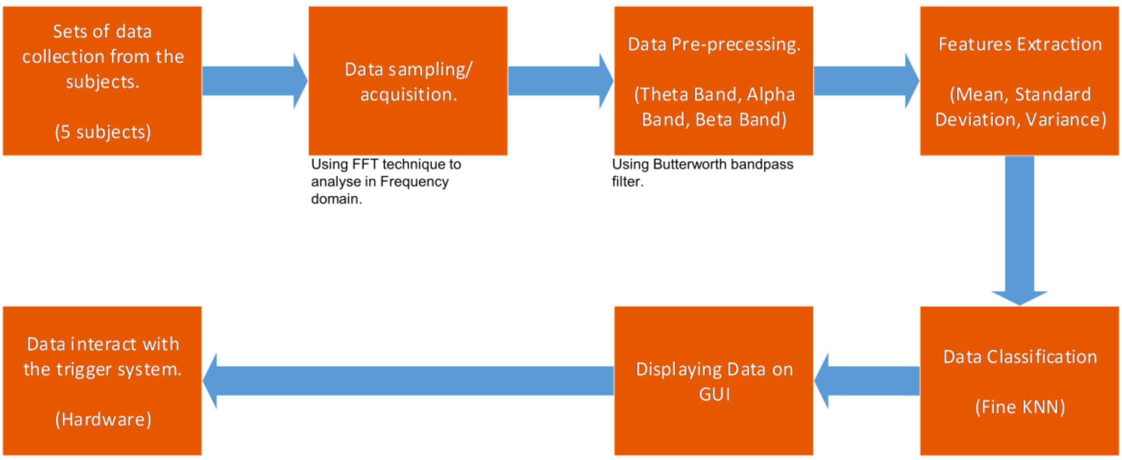



Figure 3. 3 Methodology Block Diagram of the project.

### 3.3.1 Experimental Measurement Protocol for EEG Data Collection

To collect the data more efficiently, the user must follow the procedures that were made when performing the recording of obtaining the EEG data from the subjects. The protocol in Table 3. 1 that shows below is telling on how to set up the measuring equipment where this is important to avoid measurement errors while conducting the recording.

Table 3. 1 Instruction to set up the EEG device with mobile phone and attach to user.

| Step No. | Instruction |
|---|---|
| 1 | Select at least 5 subjects from different mean ages (20-30 and 40-60) |
| 2 | Turn on the NeuroSky Mindwave Mobile headset (EEG device) and make sure the device is in Bluetooth pairing mode. |
| 3 | Download the eegID application from the Google Playstore and turn on the Bluetooth of the mobile and make sure it detects the Bluetooth address of the NeuroSky Mindwave Mobile. |
| 4 | Open the eegID application from the mobile and connect the EEG device. Make sure the status of the connection in eegID is connected. |
| 5 | Clean the surface of the subject’s forehead by using an alcohol swab sheet to make sure the surface is clean and dry. |
| 6 | Place the EEG device's overhead band on the user's head and adjust it to fit his or her head size. |
| 7 | Adjust the EEG device so that the sensor arm is firmly in contact with the user's forehead. |
| 8 | Ensure the rubber ear hoop stays in place behind the left ear, then clamp the headset's ear-clip to the earlobe. |
| 9 | Adjust the forehead sensor until the reading of poor signal in eegID application is 0. |

The protocol continues with steps while recording the EEG data from the subjects. Table 3. 2 shows the procedure while conducting the recording session. All these steps are required to be followed to make sure the brain wave pattern is corresponding to the experiment. They both experiment conditions are taken in driving simulation.

Table 3. 2 The procedure of recording while conducting the experiment.

| Step No. | Instruction |
|---|---|
| 1 | Set the real driving simulator device and connect to the PC. |
| 2 | Run the program Need for Speed (Carbon) on PC. |
| 3 | Setting up the controller using the joystick (driving simulator). |
| 4 | Select to Racing Mode (Active Condition) in-game and choose difficulty to hard and set the lap to 2 laps. Give the subject to try and make sure that he is comfortable with the gameplay mode. |
| 5 | Repeat step 4 for the selecting of racing mode. Set the difficulty to hard and the lap to 15 laps. |
| 6 | Set the time limit for recording in the eegID application to 20 minutes then start recording. |
| 7 | After finishing the recording, send the CSV file from the eegID application to Google Drive. |
| 8 | Select to World Run Mode (Relax Condition) in the game. Give the subject to try to drive the gameplay for 5 minutes and make sure that he is comfortable with the gameplay mode. |
| 9 | Set the time limit for recording in the eegID application to 20 minutes then start recording. |
| 10 | After finishing, the recording, send the CSV file from the eegID application to Google Drive. |

Figure 3. 4 below shows the racing mode, map circuit, and the setting for collecting data in the Active Condition state. While below shows the setting to induce the Relax Condition state while recording the EEG data from subjects.


Figure 3. 4 Active Condition mode setup for recording the EEG data.


Figure 3. 5 Relax Condition mode setup for recording the EEG data.

### 3.3.2 EEG Signal Pre-Processing

EEG signals are made up of five separate waves, each with its unique frequency band: Delta, Theta, Alpha, Beta, and Gamma waves. In this experiment, however, only the Theta, Alpha, and Beta waves are collected and used to investigate microsleep events. The raw data that was collected usually contain unwanted noises. To remove the noise, filtering is needed. Using MATLAB, the raw data collected is imported into the software and only read the RAW Value in the spreadsheet. Next, eliminate the signal outside the range of -300 to 300.

The current signal is in the time domain. To convert the signal into the frequency domain, Fast Fourier Transform (FFT) is being used in MATLAB. The sampling frequency is set to 512Hz and limits the frequency from 0.5Hz until 50Hz. The converting is to classify the frequency band which is Theta, Alpha, and Beta.

### 3.3.3 Feature Extraction

The process of defining a set of features is known as feature extraction. The importance of feature extraction in providing the most efficient analysis cannot be overstated. The EEG signal was analyzed to power spectral density (PSD) feature extraction in this investigation. PSD is defined as the ratio of power content to the frequency of a signal. The Theta, Alpha, and Beta bands are extracted using PSD. Theta, Alpha, and Gamma bands are then classified using a bandpass low filter. Theta band has a frequency range of 4-7Hz, the Alpha band is 8-12Hz, and the Beta band is 13-30Hz. The mean and standard deviation can be obtained from the PSD.

### 3.3.4 Feature Classification

In this research, the KNN classifier is preferred, and it is utilized to categorize the EEG signal's specified feature. The KNN classifier was chosen over other classifiers because it is a basic and intuitive method of classification that researchers utilize to classify signals. The KNN classifier compares new samples (also referred to as testing data) to baseline data (often referred to as training data) and returns a result. The KNN classifier tested with 4 different types of KNN which are Fine KNN, Medium KNN, Coarse KNN, and Weighted KNN.

## 3.4 Hardware Development

The development and optimization of electronics and mechanical system elements that conduct various computational operations are all part of hardware development. A hardware interface is frequently used to connect various elements and objects.

### 3.4.1 NeuroSky Mindwave Mobile Headset

The EEG data of the subjects are captured from the scalp outside the skull using the NeuroSky Mindwave Mobile Headset in this study, which is a non-invasive BCI. It has a single dry electrode that is put on the left forehead at the FP1 position as described by the worldwide 10-20 EEG electrode placement system in Figure 3.6, as well as an ear clip that serves as the device's grounding. It just has one channel electrode and one grounding, making it much easier to put on and adjust to the user's head size. Figure 3.7 displays the NeuroSky Mindwave Mobile Headset, which is used to collect EEG data and then to detect microsleep. The NeuroSky Mindwave Mobile Headset has a sampling frequency of 512Hz.

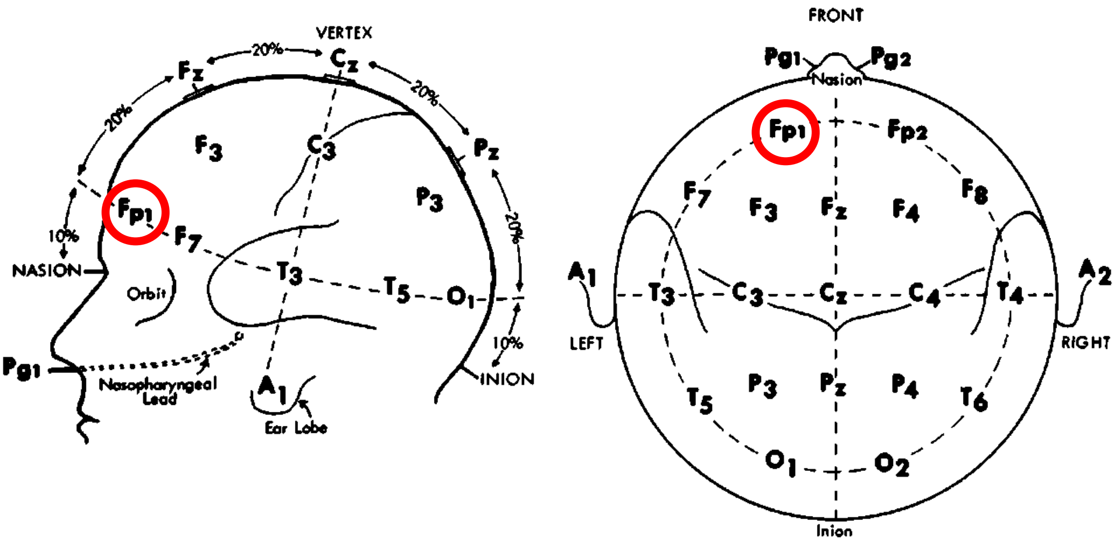



Figure 3. 6 The position of FP1 in 10-20 EEG electrode placement system.


Figure 3. 7 NeuroSky Mindwave Mobile Headset with labeling parts.

### 3.4.2 Driving Simulator Device

PXN is a global leader in the design and manufacture of professional gaming accessories such as steering wheels and game controllers arcade joysticks for flight handsets and gaming keyboards. PXN has acquired several software and hardware technologies, as well as more than 20 patents and software copyrights in the United States and abroad. These items are available for PC, game consoles, and mobile devices, and they cater to players' ever-changing needs. Figure 3.8 below shows the model of the PXN V3II steering wheel controller that used in this project.


Figure 3. 8 PXN V3II model.

### 3.4.3 Arduino UNO

Arduino UNO that shown in Figure 3.9 is used to control the trigger system. The trigger which consists of two pieces Light Emitting Diode (LED), vibration motor, and piezzo buzzer are used to trigger the driver when drowsy appearance is detected. Arduino UNO is selected to be the microcontroller because of its capability to control multiple devices in one time.

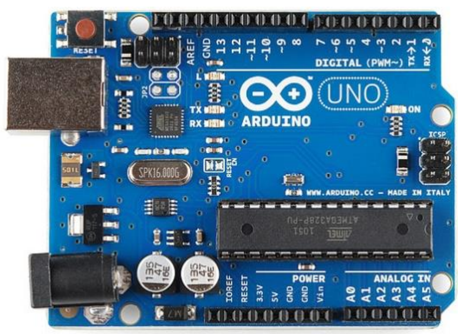



Figure 3. 9 Arduino UNO.

The pin used are Digital Input Output pin 5, 7, 8, and 9 for this project. Table 3.3 below shows the summary of Arduino UNO pin used.

Table 3. 3 Arduino Pin used and their function.

| Pin No. | Function |
|---|---|
| Digital I/O 5 | Vibration motor |
| Digital I/O 7 | Left LED |
| Digital I/O 8 | Right LED |
| Digital I/O 9 | Piezzo Buzzer |

### 3.4.4 Light Emitting Diode

The type of LED used is super bright type with red colour output. Its luminous intensity can be achieved from range 3000mcd until 4000mcd. The forward voltage is range 2V until 2.4V. The Figure 3.10 below shows the type of LED used in this project.

Figure 3. 10 Light Emitting Diode (LED).

### 3.4.5 Piezo Buzzer

The piezo buzzer or we also can call it passive buzzer requires pulses and frequencies to produce sounds. In this project, the frequency used is 3000Hz. The operating voltage for this buzzer is 5V which can directly connect to the Arduino UNO pin since the Arduino can provide up to 5V to digital input output pin. Figure 3.11 shows the buzzer used in this project.

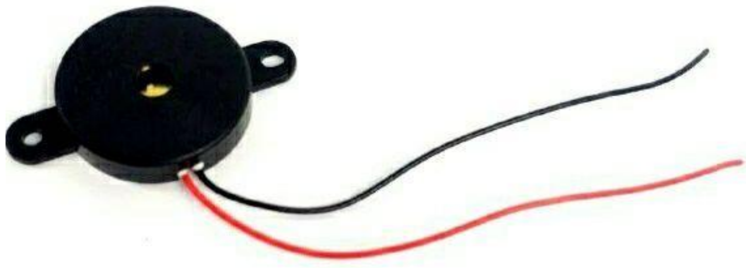


Figure 3. 11 Piezo Buzzer.

### 3.4.6 Vibration Motor

Vibration motor is used to vibrate the steering wheel when the drowsiness appearance is detected. This device gives 11000 rotation per minute (rpm) when run in 5V and the draw current is 100mA. The dimension of this motor is 10mm diameter and 2.7mm thick made it suitable to attach to the steering due to small size. Figure 3.12 shows the vibration motor used in this project.

Figure 3. 12 Mini Disc Vibrating Motor 1027.

### 3.4.7 Circuit Diagram and Real Development

Circuit diagram has been constructed by using Proteus software to demonstrate in simulation before build the real circuit. Figure 3.13 and Figure 3.14 show the circuit design and real development.

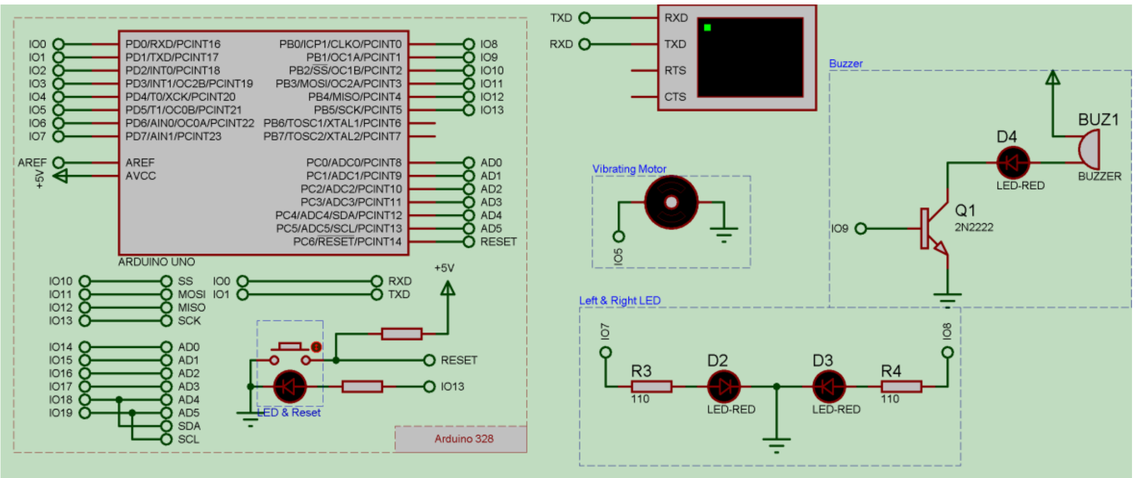



Figure 3. 13 Circuit Schematic of the project

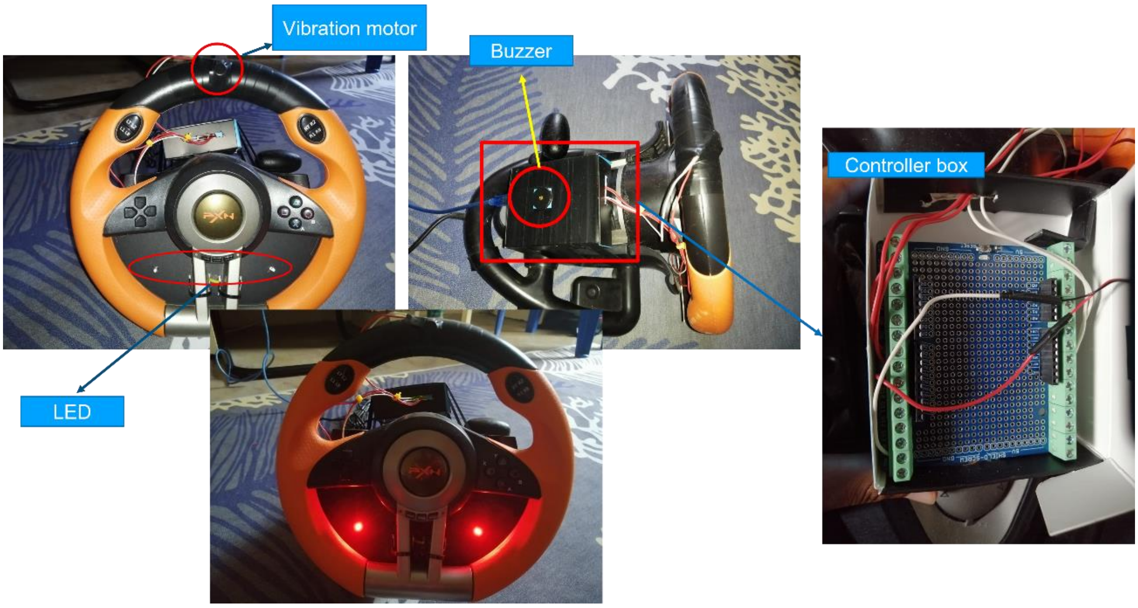



Figure 3. 14 Hardware Assembly

## 3.5 Software Development

Software development is a set of computer science activities concerned with the creation, design, deployment, and maintenance of software. Programmers, software engineers, and software developers are the people who work on software. These jobs interact and overlap, and the relationships between them vary a lot depending on the development department and community.

### 3.5.1 EEGID Mobile Application

Users can connect their mobile phone to the NeuroSky Mindwave Moblie Headset through Bluetooth using the eegID application screen illustrated in Figures 3.15, respectively. Users can use eegID by Isomer Programming LLC to connect to a NeuroSky(TM) MindWave Mobile Headset (or others compatible with ThinkGear(TM) technology) through Bluetooth(TM) to examine Electroencephalography (EEG) data, specifically these fields: Attention Level, Meditation Level, Blink Strength, PoorSignal, EEG Raw Value, EEG Raw Value Volts, EEG Raw Value, EEG Raw Value Volts, EEG Raw Value, EEG Raw Value Volts, EEG Raw Value, EEG Raw Value Delta(1-3Hz), Theta(4-7Hz), Alpha Low(8-9Hz), Alpha High(10-12Hz), Beta Low(13-17Hz), Beta High(18-30Hz), Gamma Low(31-40Hz), and Gamma Mid(31-40Hz) are the different types of frequencies (41-50Hz). The EEG data can be recorded for as long as the user wants it to be recorded, and it can be terminated at any time.

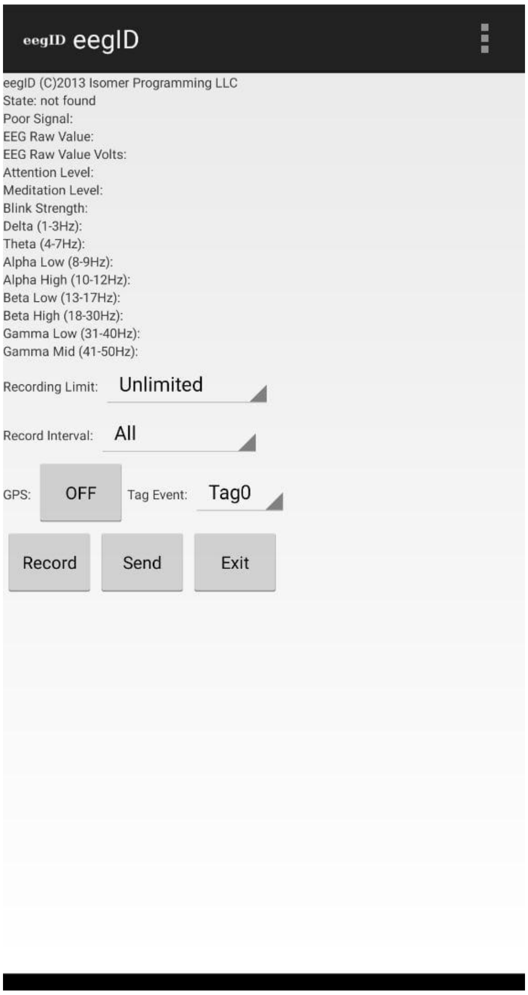



Figure 3. 15: eegID Mobile application home screen

### 3.5.2 MATLAB R2018a Software

Figure 3.16 shows MATLAB (matrix laboratory), a high-performance language for technical computing. It combines computing, programming, and visualization in a user-friendly interface, with problems and solutions written in standard mathematical notation. The feature extraction from the raw data is determined using MATLAB in this project. It was also used to calculate the standard deviation and mean of the EEG band, as well as the KNN classifier and GUI.

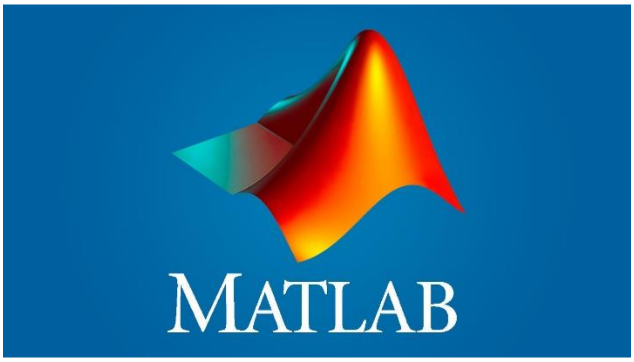



Figure 3. 16: MATLAB Software version 2018a

#### 3.5.2.1 MATLAB App Designer

App Designer is an application inside the Matlab software that provide tool for the user to build Graphical User Interface (GUI). The application is user friendly and easy to use. However, by using this application, it is unable to communicate directly with Arduino UNO. Figure 3.17 shows the view of App Designer inside MATLAB.

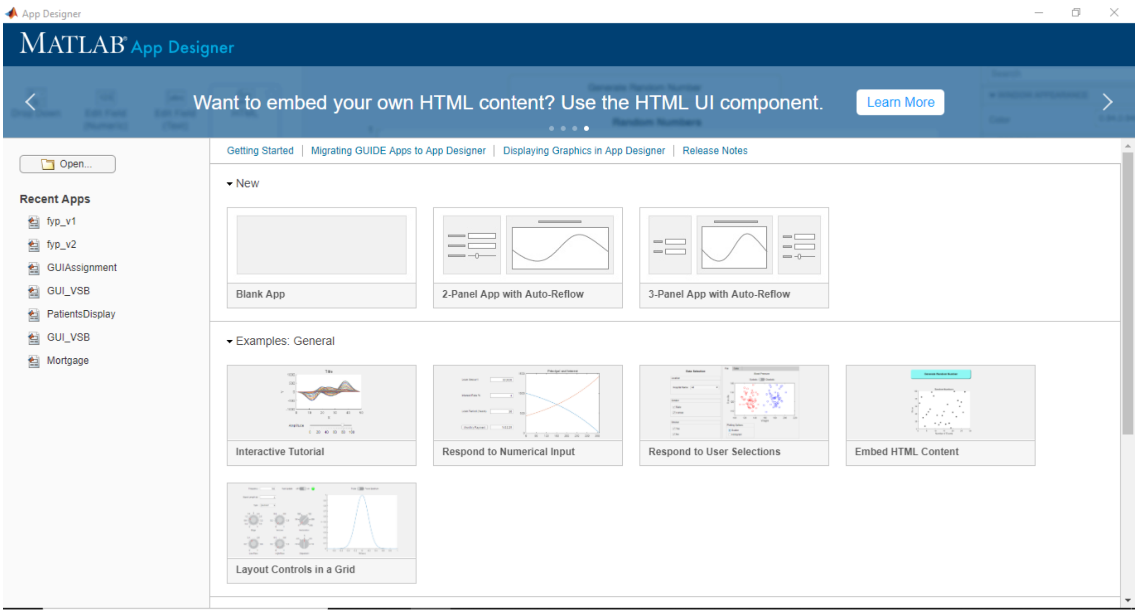



Figure 3. 17 App Designer in Matlab used for build Graphical User Interface (GUI).

# CHAPTER 4

# RESULTS AND DISCUSSION

## 4.1 Introduction

In this chapter, an analysis of the result of microsleep measurement of the proposed system will be presented and discussed.

## 4.2 EEG Raw Data

The EEG Raw Data was recorded from multiple subjects that were selected. The data collection for Active Condition and Relax Condition were recorded using the same subjects shown in Figure 4.1 below. Figure 4.2 shows the EEG Raw data selected from the CSV files from all subjects while Figure 4.3 and Figure 4.4 show the graph of EEG Raw data in the time domain for each subject. Table 4.1 shows the list of subjects with their gender and age.

Table 4. 1: List of subjects selected for this research

| Subject | Gender | Age |
|---|---|---|
| 1 | Male | 55 |
| 2 | Male | 30 |
| 3 | Male | 24 |
| 4 | Male | 23 |
| 5 | Male | 24 |

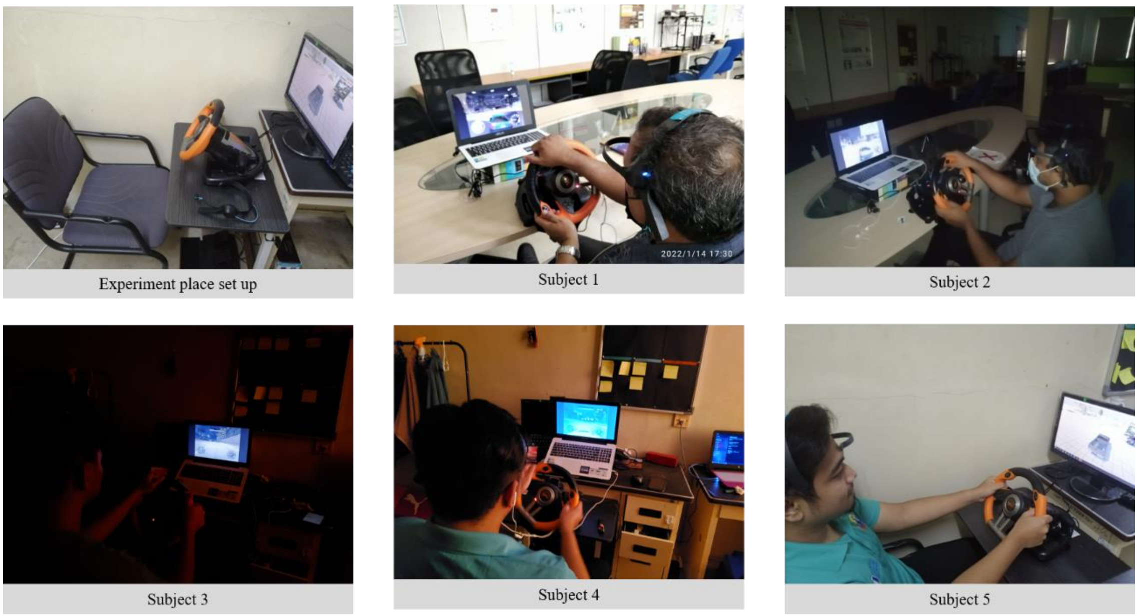
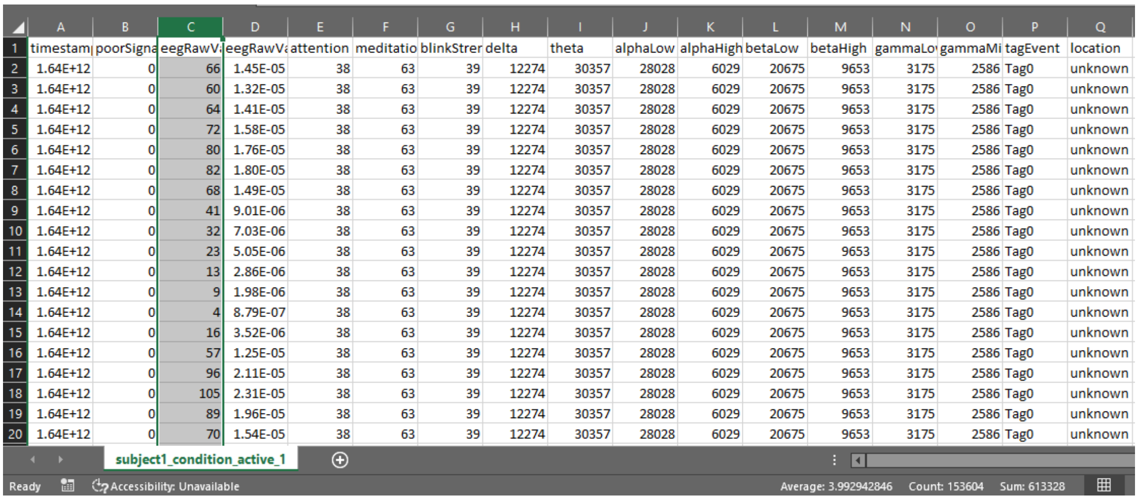



Figure 4. 1: Selected subjects for data recording.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | timestam | poorSigna | eegRawV | eegRawV | attention | meditatio | blinkStren | delta | theta | alphaLow | alphaHigh | betaLow | betaHigh | gammaLo | gammaMi | tagEvent | location |
| 2 | 1.64E+12 | 0 | 66 | 1.45E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 3 | 1.64E+12 | 0 | 60 | 1.32E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 4 | 1.64E+12 | 0 | 64 | 1.41E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 5 | 1.64E+12 | 0 | 72 | 1.58E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 6 | 1.64E+12 | 0 | 80 | 1.76E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 7 | 1.64E+12 | 0 | 82 | 1.80E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 8 | 1.64E+12 | 0 | 68 | 1.49E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 9 | 1.64E+12 | 0 | 41 | 9.01E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 10 | 1.64E+12 | 0 | 32 | 7.03E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 11 | 1.64E+12 | 0 | 23 | 5.05E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 12 | 1.64E+12 | 0 | 13 | 2.86E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 13 | 1.64E+12 | 0 | 9 | 1.98E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 14 | 1.64E+12 | 0 | 4 | 8.79E-07 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 15 | 1.64E+12 | 0 | 16 | 3.52E-06 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 16 | 1.64E+12 | 0 | 57 | 1.25E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 17 | 1.64E+12 | 0 | 96 | 2.11E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 18 | 1.64E+12 | 0 | 105 | 2.31E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 19 | 1.64E+12 | 0 | 89 | 1.96E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |
| 20 | 1.64E+12 | 0 | 70 | 1.54E-05 | 38 | 63 | 39 | 12274 | 30357 | 28028 | 6029 | 20675 | 9653 | 3175 | 2586 | Tag0 | unknown |



Figure 4. 2: The highlighted is the EEG raw value that was selected for analysis.

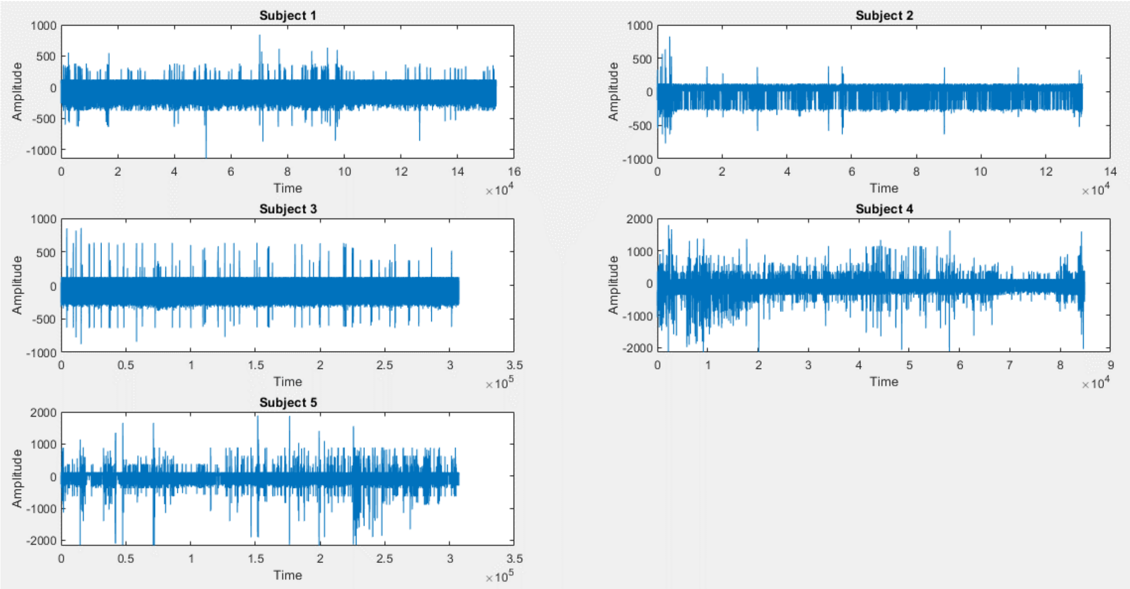



Figure 4. 3: Raw EEG Data for each subject in active condition.

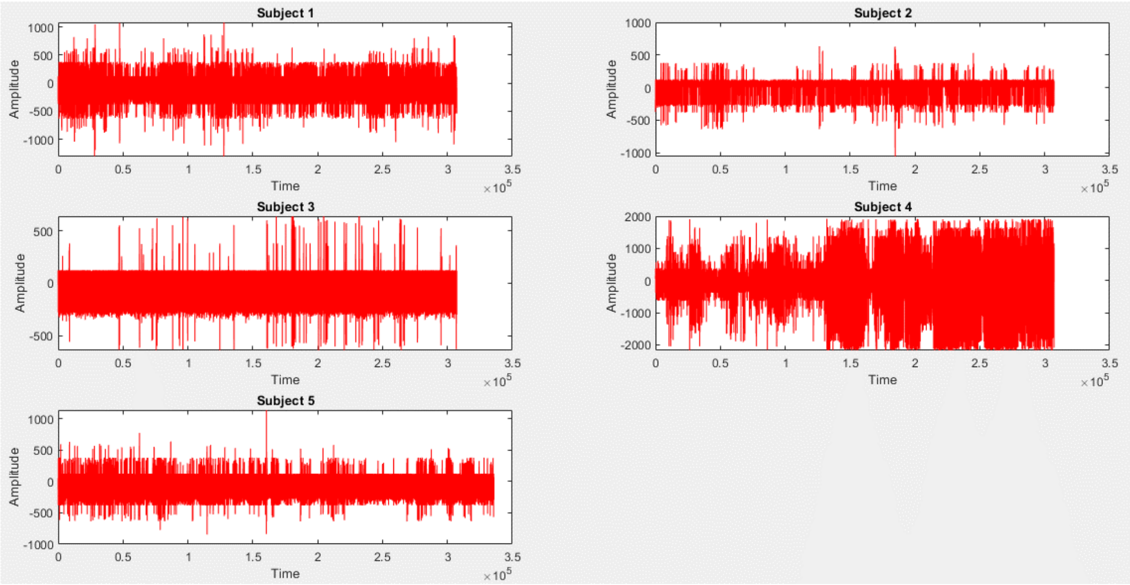



Figure 4. 4: Raw EEG Data for each subject in relaxed condition

### 4.2.1 EEG Raw Data in Frequency Domain

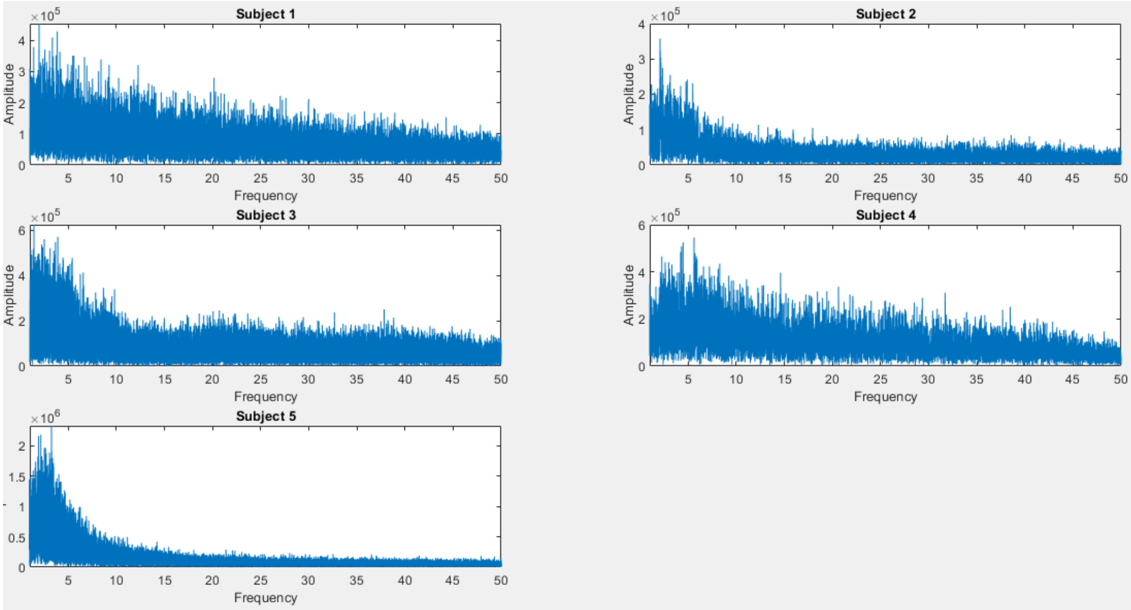



Figure 4. 5 FFT of the EEG Raw Data from the active condition in the frequency domain.

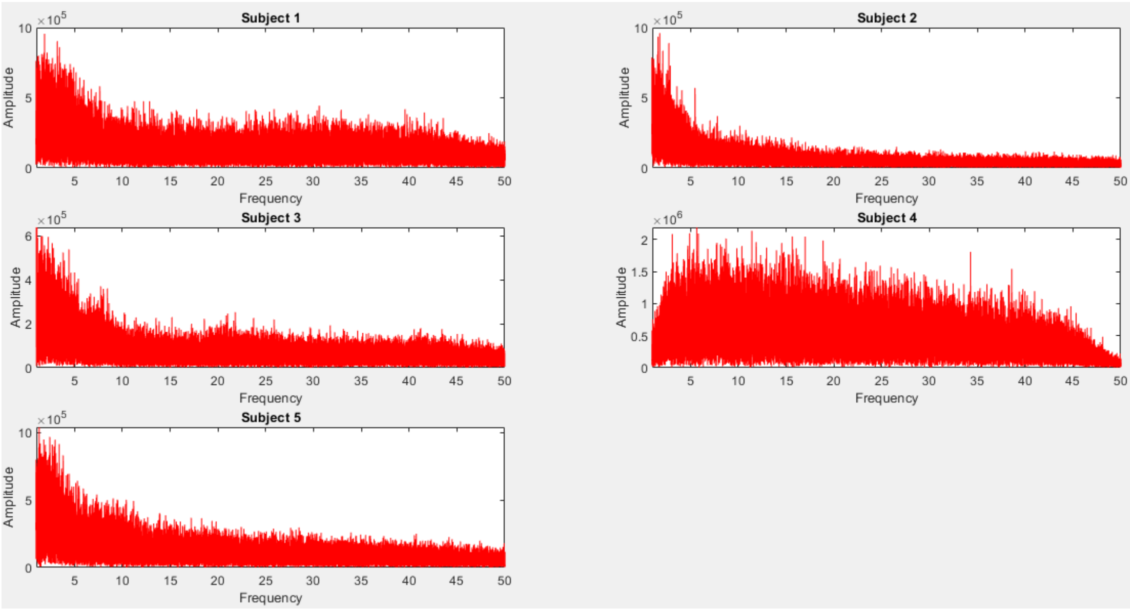



Figure 4. 6 FFT of the EEG Raw Data from the relaxed condition in the frequency domain.

### 4.2.2 EEG Feature Extraction

The purpose of feature extraction is to extract the specific band wave as illustrated in Figure 4.7 until Figure 4.17 where theta, alpha, and the beta band were extracted from the raw EEG signal. Then, the standard deviation, variance, and mean of each band were calculated from the power spectrum density of the band as illustrated for each subject. Figure 4.18 until Figure 4.23 are the analysis that is being observed to determine what band is at the highest amplitude during the active and drowsy condition.

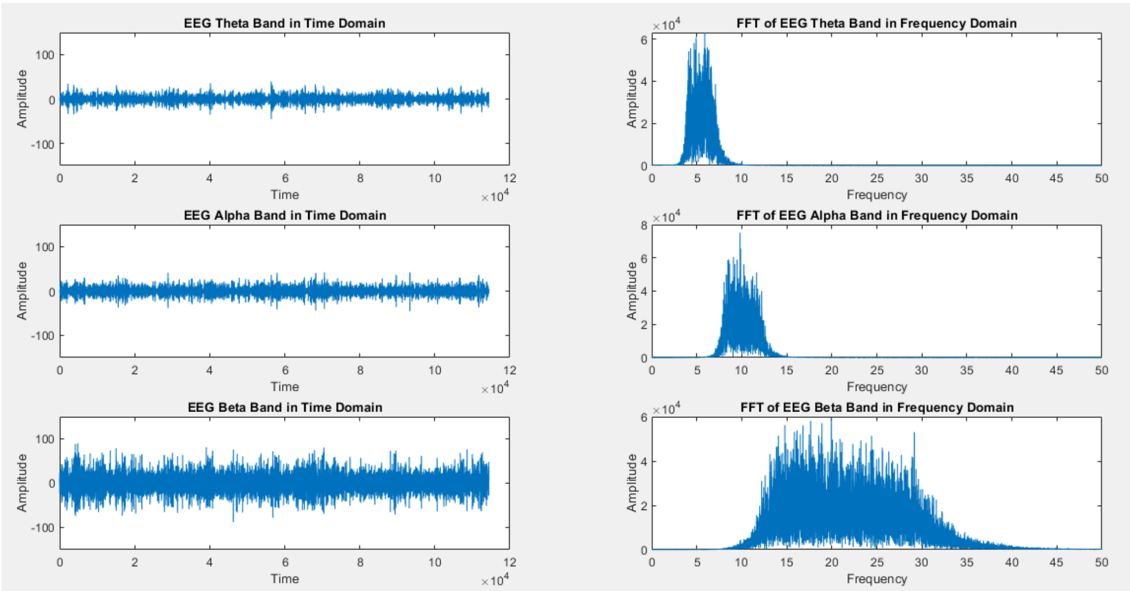



Figure 4. 7 Theta, Alpha, and Beta band in the time domain and frequency domain

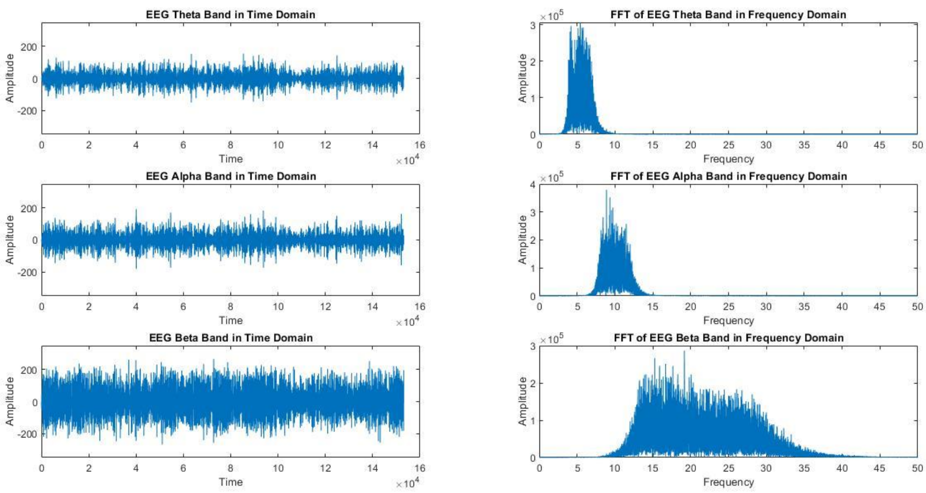



Figure 4. 8 Theta, Alpha, and Beta band for Subject 1 during active state.

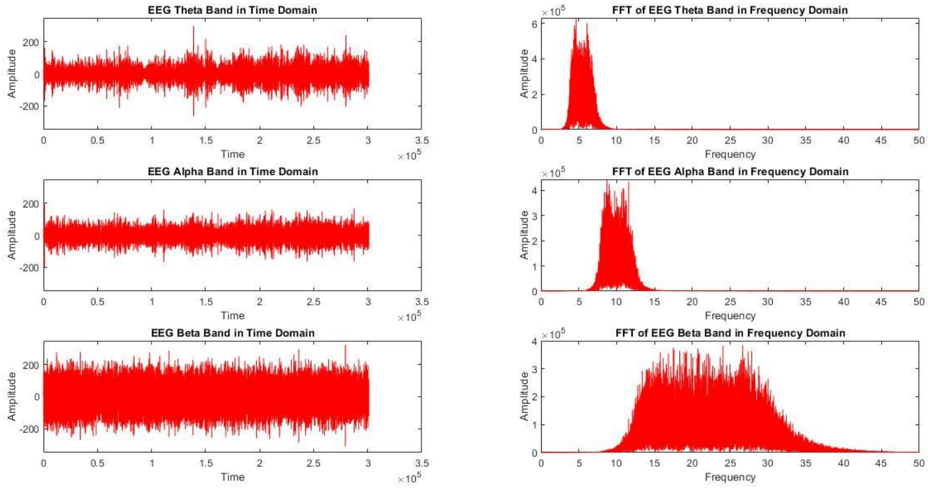



Figure 4. 9 Theta, Alpha, and Beta band for Subject 1 during drowsy state.

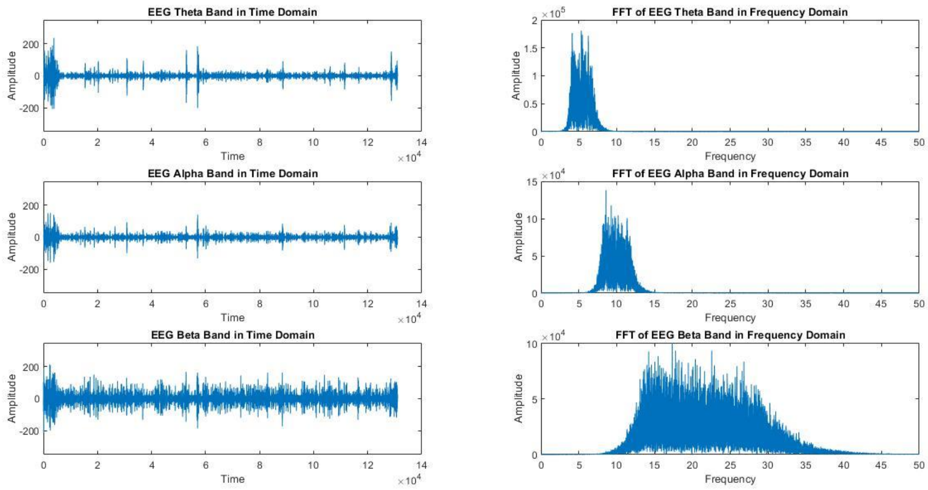



Figure 4. 10 Theta, Alpha, and Beta band for Subject 2 during active state.

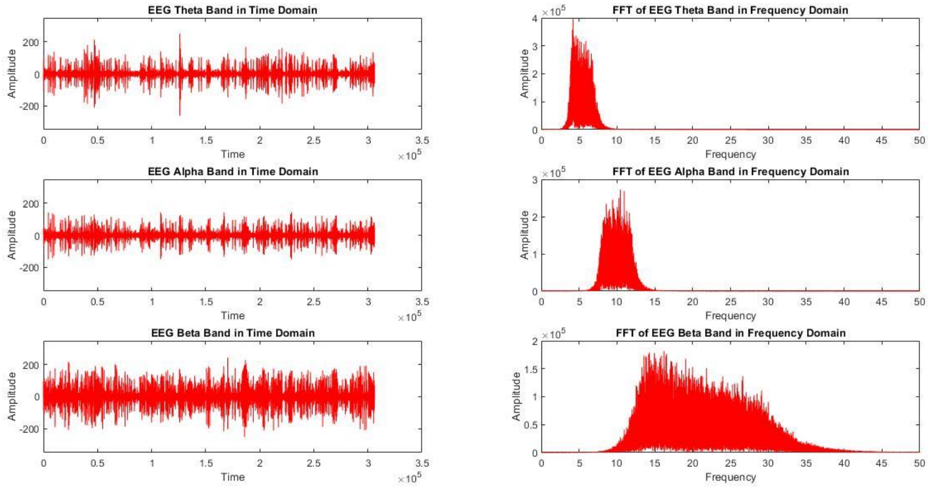



Figure 4. 11 Theta, Alpha, and Beta band for Subject 2 during drowsy state.

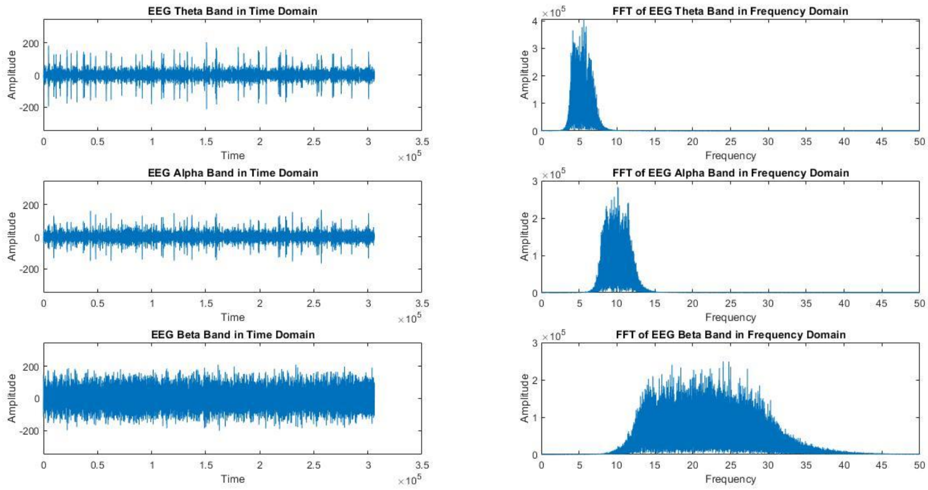



Figure 4. 12 Theta, Alpha, and Beta band for Subject 3 during active state.

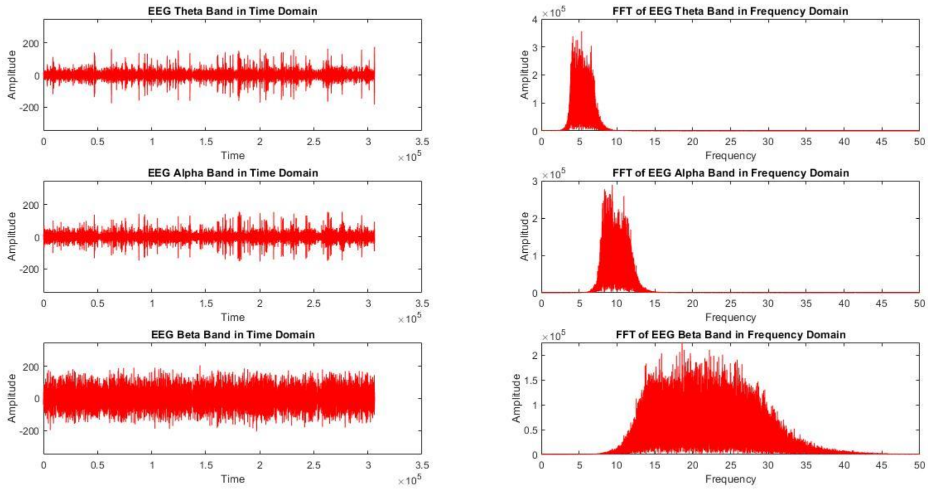



Figure 4. 13 Theta, Alpha, and Beta band for Subject 3 during drowsy state.

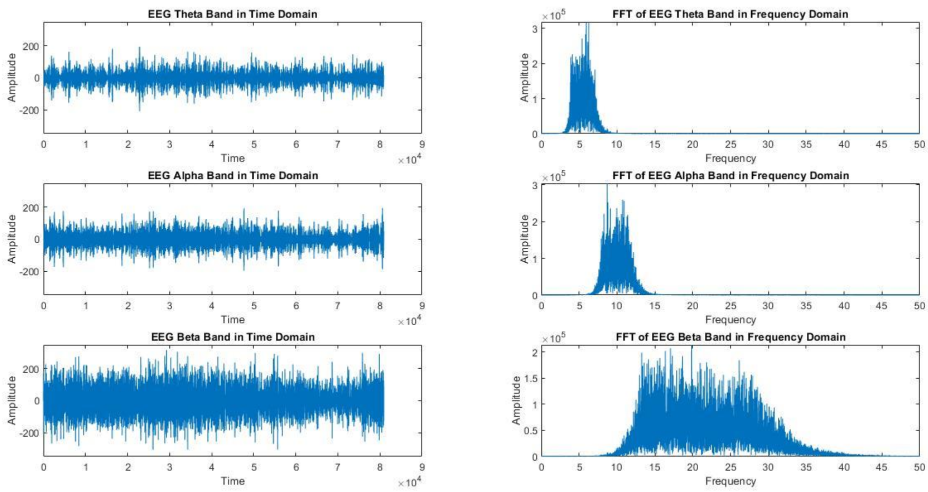



Figure 4. 14 Theta, Alpha, and Beta band for Subject 4 during active state.

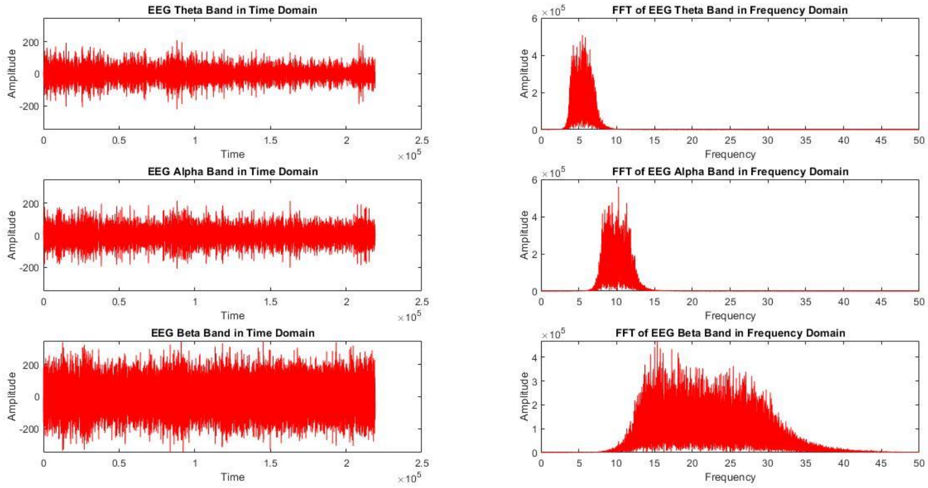



Figure 4. 15 Theta, Alpha, and Beta band for Subject 4 during drowsy state.

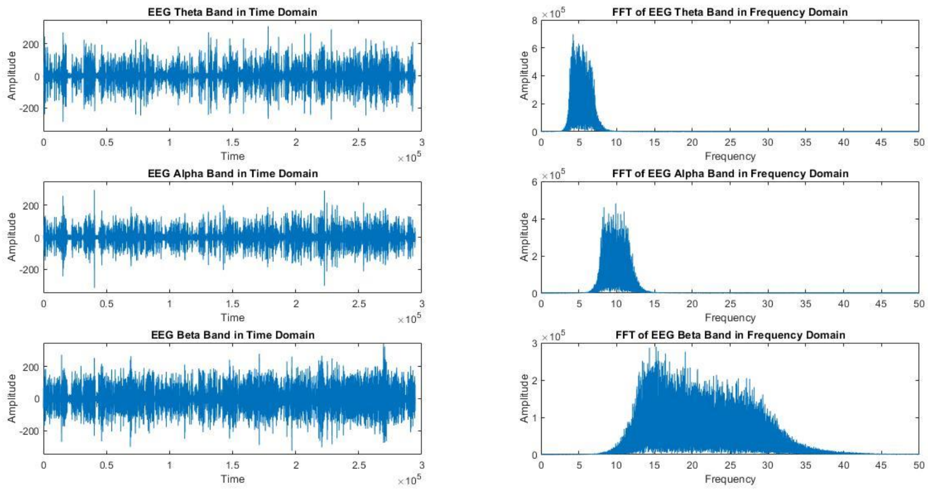



Figure 4. 16 Theta, Alpha, and Beta band for Subject 5 during active state.

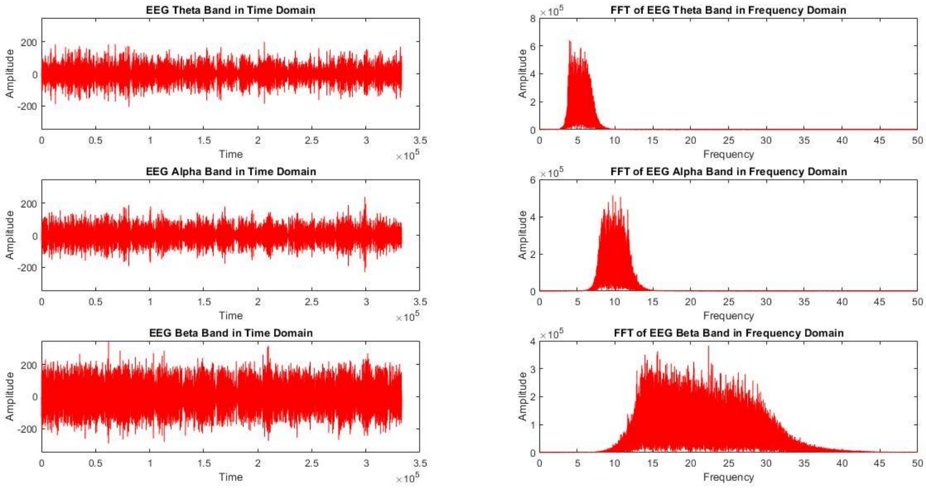
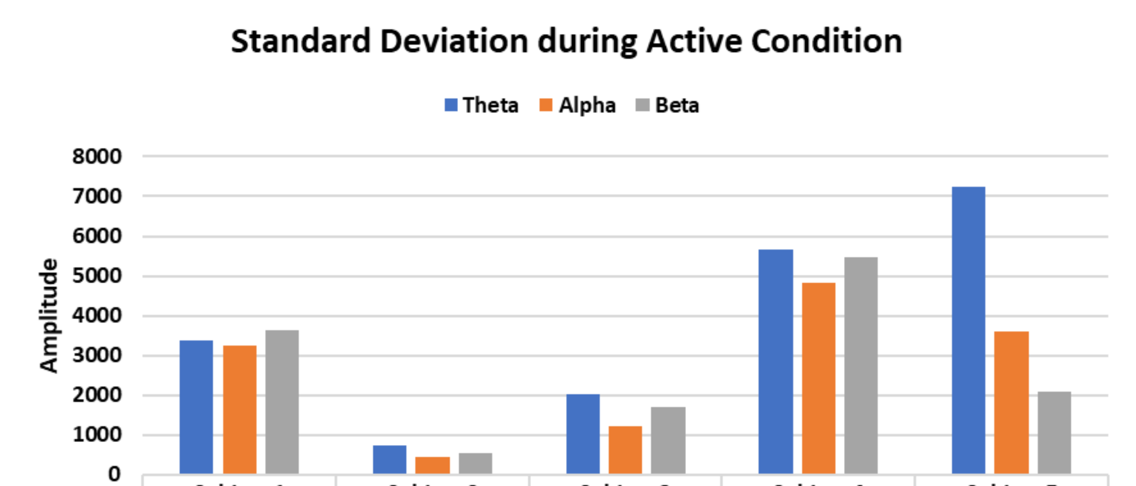
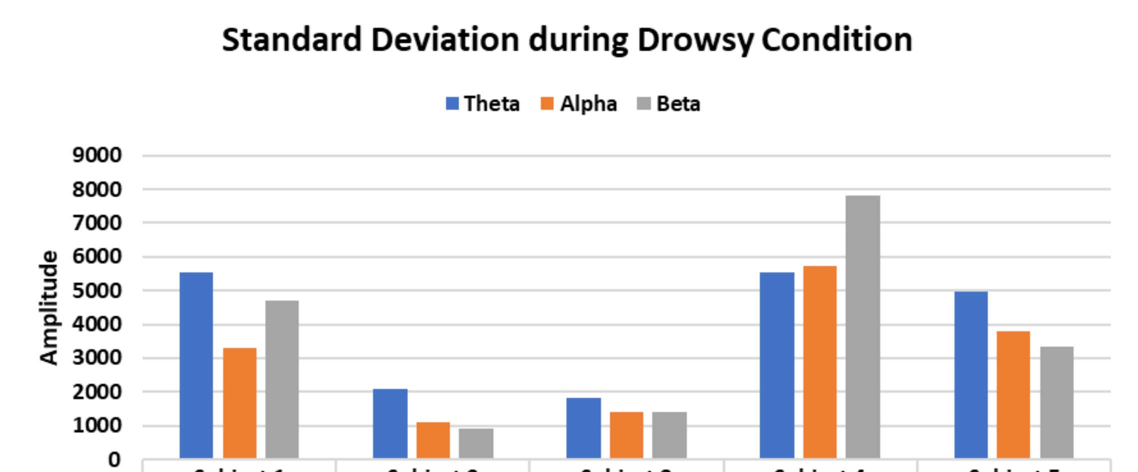



Figure 4. 17 Theta, Alpha, and Beta band for Subject 5 during drowsy state.



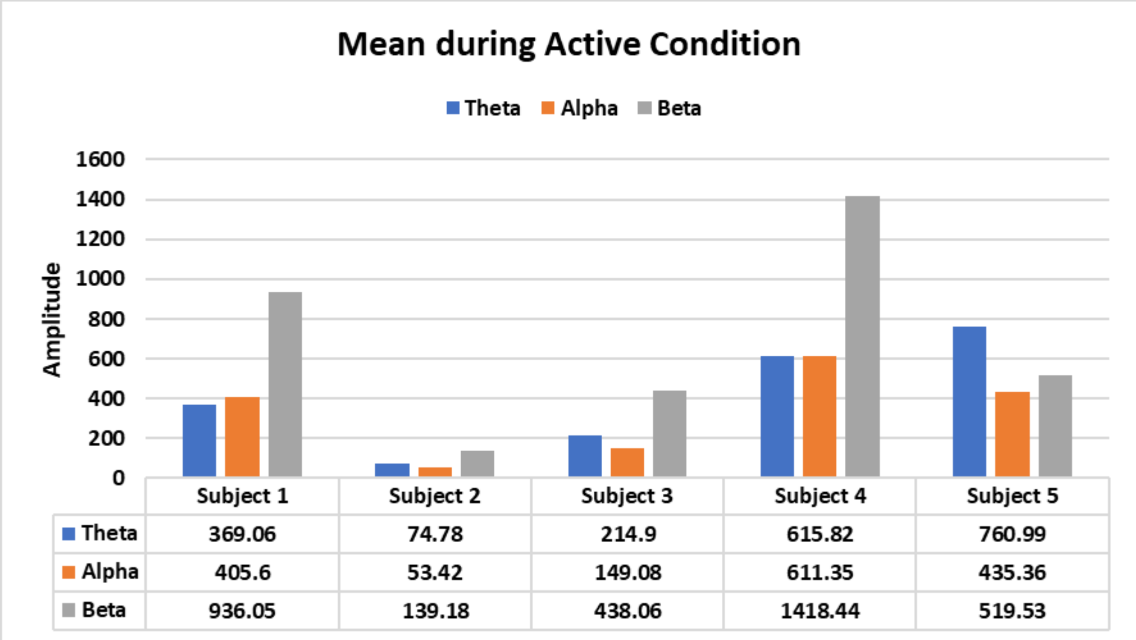

| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 3358.874 | 725.62 | 2028.71 | 5664.85 | 7250.79 |
| Alpha | 3248.29 | 430.05 | 1199.85 | 4829.54 | 3596.28 |
| Beta | 3640.18 | 532.03 | 1683.03 | 5466.7 | 2073.4 |

Figure 4. 18 Histogram of analyzing standard deviation of Theta, Alpha, and Beta band of all subjects during active state.



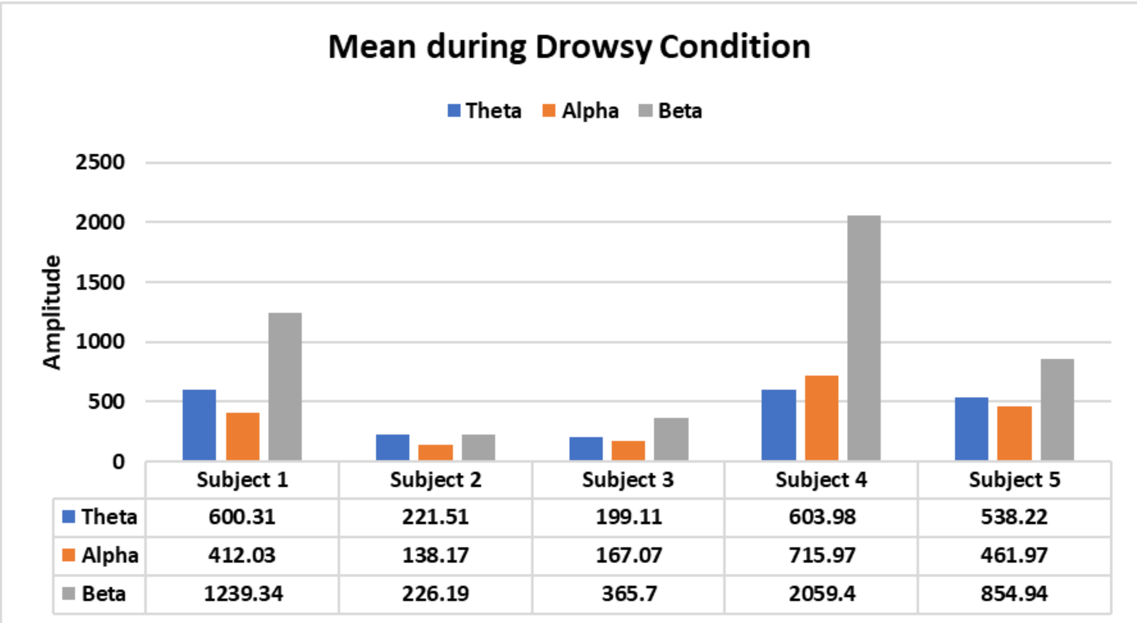

| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 5535.68 | 2074.73 | 1831.08 | 5537.77 | 4974.85 |
| Alpha | 3309.35 | 1117.1 | 1384.33 | 5724.21 | 3777.91 |
| Beta | 4684.23 | 895.65 | 1409.83 | 7810.44 | 3340.52 |

Figure 4. 19 Histogram of analyzing standard deviation of Theta, Alpha, and Beta band of all subjects during drowsy state.



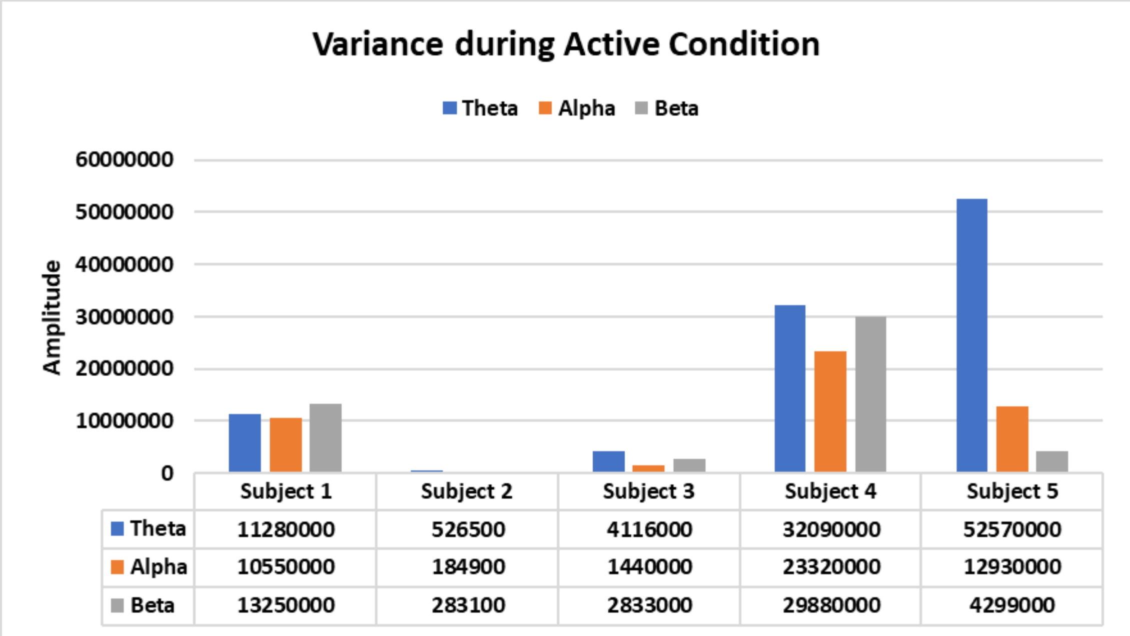

| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 369.06 | 74.78 | 214.9 | 615.82 | 760.99 |
| Alpha | 405.6 | 53.42 | 149.08 | 611.35 | 435.36 |
| Beta | 936.05 | 139.18 | 438.06 | 1418.44 | 519.53 |

Figure 4. 20 Histogram of analyzing mean of Theta, Alpha, and Beta band of all subjects during active state.



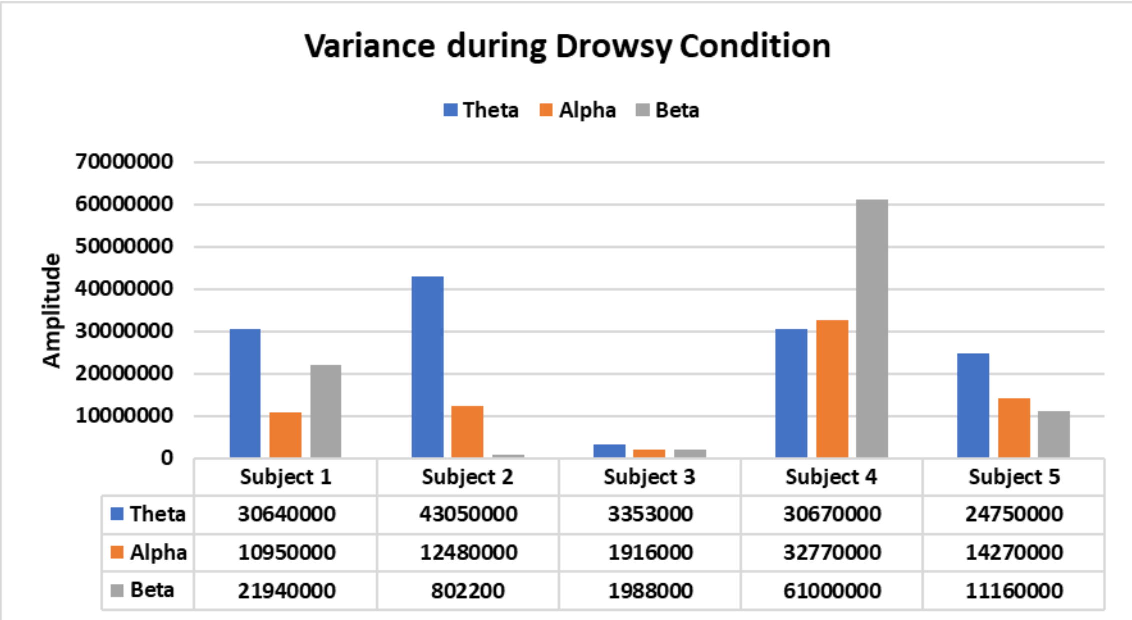

| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 600.31 | 221.51 | 199.11 | 603.98 | 538.22 |
| Alpha | 412.03 | 138.17 | 167.07 | 715.97 | 461.97 |
| Beta | 1239.34 | 226.19 | 365.7 | 2059.4 | 854.94 |

Figure 4. 21 Histogram of analyzing mean of Theta, Alpha, and Beta band of all subjects during drowsy state.




| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 11280000 | 526500 | 4116000 | 32090000 | 52570000 |
| Alpha | 10550000 | 184900 | 1440000 | 23320000 | 12930000 |
| Beta | 13250000 | 283100 | 2833000 | 29880000 | 4299000 |

Figure 4. 22 Histogram of analyzing variance of Theta, Alpha, and Beta band of all subjects during active state.




| | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|---|---|---|---|---|---|
| Theta | 30640000 | 43050000 | 3353000 | 30670000 | 24750000 |
| Alpha | 10950000 | 12480000 | 1916000 | 32770000 | 14270000 |
| Beta | 21940000 | 802200 | 1988000 | 61000000 | 11160000 |

Figure 4. 23 Histogram of analyzing variance of Theta, Alpha, and Beta band of all subjects during drowsy state.

## 4.3 Graphical User Interface (GUI)

The GUI was created after building the block diagram in Chapter 3, which is used for the user interacting with the system as well as illustrating the drowsiness signal and results. Figure 4.24 depicts the total number of subjects and their behavior for this project, which is divided into two categories; Relaxed State and Driving State. In this GUI, there are five graphics that show the EEG signal processing which are Raw EEG Data, Filtered EEG Data in Frequency Domain, Theta Band, Alpha Band, and Beta Band. The other three graphics show the EEG feature extraction which are, standard deviation, mean, and variance.

On top left part, the user can select which subject and what their state to analyze the signal. There is a section at the bottom part which tell current subject and their state while analyzing the signal. The driver microsleep or drowsiness appearance status can be seen by using a light indicator which will turn to red colour if drowsiness appearance is detected while green colour if it is not detected based on comparison between Theta and Beta Band.

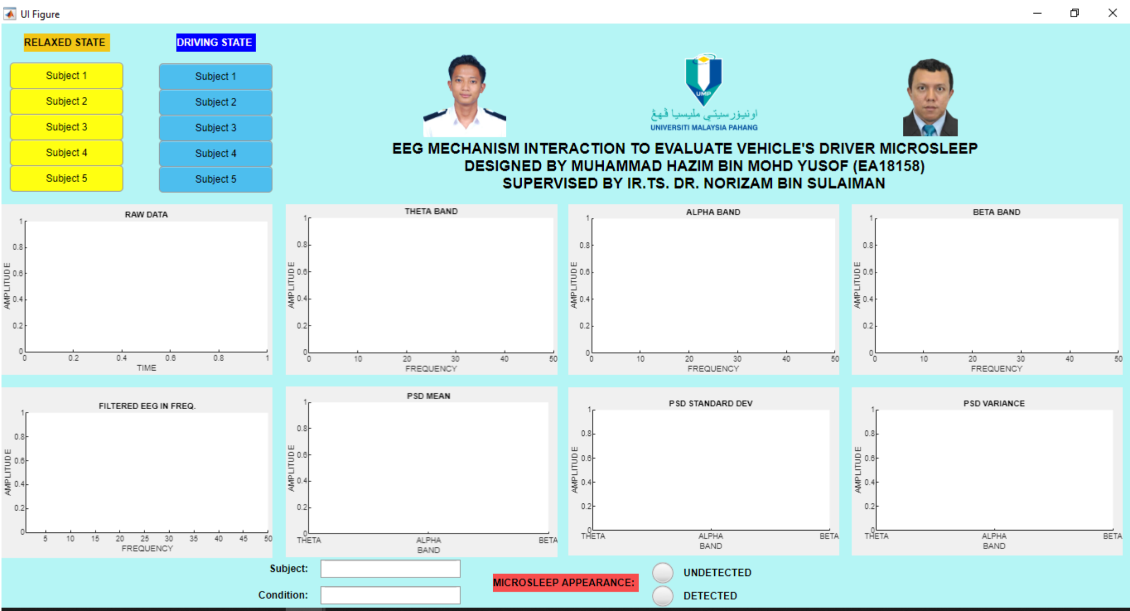



Figure 4. 24 Design of GUI in Matlab.

Figure 4.25 shows below the result of GUI for Subject 1 during Relaxed State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

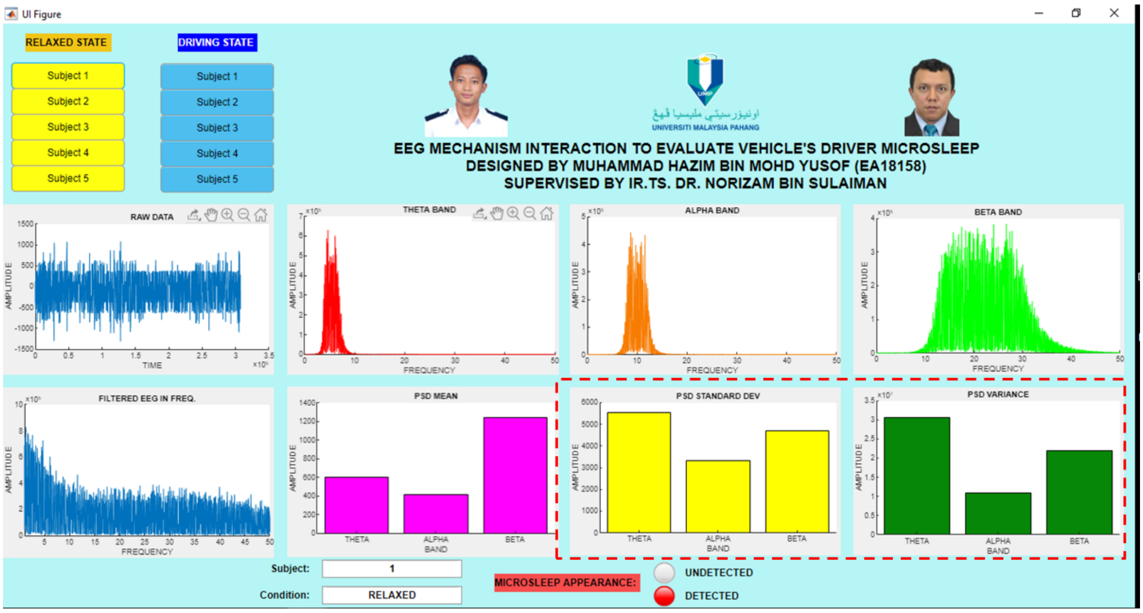



Figure 4. 25 GUI for Subject 1 during Relaxed State.

Figure 4.26 shows below the result of GUI for Subject 2 during Relaxed State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

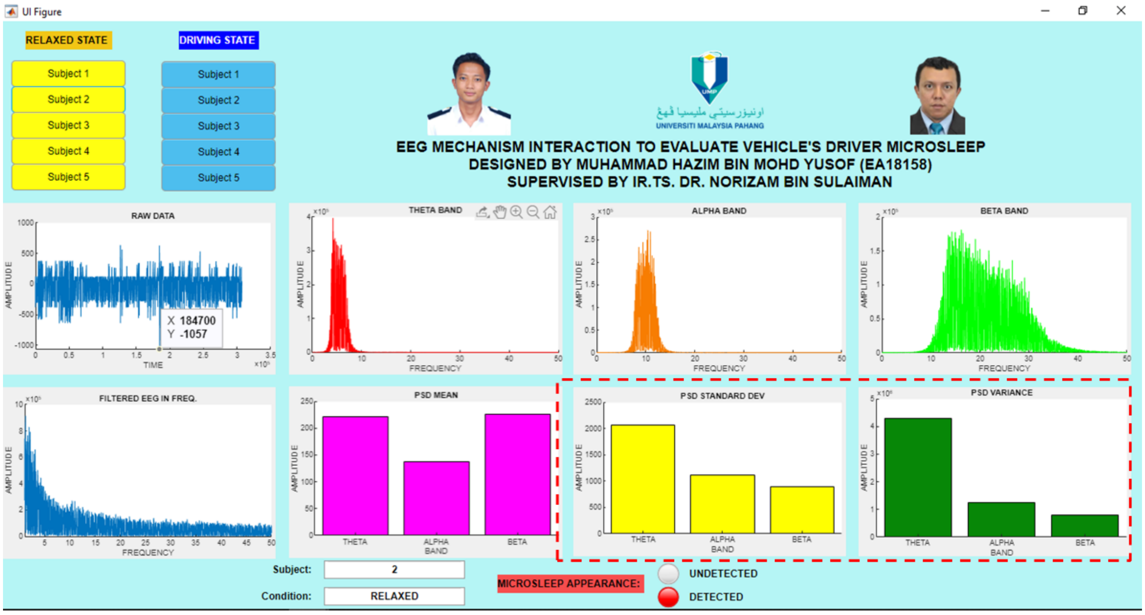



Figure 4. 26 GUI for Subject 2 during Relaxed State.

Figure 4.27 shows below the result of GUI for Subject 3 during Relaxed State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

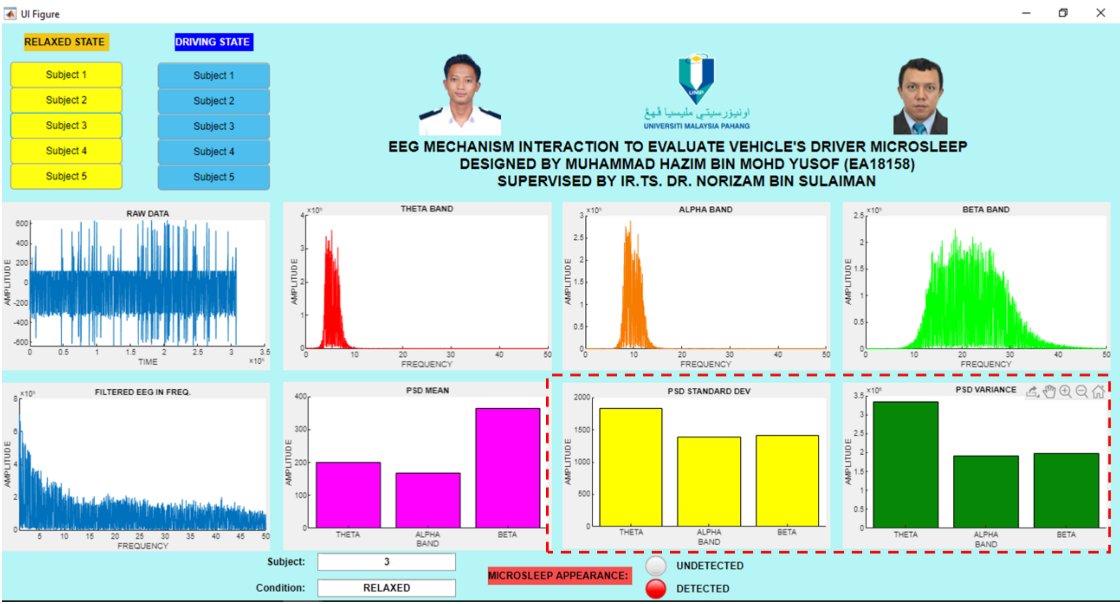



Figure 4. 27 GUI for Subject 3 during Relaxed State.

Figure 4.28 shows below the result of GUI for Subject 4 during Relaxed State. By referring to the indicator of microsleep appearance, the green light is showed and it not detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is lower than Beta Band and drowsinees appearance is not detected.

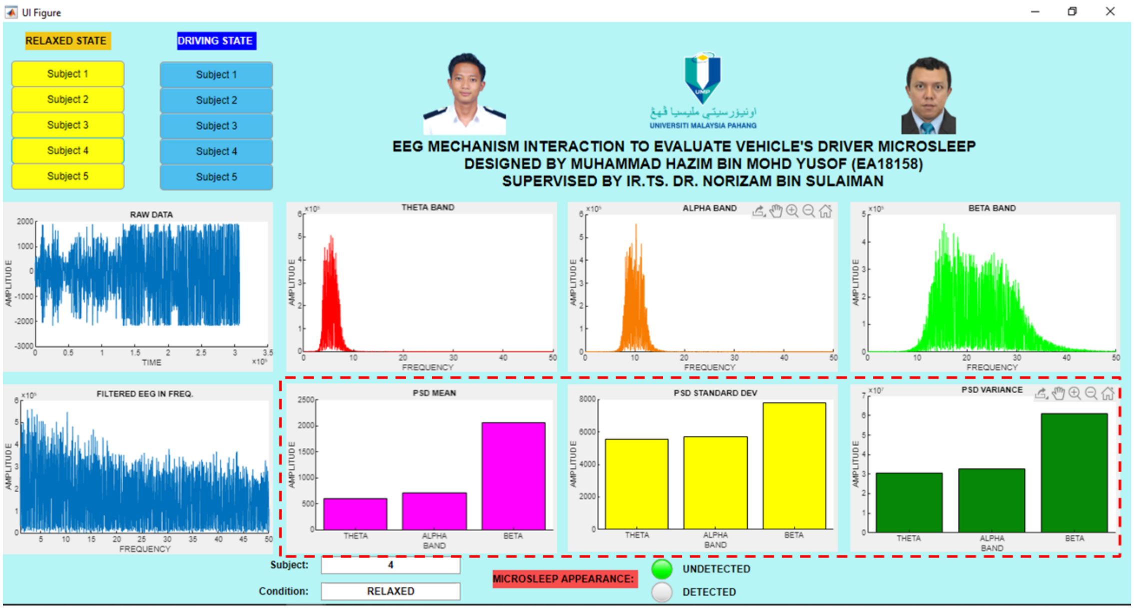



Figure 4. 28 GUI for Subject 4 during Relaxed State.

Figure 4.29 shows below the result of GUI for Subject 5 during Relaxed State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

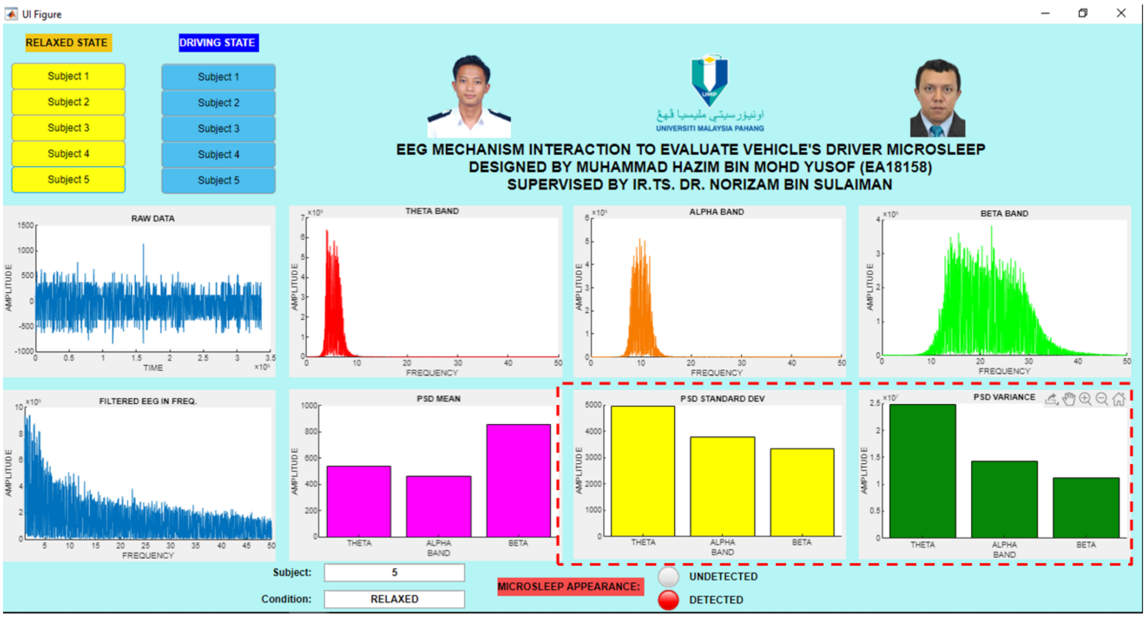



Figure 4. 29 GUI for Subject 5 during Relaxed State.

Figure 4.30 shows below the result of GUI for Subject 1 during Driving State. By referring to the indicator of microsleep appearance, the green light is showed and it not detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is lower than Beta Band and drowsinees appearance is not detected.

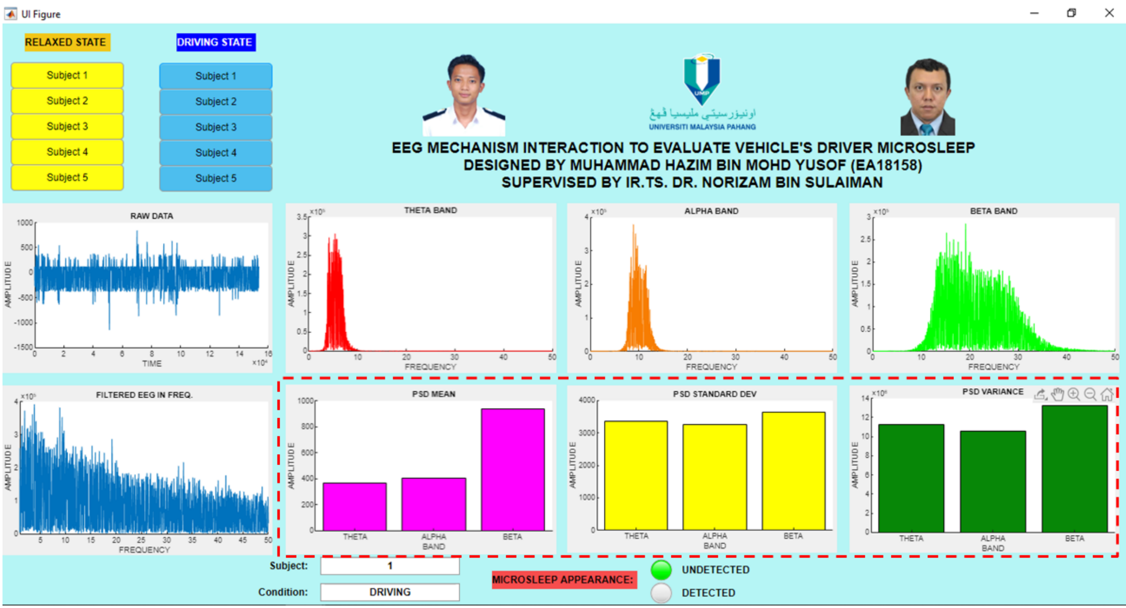



Figure 4. 30 GUI for Subject 1 during Driving State.

Figure 4.31 shows below the result of GUI for Subject 2 during Driving State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

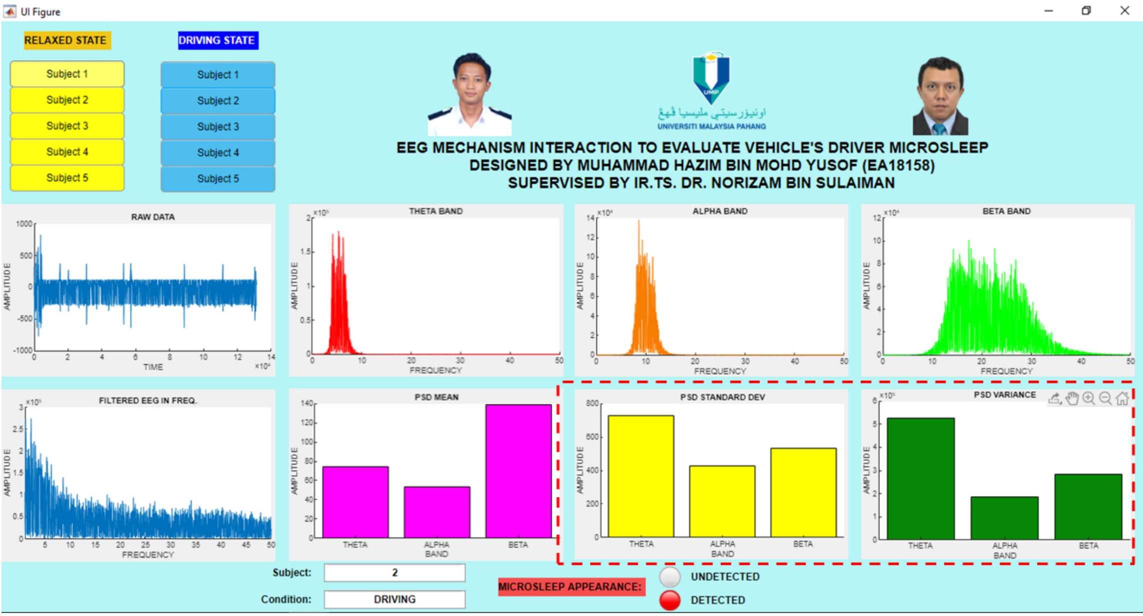



Figure 4. 31 GUI for Subject 2 during Driving State.

Figure 4.32 shows below the result of GUI for Subject 3 during Driving State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

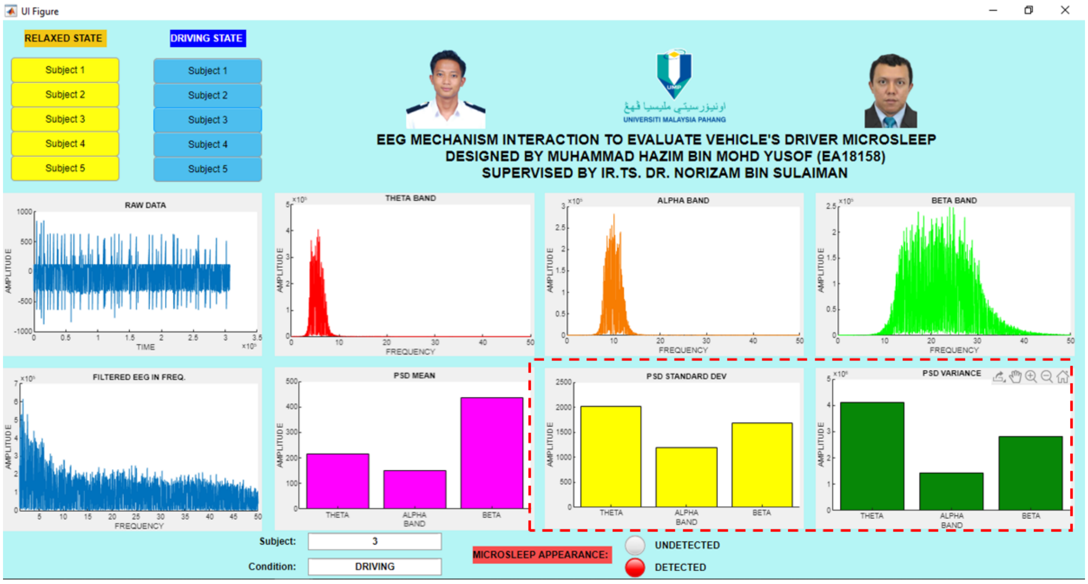



Figure 4. 32 GUI for Subject 3 during Driving State.

Figure 4.33 shows below the result of GUI for Subject 4 during Driving State. By referring to the indicator of microsleep appearance, the green light is showed and it not detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is lower than Beta Band and drowsinees appearance is not detected.

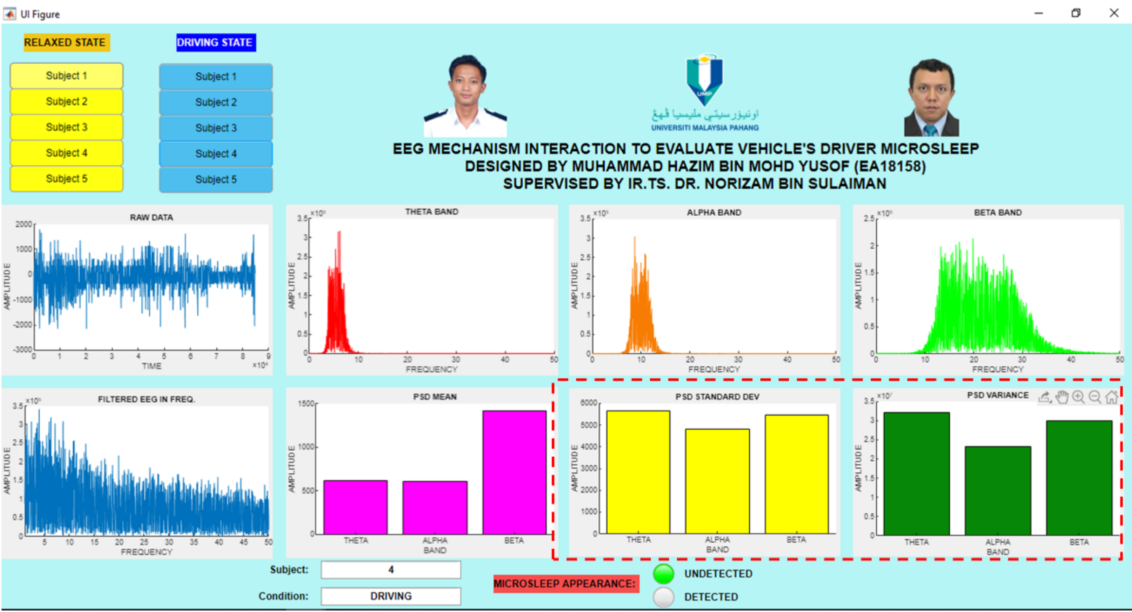



Figure 4. 33 GUI for Subject 4 during Driving State.

Figure 4.34 shows below the result of GUI for Subject 5 during Driving State. By referring to the indicator of microsleep appearance, the red light is showed and it detect the drowsiness appearance. Thus, by compare the strength of Theta Band and Beta Band, the Theta Band is higher than Beta Band and drowsinees appearance is detected.

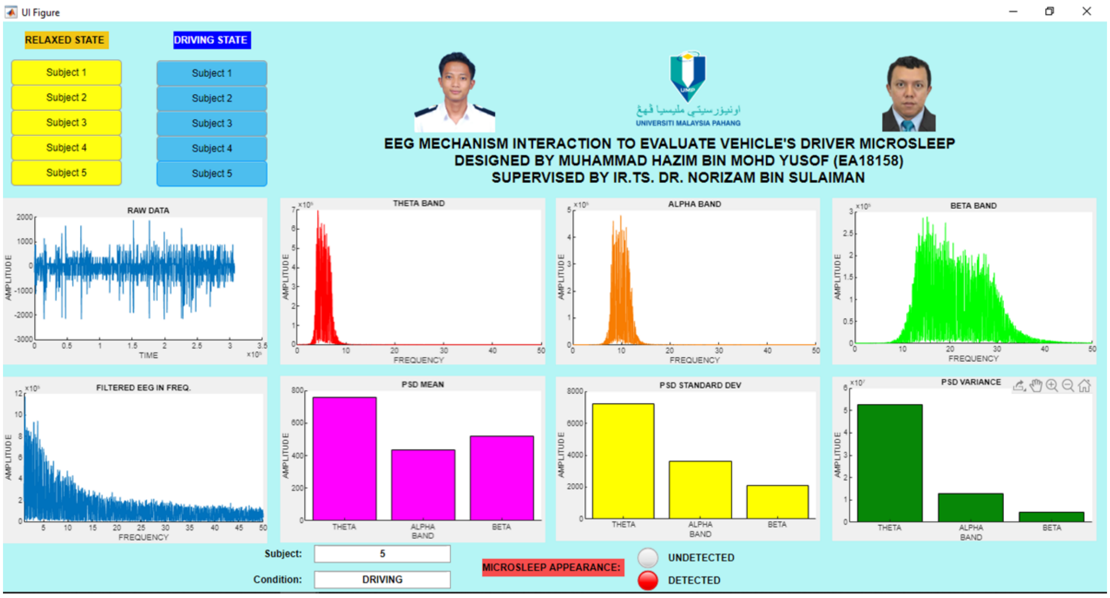



Figure 4. 34 GUI for Subject 5 during Driving State.

Based on the analysis, Table 4.2 below shows the summarisation for subjects that had drowsy appearance detected during both active and drowsy conditions.

Table 4. 2 Result of drowsy appearance for all subjects.

| SUBJECT | AGE | GENDER | DROWSY APPEARANCE | |
|---|---|---|---|---|
| | | | ACTIVE CONDITION | DROWSY CONDITION |
| 1 | >50 | MALE | ✗ | ✓ |
| 2 | 27 | MALE | ✓ | ✓ |
| 3 | 24 | MALE | ✓ | ✓ |
| 4 | 24 | MALE | ✗ | ✗ |
| 5 | 24 | MALE | ✓ | ✓ |

## 4.4 KNN Classification

In features classification, the ratio of testing-training that were chose was (75:25) percentage due to a lot number of data which is 37000 data. For the features selection that used in this project are raw eeg data, theta value, alpha value, and beta value. The idea was to indicate which features can be used to compare between drowsy state and active state. The highest accuracy of the result shows the best features selection. The model of classification also was analyzed on which model of K-NN Classification is the most suitable for these data. The model of K-NN classification that were analyzed are Fine K-NN, Medium K-NN, Coarse K-NN and Weighted K-NN. The classification was handled by using Matlab App Classifier that shown in Figure 4.35.

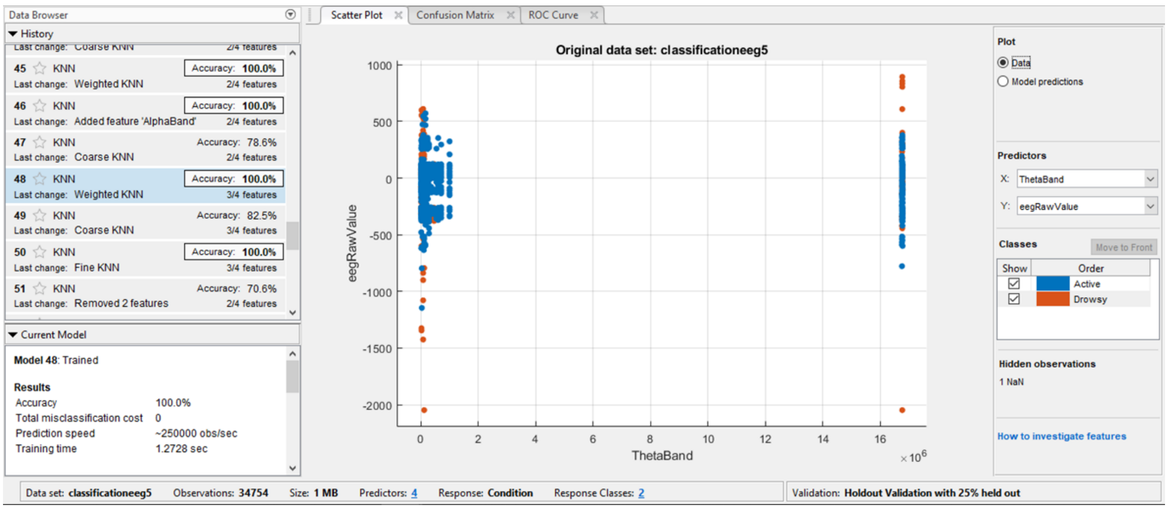



Figure 4. 35 App Classifier in Matlab that used in this porject.

Multiple numbers of test had been run and the result of the classification can be seen in Table 4.3 that shown below. The result shows that Theta band is the best features selection as it gave the highest accuracy (99.4%) by using Weighted KNN, Fine KNN, and Medium KNN. For two features combined, the best choice is Theta and Beta band features which gave accuracy at 100% for Fine KNN, Medium KNN, and Weighted KNN while 81.2% for Coarse KNN. In selection which classifier is the best, all features selection is combined and Weighted KNN is the best classifier with the accuracy at 77.3%. Figures 4.36 until Figures 4.37 shows the matrix confusion of the result.

Table 4. 3 Accuracy of Features Selection with different Classifier.

| Features Selection | | | | Accuracy (%) | | | |
|---|---|---|---|---|---|---|---|
| | | | | Fine KNN | Medium KNN | Coarse KNN | Weighted KNN |
| | | | EEGRawData | 48.7 | 48.8 | 48.7 | 48.7 |
| | | | Theta Band | 99.4 | 99.4 | 77.7 | 99.4 |
| | | | Alpha Band | 99.3 | 99.3 | 71.8 | 99.3 |
| | | | Beta Band | 99.3 | 99.3 | 79.4 | 99.3 |
| | | Theta Band | Beta Band | 100 | 100 | 81.2 | 100 |
| | | Theta Band | Alpha Band | 100 | 100 | 78.6 | 100 |
| | | EEGRawData | Theta Band | 70.6 | 63.2 | 64.5 | 71 |
| | | EEGRawData | Alpha Band | 65.3 | 57.9 | 57.4 | 65.7 |
| | | EEGRawData | Beta Band | 72.3 | 68.2 | 69.2 | 73.1 |
| | Theta Band | Alpha Band | Beta Band | 100 | 100 | 82.5 | 100 |
| EEGRawData | Theta Band | Alpha Band | Beta Band | 75.7 | 70.8 | 67.6 | 77.3 |

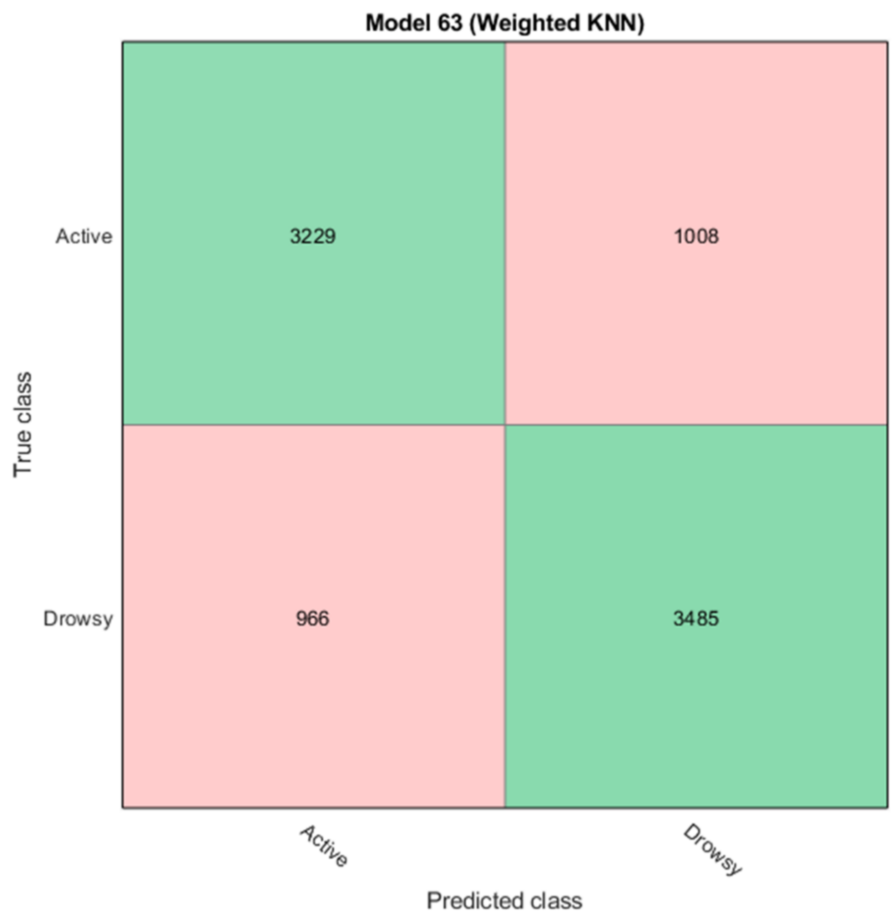



Figure 4. 36 Confusion matrix of true positive and true negative of the testing data.

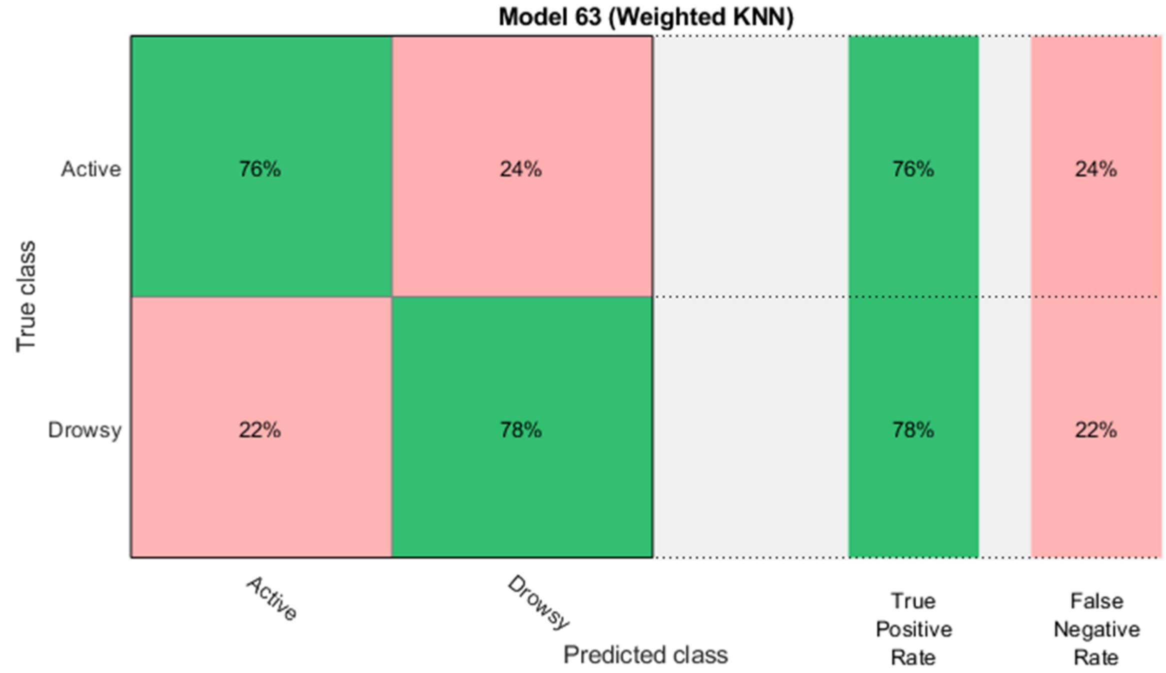



Figure 4. 37 Confusion matrix result by selecting EEG Raw Data, Theta Band, Alpha Band, and Beta Band as features selection.

## 4.5 Summary

In a nutshell, the research's findings and outputs were plotted and tallied in order to examine and determine the microsleep event. The study used EEG data from five people of varying ages to explore sleepiness accuracy and see if the age and gender of the participants had an impact on the results. In addition, the classification accuracy is tested using Weighted KNN classification to determine the appropriate training to testing ratio to employ in this study, which is 72:25. In a nutshell, the research's findings and outputs were plotted and tallied in order to examine and determine the microsleep event.

# CHAPTER 5

# CONCLUSION & RECOMMENDATION

## 5.1 Introduction

In this project, the measurement and analysis of microsleep already achieved and being researched from time to time. The final analysis cannot be stated yet as the experiment need to be conducted in real life where the subject needed to recorded using a real driving situation in order to find the difference of the EEG signal behavior. Next future work will be conducted in next semester.

## 5.2 Conclusion

Finally, the difficulties raised in the problem statement section were satisfactorily resolved throughout this study. In addition, this study met all of the objectives and project scopes outlined in Chapter 1. The EEG Mechanism Interaction to Evaluate Vehicle Driver's Microsleep project was completed successfully, and it was able to detect drowsiness vehicle drivers while driving with a low-cost single-channel EEG headset. For signalling microsleep parameters, the system was able to capture and classify the EEG data into Theta, Alpha, and Beta waves. In addition, the system was able to extract and evaluate the microsleep parameter in Matlab. Finally, the alert system with light, alarm, and steering vibration functions was successfully designed and manufactured utilising the cost-effective microsleep detecting technology. A single-channel EEG headgear, an Arduino microcontroller, and a PC make up the technology developed in this study. It was created with the intention of classifying the collected EEG signal into three frequency waves: Theta, Alpha, and Beta. The Matlab GUI can analyse and distinguish between microsleep and non-microsleep events, and then activate the outputs when a microsleep event occurs.

## 5.3 Recommendation for Future Work

The most important change is to convert the system to an online or real-time microsleep detection system that can warn and notify the driver on the spot anytime a microsleep occurrence is detected. The alarm and steering vibration alert systems will be useless unless the system has an online detection system that can activate the alert system immediately after the driver falls asleep. As a result, an online microsleep detection system can be constructed by collecting a significant amount of EEG data that includes all of the user's probable emotions. This is because the user's emotions might have a substantial impact on the system's outcome.

## 5.4 Impact to Society and Environment

For vehicle drivers, particularly long-distance drivers, the proposed EEG-based microsleep detection technology is critical. Microsleep is particularly harmful for drivers because it causes them to be unconscious and lose focus while driving. In addition, microsleep is one of the leading causes of fatal traffic accidents in Malaysia. As a result, by designing an EEG-based microsleep detection system, the accident rate can be significantly lowered.

# REFERENCES


[1] Ben Dkhil, M., Wali, A., & Alimi, A. M. (2015). Drowsy driver detection by EEG analysis using Fast Fourier Transform. 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA). https://doi.org/10.1109/isda.2015.7489245

[2] Inoue, R., Sugi, T., Matsuda, Y., Goto, S., Nohira, H., & Mase, R. (2019). Recording and Characterization of EEGs by Using Wearable EEG Device. 2019 19th International Conference on Control, Automation and Systems (ICCAS). https://doi.org/10.23919/iccas47443.2019.8971564

[3] Skorucak, J., Hertig-Godeschalk, A., Schreier, D. R., Malafeev, A., Mathis, J., & Achermann, P. (2019). Automatic detection of microsleep episodes with feature-based machine learning. Sleep, 43(1). https://doi.org/10.1093/sleep/zsz225

[4] Wang, C., Guragain, B., Verma, A. K., Archer, L., Majumder, S., Mohamud, A., Flaherty-Woods, E., Shapiro, G., Almashor, M., Lenne, M., Myers, R., Kuo, J., Yang, S., Wilson, N., & Tavakolian, K. (2020). Spectral Analysis of EEG During Microsleep Events Annotated via Driver Monitoring System to Characterize Drowsiness. IEEE Transactions on Aerospace and Electronic Systems, 56(2), 1346–1356. https://doi.org/10.1109/taes.2019.2933960

[5] Swetapadma, A., & Swain, B. R. (2016). A data mining approach for sleep wave and sleep stage classification. 2016 International Conference on Inventive Computation Technologies (ICICT). https://doi.org/10.1109/inventive.2016.7830130

[6] You, S. D., & Wu, Y.-C. (2019). Classification Consistency of Concentrated and Relaxed Mental States with EEG Signals. 2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech). https://doi.org/10.1109/lifetech.2019.8884035

[7] Xiang, H., Zeng, T., & Yang, Y. (2020). A Novel Sleep Stage Classification via Combination of Fast Representation Learning and Semantic-to-Signal Learning. 2020 International Joint Conference on Neural Networks (IJCNN). https://doi.org/10.1109/ijcnn48605.2020.9206994

[8] Moncada, F., Gonzalez, V. M., Alvarez, V., Garcia, B., & Villar, J. R. (2021). A Preliminary Study on Automatic Detection and Filtering of Artifacts from EEG Signals. 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS). https://doi.org/10.1109/cbms52027.2021.00046

[9] Kaveh, R., Doong, J., Zhou, A., Schwendeman, C., Gopalan, K., Burghardt, F., Arias, A. C., Maharbiz, M., & Muller, R. (2019). A Wireless, Multielectrode, User-generic Ear EEG Recording System. 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS). https://doi.org/10.1109/biocas.2019.8918700

[10] Venkatasubramanian, U., Pearson, J. F., Beckert, L., & Jones, R. D. (2019). Characteristic changes in the EEG signals between microsleeps and preceding responsive states. 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). https://doi.org/10.1109/embc.2019.8857718

[11] Shoorangiz, R., Buriro, A. B., Weddell, S. J., & Jones, R. D. (2019). Detection and Prediction of Microsleeps from EEG using Spatio-Temporal Patterns. 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). https://doi.org/10.1109/embc.2019.8857962

[12] Buriro, A. B., Shoorangiz, R., Weddell, S. J., & Jones, R. D. (2018). Predicting Microsleep States Using EEG Inter-Channel Relationships. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(12), 2260–2269. https://doi.org/10.1109/tnsre.2018.2878587

# APPENDIX A

# APPENDIX 1: GANTT CHART

| SEMESTER 1 | | | MONTHS/WEEKS | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | OCTOBER 2021 | | | NOVEMBER 2021 | | | | | DECEMBER 2021 | | | | JANUARY 2022 | | | |
| **Task** | **Start Date** | **End Date** | **W1** | **W2** | **W3** | **W4** | **W5** | **W6** | **B** | **W7** | **W8** | **W9** | **B** | **W10** | **W11** | **W12** | **W13** | **W14** |
| Literature Review | 13/10/2021 | 1/6/2022 | | | | | | | | | | | | | | | | |
| Matlab Training | 13/11/2021 | 1/6/2022 | | | | | | | | | | | | | | | | |
| Workplace set up | 29/11/2021 | 6/12/2021 | | | | | | | | | | | | | | | | |
| EEG Data Recording | 7/12/2021 | 7/1/2022 | | | | | | | | | | | | | | | | |
| Data Acquisition | 7/12/2021 | 7/1/2022 | | | | | | | | | | | | | | | | |
| EEG Datasets Pre-processing | 13/12/2021 | 20/1/2022 | | | | | | | | | | | | | | | | |
| Alpha & Theta Band Analyze | 13/12/2021 | 7/4/2022 | | | | | | | | | | | | | | | | |
| EEG Featrues & Classification | 3/1/2021 | 7/4/2022 | | | | | | | | | | | | | | | | |
| Submit Report PSM 1 | 20/1/2022 | 28/1/2022 | | | | | | | | | | | | | | | | |

| SEMESTER 2 | | | MONTHS/WEEKS | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MARCH 2022 | | | | APRIL 2022 | | | | MAY 2022 | | | | | JUNE 2022 | | |
| **Task** | **Start Date** | **End Date** | **W1** | **W2** | **W3** | **W4** | **W5** | **W6** | **W7** | **W8** | **B** | **W9** | **W10** | **W11** | **W12** | **W13** | **W14** | **R** |
| Literature Review | 7/3/2022 | 1/6/2022 | | | | | | | | | | | | | | | | |
| Matlab Training | 7/3/2022 | 1/6/2022 | | | | | | | | | | | | | | | | |
| Theta, Alpha & Beta Band Analyze | 7/3/2022 | 7/4/2022 | | | | | | | | | | | | | | | | |
| EEG Features & Classification | 7/3/2022 | 7/4/2022 | | | | | | | | | | | | | | | | |
| Conversion of EEG Features to Device Control | 7/4/2022 | 31/5/2022 | | | | | | | | | | | | | | | | |
| GUI Construction | 20/4/2022 | 31/5/2022 | | | | | | | | | | | | | | | | |
| System Testing | 15/5/2022 | 17/6/2022 | | | | | | | | | | | | | | | | |
| Documentation | 7/3/2022 | 20/6/2022 | | | | | | | | | | | | | | | | |
| Submit Report PSM 2 | 18/6/2022 | 26/6/2022 | | | | | | | | | | | | | | | | |

# APPENDIX B
# SAMPLE APPENDIX 2

## MATLAB CODING

```
figure(1)
eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s5_condition_relax.csv");
eeg_raw_data_active=eeg_data_active(:,3);
subplot(2,2,1)
plot(eeg_raw_data_active)
title('Raw data in Time Domain');
xlabel('Time');
ylabel('Amplitude');


%%Filtered EEG Data%%
filtered_eeg_data_active=eeg_raw_data_active(eeg_raw_data_active <=
300 & eeg_raw_data_active >= -600);
%xlswrite('Filtered Data 1.xlsx',filtered_eeg_data_active)
subplot(2,2,2)
plot(filtered_eeg_data_active)
title('Filtered EEG Data');
xlabel('Time');
ylabel('Amplitude');
ylim([-350 350]);

%%FFT Raw EEG%%
[Raw_X, Raw_fft]=my_fft(eeg_raw_data_active);
subplot(2,2,3);
plot(Raw_X, abs(Raw_fft))
title('FFT of Raw EEG in Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');

%%FFT Filtered EEG%%
[Filtered_eeg_X, Filtered_eeg_fft]=my_fft(filtered_eeg_data_active);
subplot(2,2,4);
plot(Filtered_eeg_X, abs(Filtered_eeg_fft));
title('FFT of Filtered EEG in Frequency Domain');
xlabel('Frequncy');
ylabel('Amplitude');
xlim([1 50]);

%%Plotting Alpha % Theta%%
figure(2)
%%Plot Theta Band%%
Theta_time=my_Theta_filter(filtered_eeg_data_active);
subplot(3,2,1)
plot(Theta_time,'r')
title('EEG Theta Band in Time Domain')
xlabel('Time')
ylabel('Amplitude')
ylim([-350 350])
```

```
%%Plot Alpha band%%
Alpha_time=my_Alpha_filter(filtered_eeg_data_active);
subplot(3,2,3)
plot(Alpha_time,'r')
title('EEG Alpha Band in Time Domain');
xlabel('Time');
ylabel('Amplitude');
ylim([-350 350])

%%Plot Beta Band%%
Beta_time=my_Beta_filter(filtered_eeg_data_active);
subplot(3,2,5)
plot(Beta_time,'r')
title('EEG Beta Band in Time Domain')
xlabel('Time')
ylabel('Amplitude')
ylim([-350 350])

%%ALPHA FREQUENCY DOMAIN%%

%%THETA FREQUENCY DOMAIN%%
[Raw_X_Theta, Raw_fft_Theta]=my_fft(Theta_time);
subplot(3,2,2);
plot(Raw_X_Theta, abs(Raw_fft_Theta),'r');
title('FFT of EEG Theta Band in Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');
xlim([0 50])

[Raw_X_Alpha, Raw_fft_Alpha]=my_fft(Alpha_time);
%y=[Raw_X_Alpha, abs(Raw_fft_Alpha)];
%xlswrite('Alpha Data 1.xlsx', y);
subplot(3,2,4);
plot(Raw_X_Alpha, abs(Raw_fft_Alpha),'r')
title('FFT of EEG Alpha Band in Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');
xlim([0 50])

[Raw_X_Beta, Raw_fft_Beta]=my_fft(Beta_time);
%y=[Raw_X_Alpha, abs(Raw_fft_Alpha)];
%xlswrite('Alpha Data 1.xlsx', y);
subplot(3,2,6);
plot(Raw_X_Beta, abs(Raw_fft_Beta),'r')
title('FFT of EEG Beta Band in Frequency Domain');
xlabel('Frequency');
ylabel('Amplitude');
xlim([0 50])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%PSD using FFT Formula%%
%figure(3)
%PS=abs(Raw_fft);
%PS2=PS.^2;
%PS3=PS2/length(Raw_fft);
%xlswrite('PSpectrum Data1.xlsx',PS3);
```

```
%subplot(2,4,1);
%plot(PS3)
%title('Power Spectrum EEG')
%xlabel('Frequency');
%ylabel('Amplitude');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%


%%PSD using PWELCH Method%%
figure(3)

%%PSD using PWELCH Method (THETA)%%
PSD_Pwelch_theta=pwelch(Theta_time);
%%Mean Theta%%
mean_PSD_Pwelch_theta=mean(PSD_Pwelch_theta);
xlswrite('mean_std_relax_s5.xlsx',mean_PSD_Pwelch_theta,'mean_theta');
subplot(3,3,1);
bar(mean_PSD_Pwelch_theta);
title('PSD Theta Mean');
ylabel('Amplitude');
ylim([0 2000]);
%%Standard Dev Theta%%
SD_PSD_Pwelch_theta=std(PSD_Pwelch_theta);
xlswrite('mean_std_relax_s5.xlsx',SD_PSD_Pwelch_theta,'std_theta');
subplot(3,3,4);
bar(SD_PSD_Pwelch_theta);
title('PSD Theta Standard Dev');
ylabel('Amplitude');
ylim([0 3500]);
%%VARIANCE THETA%%%%
VAR_PSD_Pwelch_theta=var(PSD_Pwelch_theta);
subplot(3,3,7);
bar(VAR_PSD_Pwelch_theta);
title('Theta Variance');
ylabel('Amplitude');


%%PSD using PWELCH Method (ALPHA)%%
PSD_Pwelch_alpha=pwelch(Alpha_time);
%%Mean Alpha%%
mean_PSD_Pwelch_alpha=mean(PSD_Pwelch_alpha);
xlswrite('mean_std_relax_s5.xlsx',mean_PSD_Pwelch_alpha,'mean_alpha');
subplot(3,3,2);
bar(mean_PSD_Pwelch_alpha,'r');
title('PSD Alpha Mean');
ylabel('Amplitude')
ylim([0 2000]);
%%Standard Dev Alpha%%
SD_PSD_Pwelch_alpha=std(PSD_Pwelch_alpha);
xlswrite('mean_std_relax_s5.xlsx',SD_PSD_Pwelch_alpha,'std_alpha');
subplot(3,3,5);
bar(SD_PSD_Pwelch_alpha,'r');
title('PSD Alpha Standard Dev');
ylabel('Amplitude')
ylim([0 3500]);
%%VARIANCE ALPHA%%%%
```

```
VAR_PSD_Pwelch_alpha=var(PSD_Pwelch_alpha);
subplot(3,3,8);
bar(VAR_PSD_Pwelch_alpha,'r');
title('Alpha Variance');
ylabel('Amplitude');

%%PSD using PWELCH Method (BETA)%%
PSD_Pwelch_beta=pwelch(Beta_time);
%%Mean Beta%%
mean_PSD_Pwelch_beta=mean(PSD_Pwelch_beta);
xlswrite('mean_std_relax_s5.xlsx',mean_PSD_Pwelch_beta,'mean_beta');
subplot(3,3,3);
bar(mean_PSD_Pwelch_beta,'g');
title('PSD Beta Mean');
ylabel('Amplitude')
ylim([0 2000]);
%%Standard Dev Alpha%%
SD_PSD_Pwelch_beta=std(PSD_Pwelch_beta);
xlswrite('mean_std_relax_s5.xlsx',SD_PSD_Pwelch_beta,'std_beta');
subplot(3,3,6);
bar(SD_PSD_Pwelch_beta,'g');
title('PSD Beta Standard Dev');
ylabel('Amplitude')
ylim([0 3500]);
%%VARIANCE BETA%%%%
VAR_PSD_Pwelch_beta=var(PSD_Pwelch_beta);
subplot(3,3,9);
bar(VAR_PSD_Pwelch_beta,'g');
title('Beta Variance');
ylabel('Amplitude');



%%HISTOGRAM MEAN ALPHA & THETA%%
figure(4)
x=[mean_PSD_Pwelch_theta, mean_PSD_Pwelch_alpha,
mean_PSD_Pwelch_beta];
bar(x)
title('Mean Histogram of Theta, Alpha & Beta Band');
ylabel('Amplitude');
set(gca,'XTickLabel',{'Theta','Alpha','Beta'})


%%HISTOGRAM STANDARD DEV ALPHA & THETA%%
figure(5)
y=[SD_PSD_Pwelch_theta, SD_PSD_Pwelch_alpha, SD_PSD_Pwelch_beta];
bar(y)
title('STD Histogram of Theta, Alpha & Beta Band');
ylabel('Amplitude');
set(gca,'XTickLabel',{'Theta','Alpha','Beta'});


%%HISTOGRAM VARIANCE ALPHA & THETA%%
figure(6)
z=[VAR_PSD_Pwelch_theta, VAR_PSD_Pwelch_alpha, VAR_PSD_Pwelch_beta];
bar(z)
title('VAR Histogram of Theta, Alpha & Beta Band');
```

```
ylabel('Amplitude');
set(gca,'XTickLabel',{'Theta','Alpha','Beta'});
```

### APP DESIGNER MATLAB CODING

```
classdef fyp_v1 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                   matlab.ui.Figure
        Subject1Button             matlab.ui.control.Button
        UIAxes                     matlab.ui.control.UIAxes
        UIAxes_2                   matlab.ui.control.UIAxes
        UIAxes_3                   matlab.ui.control.UIAxes
        UIAxes_4                   matlab.ui.control.UIAxes
        UIAxes_5                   matlab.ui.control.UIAxes
        UIAxes_6                   matlab.ui.control.UIAxes
        Image                      matlab.ui.control.Image
        EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel
matlab.ui.control.Label
        DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label
matlab.ui.control.Label
        SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel
matlab.ui.control.Label
        DetectLamp                 matlab.ui.control.Lamp
        NoDetectLamp               matlab.ui.control.Lamp
        MICROSLEEPAPPEARANCELabel  matlab.ui.control.Label
        Subject2Button             matlab.ui.control.Button
        Subject3Button             matlab.ui.control.Button
        RELAXEDSTATELabel          matlab.ui.control.Label
        Subject4Button             matlab.ui.control.Button
        Subject5Button             matlab.ui.control.Button
        UIAxes_7                   matlab.ui.control.UIAxes
        Subject1Button_2           matlab.ui.control.Button
        Subject2Button_2           matlab.ui.control.Button
        Subject3Button_2           matlab.ui.control.Button
        Subject4Button_2           matlab.ui.control.Button
        Subject5Button_2           matlab.ui.control.Button
        DRIVINGSTATELabel          matlab.ui.control.Label
        UNDETECTEDLabel            matlab.ui.control.Label
        DETECTEDLabel              matlab.ui.control.Label
        SubjectLabel               matlab.ui.control.Label
        SubjectEditField           matlab.ui.control.EditField
        ConditionLabel             matlab.ui.control.Label
        ConditionEditField         matlab.ui.control.EditField

EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2
matlab.ui.control.Label
        Image2                     matlab.ui.control.Image
        UIAxes_8                   matlab.ui.control.UIAxes
        Image3                     matlab.ui.control.Image
    end
```

```
    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: Subject1Button
        function Subject1ButtonPushed(app, event)

            app.SubjectEditField.Value = '1';
            app.ConditionEditField.Value = 'RELAXED';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %greY
            app.DetectLamp.Color = [1 0 0]; %RED

            a1_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s1_condition_relax.csv");
            a1_eeg_raw_data_active=a1_eeg_data_active(:,3);
            plot(app.UIAxes,[a1_eeg_raw_data_active]);


a1_filtered_eeg_data_active=a1_eeg_raw_data_active(a1_eeg_raw_data_act
ive <= 300 & a1_eeg_raw_data_active >= -600);
            [a1_Raw_X, a1_Raw_fft]=my_fft(a1_eeg_raw_data_active);
            [a1_Filtered_eeg_X,
a1_Filtered_eeg_fft]=my_fft(a1_filtered_eeg_data_active);
            plot(app.UIAxes_7, a1_Filtered_eeg_X,
abs(a1_Filtered_eeg_fft));


a1_Theta_time=my_Theta_filter(a1_filtered_eeg_data_active);

a1_Alpha_time=my_Alpha_filter(a1_filtered_eeg_data_active);
            a1_Beta_time=my_Beta_filter(a1_filtered_eeg_data_active);

            [a1_Raw_X_Theta, a1_Raw_fft_Theta]=my_fft(a1_Theta_time);
            plot(app.UIAxes_2,a1_Raw_X_Theta, abs(a1_Raw_fft_Theta));

            [a1_Raw_X_Alpha, a1_Raw_fft_Alpha]=my_fft(a1_Alpha_time);
            plot(app.UIAxes_3,a1_Raw_X_Alpha, abs(a1_Raw_fft_Alpha));

            [a1_Raw_X_Beta, a1_Raw_fft_Beta]=my_fft(a1_Beta_time);
            plot(app.UIAxes_4,a1_Raw_X_Beta, abs(a1_Raw_fft_Beta));

            a1_PSD_Pwelch_theta=pwelch(a1_Theta_time);
            a1_mean_PSD_Pwelch_theta=mean(a1_PSD_Pwelch_theta);
            a1_SD_PSD_Pwelch_theta=std(a1_PSD_Pwelch_theta);
            a1_var_PSD_Pwelch_theta=var(a1_PSD_Pwelch_theta);

            a1_PSD_Pwelch_alpha=pwelch(a1_Alpha_time);
            a1_mean_PSD_Pwelch_alpha=mean(a1_PSD_Pwelch_alpha);
            a1_SD_PSD_Pwelch_alpha=std(a1_PSD_Pwelch_alpha);
            a1_var_PSD_Pwelch_alpha=var(a1_PSD_Pwelch_alpha);

            a1_PSD_Pwelch_beta=pwelch(a1_Beta_time);
            a1_mean_PSD_Pwelch_beta=mean(a1_PSD_Pwelch_beta);
            a1_SD_PSD_Pwelch_beta=std(a1_PSD_Pwelch_beta);
            a1_var_PSD_Pwelch_beta=var(a1_PSD_Pwelch_beta);

            a1_x=[a1_mean_PSD_Pwelch_theta; a1_mean_PSD_Pwelch_alpha;
a1_mean_PSD_Pwelch_beta];
```

```
            bar(app.UIAxes_5,a1_x);

            a1_y=[a1_SD_PSD_Pwelch_theta, a1_SD_PSD_Pwelch_alpha,
a1_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,a1_y);

            a1_z=[a1_var_PSD_Pwelch_theta, a1_var_PSD_Pwelch_alpha,
a1_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,a1_z);



        end

        % Button pushed function: Subject2Button
        function Subject2ButtonPushed(app, event)
            app.SubjectEditField.Value = '2';
            app.ConditionEditField.Value = 'RELAXED';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %green
            app.DetectLamp.Color = [1 0 0]; %red

            a2_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s2_condition_relax.csv");
            a2_eeg_raw_data_active=a2_eeg_data_active(:,3);
            plot(app.UIAxes,[a2_eeg_raw_data_active]);


a2_filtered_eeg_data_active=a2_eeg_raw_data_active(a2_eeg_raw_data_act
ive <= 300 & a2_eeg_raw_data_active >= -600);
            [a2_Raw_X, a2_Raw_fft]=my_fft(a2_eeg_raw_data_active);
            [a2_Filtered_eeg_X,
a2_Filtered_eeg_fft]=my_fft(a2_filtered_eeg_data_active);
            plot(app.UIAxes_7, a2_Filtered_eeg_X,
abs(a2_Filtered_eeg_fft));


a2_Theta_time=my_Theta_filter(a2_filtered_eeg_data_active);

a2_Alpha_time=my_Alpha_filter(a2_filtered_eeg_data_active);
            a2_Beta_time=my_Beta_filter(a2_filtered_eeg_data_active);

            [a2_Raw_X_Theta, a2_Raw_fft_Theta]=my_fft(a2_Theta_time);
            plot(app.UIAxes_2,a2_Raw_X_Theta, abs(a2_Raw_fft_Theta));

            [a2_Raw_X_Alpha, a2_Raw_fft_Alpha]=my_fft(a2_Alpha_time);
            plot(app.UIAxes_3,a2_Raw_X_Alpha, abs(a2_Raw_fft_Alpha));

            [a2_Raw_X_Beta, a2_Raw_fft_Beta]=my_fft(a2_Beta_time);
            plot(app.UIAxes_4,a2_Raw_X_Beta, abs(a2_Raw_fft_Beta));

            a2_PSD_Pwelch_theta=pwelch(a2_Theta_time);
            a2_mean_PSD_Pwelch_theta=mean(a2_PSD_Pwelch_theta);
            a2_SD_PSD_Pwelch_theta=std(a2_PSD_Pwelch_theta);
            a2_var_PSD_Pwelch_theta=var(a2_PSD_Pwelch_theta);

            a2_PSD_Pwelch_alpha=pwelch(a2_Alpha_time);
            a2_mean_PSD_Pwelch_alpha=mean(a2_PSD_Pwelch_alpha);
```

```
            a2_SD_PSD_Pwelch_alpha=std(a2_PSD_Pwelch_alpha);
            a2_var_PSD_Pwelch_alpha=var(a2_PSD_Pwelch_alpha);

            a2_PSD_Pwelch_beta=pwelch(a2_Beta_time);
            a2_mean_PSD_Pwelch_beta=mean(a2_PSD_Pwelch_beta);
            a2_SD_PSD_Pwelch_beta=std(a2_PSD_Pwelch_beta);
            a2_var_PSD_Pwelch_beta=var(a2_PSD_Pwelch_beta);

            a2_x=[a2_mean_PSD_Pwelch_theta; a2_mean_PSD_Pwelch_alpha;
a2_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,a2_x);

            a2_y=[a2_SD_PSD_Pwelch_theta, a2_SD_PSD_Pwelch_alpha,
a2_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,a2_y);

            a2_z=[a2_var_PSD_Pwelch_theta, a2_var_PSD_Pwelch_alpha,
a2_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,a2_z);
        end

        % Button pushed function: Subject3Button
        function Subject3ButtonPushed(app, event)
            app.SubjectEditField.Value = '3';
            app.ConditionEditField.Value = 'RELAXED';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %grey
            app.DetectLamp.Color = [1 0 0]; %RED

            a3_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s3_condition_relax.csv");
            a3_eeg_raw_data_active=a3_eeg_data_active(:,3);
            plot(app.UIAxes,[a3_eeg_raw_data_active]);


a3_filtered_eeg_data_active=a3_eeg_raw_data_active(a3_eeg_raw_data_act
ive <= 300 & a3_eeg_raw_data_active >= -600);
            [a3_Raw_X, a3_Raw_fft]=my_fft(a3_eeg_raw_data_active);
            [a3_Filtered_eeg_X,
a3_Filtered_eeg_fft]=my_fft(a3_filtered_eeg_data_active);
            plot(app.UIAxes_7, a3_Filtered_eeg_X,
abs(a3_Filtered_eeg_fft));


a3_Theta_time=my_Theta_filter(a3_filtered_eeg_data_active);

a3_Alpha_time=my_Alpha_filter(a3_filtered_eeg_data_active);
            a3_Beta_time=my_Beta_filter(a3_filtered_eeg_data_active);

            [a3_Raw_X_Theta, a3_Raw_fft_Theta]=my_fft(a3_Theta_time);
            plot(app.UIAxes_2,a3_Raw_X_Theta, abs(a3_Raw_fft_Theta));

            [a3_Raw_X_Alpha, a3_Raw_fft_Alpha]=my_fft(a3_Alpha_time);
            plot(app.UIAxes_3,a3_Raw_X_Alpha, abs(a3_Raw_fft_Alpha));

            [a3_Raw_X_Beta, a3_Raw_fft_Beta]=my_fft(a3_Beta_time);
            plot(app.UIAxes_4,a3_Raw_X_Beta, abs(a3_Raw_fft_Beta));
```

```
            a3_PSD_Pwelch_theta=pwelch(a3_Theta_time);
            a3_mean_PSD_Pwelch_theta=mean(a3_PSD_Pwelch_theta);
            a3_SD_PSD_Pwelch_theta=std(a3_PSD_Pwelch_theta);
            a3_var_PSD_Pwelch_theta=var(a3_PSD_Pwelch_theta);

            a3_PSD_Pwelch_alpha=pwelch(a3_Alpha_time);
            a3_mean_PSD_Pwelch_alpha=mean(a3_PSD_Pwelch_alpha);
            a3_SD_PSD_Pwelch_alpha=std(a3_PSD_Pwelch_alpha);
            a3_var_PSD_Pwelch_alpha=var(a3_PSD_Pwelch_alpha);

            a3_PSD_Pwelch_beta=pwelch(a3_Beta_time);
            a3_mean_PSD_Pwelch_beta=mean(a3_PSD_Pwelch_beta);
            a3_SD_PSD_Pwelch_beta=std(a3_PSD_Pwelch_beta);
            a3_var_PSD_Pwelch_beta=var(a3_PSD_Pwelch_beta);

            a3_x=[a3_mean_PSD_Pwelch_theta; a3_mean_PSD_Pwelch_alpha;
a3_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,a3_x);

            a3_y=[a3_SD_PSD_Pwelch_theta, a3_SD_PSD_Pwelch_alpha,
a3_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,a3_y);

            a3_z=[a3_var_PSD_Pwelch_theta, a3_var_PSD_Pwelch_alpha,
a3_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,a3_z);
        end

        % Button pushed function: Subject4Button
        function Subject4ButtonPushed(app, event)
            app.SubjectEditField.Value = '4';
            app.ConditionEditField.Value = 'RELAXED';
            app.NoDetectLamp.Color = [0 1 0]; %GREEN
            app.DetectLamp.Color = [0.9 0.9 0.9]; %grey

            a4_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s4_condition_relax.csv");
            a4_eeg_raw_data_active=a4_eeg_data_active(:,3);
            plot(app.UIAxes,[a4_eeg_raw_data_active]);


a4_filtered_eeg_data_active=a4_eeg_raw_data_active(a4_eeg_raw_data_act
ive <= 300 & a4_eeg_raw_data_active >= -600);
            [a4_Raw_X, a4_Raw_fft]=my_fft(a4_eeg_raw_data_active);
            [a4_Filtered_eeg_X,
a4_Filtered_eeg_fft]=my_fft(a4_filtered_eeg_data_active);
            plot(app.UIAxes_7, a4_Filtered_eeg_X,
abs(a4_Filtered_eeg_fft));


a4_Theta_time=my_Theta_filter(a4_filtered_eeg_data_active);

a4_Alpha_time=my_Alpha_filter(a4_filtered_eeg_data_active);
            a4_Beta_time=my_Beta_filter(a4_filtered_eeg_data_active);

            [a4_Raw_X_Theta, a4_Raw_fft_Theta]=my_fft(a4_Theta_time);
            plot(app.UIAxes_2,a4_Raw_X_Theta, abs(a4_Raw_fft_Theta));
```

```
                [a4_Raw_X_Alpha, a4_Raw_fft_Alpha]=my_fft(a4_Alpha_time);
                plot(app.UIAxes_3,a4_Raw_X_Alpha, abs(a4_Raw_fft_Alpha));

                [a4_Raw_X_Beta, a4_Raw_fft_Beta]=my_fft(a4_Beta_time);
                plot(app.UIAxes_4,a4_Raw_X_Beta, abs(a4_Raw_fft_Beta));

                a4_PSD_Pwelch_theta=pwelch(a4_Theta_time);
                a4_mean_PSD_Pwelch_theta=mean(a4_PSD_Pwelch_theta);
                a4_SD_PSD_Pwelch_theta=std(a4_PSD_Pwelch_theta);
                a4_var_PSD_Pwelch_theta=var(a4_PSD_Pwelch_theta);

                a4_PSD_Pwelch_alpha=pwelch(a4_Alpha_time);
                a4_mean_PSD_Pwelch_alpha=mean(a4_PSD_Pwelch_alpha);
                a4_SD_PSD_Pwelch_alpha=std(a4_PSD_Pwelch_alpha);
                a4_var_PSD_Pwelch_alpha=var(a4_PSD_Pwelch_alpha);

                a4_PSD_Pwelch_beta=pwelch(a4_Beta_time);
                a4_mean_PSD_Pwelch_beta=mean(a4_PSD_Pwelch_beta);
                a4_SD_PSD_Pwelch_beta=std(a4_PSD_Pwelch_beta);
                a4_var_PSD_Pwelch_beta=var(a4_PSD_Pwelch_beta);

                a4_x=[a4_mean_PSD_Pwelch_theta; a4_mean_PSD_Pwelch_alpha;
a4_mean_PSD_Pwelch_beta];
                bar(app.UIAxes_5,a4_x);

                a4_y=[a4_SD_PSD_Pwelch_theta, a4_SD_PSD_Pwelch_alpha,
a4_SD_PSD_Pwelch_beta];
                bar(app.UIAxes_6,a4_y);

                a4_z=[a4_var_PSD_Pwelch_theta, a4_var_PSD_Pwelch_alpha,
a4_var_PSD_Pwelch_beta];
                bar(app.UIAxes_8,a4_z);
            end

            % Button pushed function: Subject5Button
            function Subject5ButtonPushed(app, event)
                app.SubjectEditField.Value = '5';
                app.ConditionEditField.Value = 'RELAXED';
                app.NoDetectLamp.Color = [0.9 0.9 0.9]; %grey
                app.DetectLamp.Color = [1 0 0]; %RED

                a5_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s5_condition_relax.csv");
                a5_eeg_raw_data_active=a5_eeg_data_active(:,3);
                plot(app.UIAxes,[a5_eeg_raw_data_active]);


a5_filtered_eeg_data_active=a5_eeg_raw_data_active(a5_eeg_raw_data_act
ive <= 300 & a5_eeg_raw_data_active >= -600);
                [a5_Raw_X, a5_Raw_fft]=my_fft(a5_eeg_raw_data_active);
                [a5_Filtered_eeg_X,
a5_Filtered_eeg_fft]=my_fft(a5_filtered_eeg_data_active);
                plot(app.UIAxes_7, a5_Filtered_eeg_X,
abs(a5_Filtered_eeg_fft));
```

```
a5_Theta_time=my_Theta_filter(a5_filtered_eeg_data_active);

a5_Alpha_time=my_Alpha_filter(a5_filtered_eeg_data_active);
            a5_Beta_time=my_Beta_filter(a5_filtered_eeg_data_active);

            [a5_Raw_X_Theta, a5_Raw_fft_Theta]=my_fft(a5_Theta_time);
            plot(app.UIAxes_2,a5_Raw_X_Theta, abs(a5_Raw_fft_Theta));

            [a5_Raw_X_Alpha, a5_Raw_fft_Alpha]=my_fft(a5_Alpha_time);
            plot(app.UIAxes_3,a5_Raw_X_Alpha, abs(a5_Raw_fft_Alpha));

            [a5_Raw_X_Beta, a5_Raw_fft_Beta]=my_fft(a5_Beta_time);
            plot(app.UIAxes_4,a5_Raw_X_Beta, abs(a5_Raw_fft_Beta));

            a5_PSD_Pwelch_theta=pwelch(a5_Theta_time);
            a5_mean_PSD_Pwelch_theta=mean(a5_PSD_Pwelch_theta);
            a5_SD_PSD_Pwelch_theta=std(a5_PSD_Pwelch_theta);
            a5_var_PSD_Pwelch_theta=var(a5_PSD_Pwelch_theta);

            a5_PSD_Pwelch_alpha=pwelch(a5_Alpha_time);
            a5_mean_PSD_Pwelch_alpha=mean(a5_PSD_Pwelch_alpha);
            a5_SD_PSD_Pwelch_alpha=std(a5_PSD_Pwelch_alpha);
            a5_var_PSD_Pwelch_alpha=var(a5_PSD_Pwelch_alpha);

            a5_PSD_Pwelch_beta=pwelch(a5_Beta_time);
            a5_mean_PSD_Pwelch_beta=mean(a5_PSD_Pwelch_beta);
            a5_SD_PSD_Pwelch_beta=std(a5_PSD_Pwelch_beta);
            a5_var_PSD_Pwelch_beta=var(a5_PSD_Pwelch_beta);

            a5_x=[a5_mean_PSD_Pwelch_theta; a5_mean_PSD_Pwelch_alpha;
a5_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,a5_x);

            a5_y=[a5_SD_PSD_Pwelch_theta, a5_SD_PSD_Pwelch_alpha,
a5_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,a5_y);

            a5_z=[a5_var_PSD_Pwelch_theta, a5_var_PSD_Pwelch_alpha,
a5_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,a5_z);
        end

        % Button pushed function: Subject1Button_2
        function Subject1Button_2Pushed(app, event)
            app.SubjectEditField.Value = '1';
            app.ConditionEditField.Value = 'DRIVING';
            app.NoDetectLamp.Color = [0 1 0]; %green
            app.DetectLamp.Color = [0.9 0.9 0.9]; %grey

            b1_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s1_condition_active.csv");
            b1_eeg_raw_data_active=b1_eeg_data_active(:,3);
            plot(app.UIAxes,[b1_eeg_raw_data_active]);
```

```
b1_filtered_eeg_data_active=b1_eeg_raw_data_active(b1_eeg_raw_data_act
ive <= 300 & b1_eeg_raw_data_active >= -600);
            [b1_Raw_X, b1_Raw_fft]=my_fft(b1_eeg_raw_data_active);
            [b1_Filtered_eeg_X,
b1_Filtered_eeg_fft]=my_fft(b1_filtered_eeg_data_active);
            plot(app.UIAxes_7, b1_Filtered_eeg_X,
abs(b1_Filtered_eeg_fft));


b1_Theta_time=my_Theta_filter(b1_filtered_eeg_data_active);

b1_Alpha_time=my_Alpha_filter(b1_filtered_eeg_data_active);
            b1_Beta_time=my_Beta_filter(b1_filtered_eeg_data_active);

            [b1_Raw_X_Theta, b1_Raw_fft_Theta]=my_fft(b1_Theta_time);
            plot(app.UIAxes_2,b1_Raw_X_Theta, abs(b1_Raw_fft_Theta));

            [b1_Raw_X_Alpha, b1_Raw_fft_Alpha]=my_fft(b1_Alpha_time);
            plot(app.UIAxes_3,b1_Raw_X_Alpha, abs(b1_Raw_fft_Alpha));

            [b1_Raw_X_Beta, b1_Raw_fft_Beta]=my_fft(b1_Beta_time);
            plot(app.UIAxes_4,b1_Raw_X_Beta, abs(b1_Raw_fft_Beta));

            b1_PSD_Pwelch_theta=pwelch(b1_Theta_time);
            b1_mean_PSD_Pwelch_theta=mean(b1_PSD_Pwelch_theta);
            b1_SD_PSD_Pwelch_theta=std(b1_PSD_Pwelch_theta);
            b1_var_PSD_Pwelch_theta=var(b1_PSD_Pwelch_theta);

            b1_PSD_Pwelch_alpha=pwelch(b1_Alpha_time);
            b1_mean_PSD_Pwelch_alpha=mean(b1_PSD_Pwelch_alpha);
            b1_SD_PSD_Pwelch_alpha=std(b1_PSD_Pwelch_alpha);
            b1_var_PSD_Pwelch_alpha=var(b1_PSD_Pwelch_alpha);

            b1_PSD_Pwelch_beta=pwelch(b1_Beta_time);
            b1_mean_PSD_Pwelch_beta=mean(b1_PSD_Pwelch_beta);
            b1_SD_PSD_Pwelch_beta=std(b1_PSD_Pwelch_beta);
            b1_var_PSD_Pwelch_beta=var(b1_PSD_Pwelch_beta);

            b1_x=[b1_mean_PSD_Pwelch_theta; b1_mean_PSD_Pwelch_alpha;
b1_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,b1_x);

            b1_y=[b1_SD_PSD_Pwelch_theta, b1_SD_PSD_Pwelch_alpha,
b1_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,b1_y);

            b1_z=[b1_var_PSD_Pwelch_theta, b1_var_PSD_Pwelch_alpha,
b1_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,b1_z);
        end

        % Button pushed function: Subject2Button_2
        function Subject2Button_2Pushed(app, event)
             app.SubjectEditField.Value = '2';
            app.ConditionEditField.Value = 'DRIVING';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %green
```

```
            app.DetectLamp.Color = [1 0 0]; %RED

            b2_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER 
6\FYP1\MATLAB\Version 3\s2_condition_active.csv");
            b2_eeg_raw_data_active=b2_eeg_data_active(:,3);
            plot(app.UIAxes,[b2_eeg_raw_data_active]);


b2_filtered_eeg_data_active=b2_eeg_raw_data_active(b2_eeg_raw_data_act
ive <= 300 & b2_eeg_raw_data_active >= -600);
            [b2_Raw_X, b2_Raw_fft]=my_fft(b2_eeg_raw_data_active);
            [b2_Filtered_eeg_X, 
b2_Filtered_eeg_fft]=my_fft(b2_filtered_eeg_data_active);
            plot(app.UIAxes_7, b2_Filtered_eeg_X, 
abs(b2_Filtered_eeg_fft));


b2_Theta_time=my_Theta_filter(b2_filtered_eeg_data_active);

b2_Alpha_time=my_Alpha_filter(b2_filtered_eeg_data_active);
            b2_Beta_time=my_Beta_filter(b2_filtered_eeg_data_active);

            [b2_Raw_X_Theta, b2_Raw_fft_Theta]=my_fft(b2_Theta_time);
            plot(app.UIAxes_2,b2_Raw_X_Theta, abs(b2_Raw_fft_Theta));

            [b2_Raw_X_Alpha, b2_Raw_fft_Alpha]=my_fft(b2_Alpha_time);
            plot(app.UIAxes_3,b2_Raw_X_Alpha, abs(b2_Raw_fft_Alpha));

            [b2_Raw_X_Beta, b2_Raw_fft_Beta]=my_fft(b2_Beta_time);
            plot(app.UIAxes_4,b2_Raw_X_Beta, abs(b2_Raw_fft_Beta));

            b2_PSD_Pwelch_theta=pwelch(b2_Theta_time);
            b2_mean_PSD_Pwelch_theta=mean(b2_PSD_Pwelch_theta);
            b2_SD_PSD_Pwelch_theta=std(b2_PSD_Pwelch_theta);
            b2_var_PSD_Pwelch_theta=var(b2_PSD_Pwelch_theta);

            b2_PSD_Pwelch_alpha=pwelch(b2_Alpha_time);
            b2_mean_PSD_Pwelch_alpha=mean(b2_PSD_Pwelch_alpha);
            b2_SD_PSD_Pwelch_alpha=std(b2_PSD_Pwelch_alpha);
            b2_var_PSD_Pwelch_alpha=var(b2_PSD_Pwelch_alpha);

            b2_PSD_Pwelch_beta=pwelch(b2_Beta_time);
            b2_mean_PSD_Pwelch_beta=mean(b2_PSD_Pwelch_beta);
            b2_SD_PSD_Pwelch_beta=std(b2_PSD_Pwelch_beta);
            b2_var_PSD_Pwelch_beta=var(b2_PSD_Pwelch_beta);

            b2_x=[b2_mean_PSD_Pwelch_theta; b2_mean_PSD_Pwelch_alpha; 
b2_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,b2_x);

            b2_y=[b2_SD_PSD_Pwelch_theta, b2_SD_PSD_Pwelch_alpha, 
b2_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,b2_y);

            b2_z=[b2_var_PSD_Pwelch_theta, b2_var_PSD_Pwelch_alpha, 
b2_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,b2_z);
```

```
        end

        % Button pushed function: Subject3Button_2
        function Subject3Button_2Pushed(app, event)
             app.SubjectEditField.Value = '3';
            app.ConditionEditField.Value = 'DRIVING';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %green
            app.DetectLamp.Color = [1 0 0]; %grey

            b3_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s3_condition_active.csv");
            b3_eeg_raw_data_active=b3_eeg_data_active(:,3);
            plot(app.UIAxes,[b3_eeg_raw_data_active]);


b3_filtered_eeg_data_active=b3_eeg_raw_data_active(b3_eeg_raw_data_act
ive <= 300 & b3_eeg_raw_data_active >= -600);
            [b3_Raw_X, b3_Raw_fft]=my_fft(b3_eeg_raw_data_active);
            [b3_Filtered_eeg_X,
b3_Filtered_eeg_fft]=my_fft(b3_filtered_eeg_data_active);
            plot(app.UIAxes_7, b3_Filtered_eeg_X,
abs(b3_Filtered_eeg_fft));


b3_Theta_time=my_Theta_filter(b3_filtered_eeg_data_active);

b3_Alpha_time=my_Alpha_filter(b3_filtered_eeg_data_active);
            b3_Beta_time=my_Beta_filter(b3_filtered_eeg_data_active);

            [b3_Raw_X_Theta, b3_Raw_fft_Theta]=my_fft(b3_Theta_time);
            plot(app.UIAxes_2,b3_Raw_X_Theta, abs(b3_Raw_fft_Theta));

            [b3_Raw_X_Alpha, b3_Raw_fft_Alpha]=my_fft(b3_Alpha_time);
            plot(app.UIAxes_3,b3_Raw_X_Alpha, abs(b3_Raw_fft_Alpha));

            [b3_Raw_X_Beta, b3_Raw_fft_Beta]=my_fft(b3_Beta_time);
            plot(app.UIAxes_4,b3_Raw_X_Beta, abs(b3_Raw_fft_Beta));

            b3_PSD_Pwelch_theta=pwelch(b3_Theta_time);
            b3_mean_PSD_Pwelch_theta=mean(b3_PSD_Pwelch_theta);
            b3_SD_PSD_Pwelch_theta=std(b3_PSD_Pwelch_theta);
            b3_var_PSD_Pwelch_theta=var(b3_PSD_Pwelch_theta);

            b3_PSD_Pwelch_alpha=pwelch(b3_Alpha_time);
            b3_mean_PSD_Pwelch_alpha=mean(b3_PSD_Pwelch_alpha);
            b3_SD_PSD_Pwelch_alpha=std(b3_PSD_Pwelch_alpha);
            b3_var_PSD_Pwelch_alpha=var(b3_PSD_Pwelch_alpha);

            b3_PSD_Pwelch_beta=pwelch(b3_Beta_time);
            b3_mean_PSD_Pwelch_beta=mean(b3_PSD_Pwelch_beta);
            b3_SD_PSD_Pwelch_beta=std(b3_PSD_Pwelch_beta);
            b3_var_PSD_Pwelch_beta=var(b3_PSD_Pwelch_beta);

            b3_x=[b3_mean_PSD_Pwelch_theta; b3_mean_PSD_Pwelch_alpha;
b3_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,b3_x);
```

```
            b3_y=[b3_SD_PSD_Pwelch_theta, b3_SD_PSD_Pwelch_alpha,
b3_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,b3_y);

            b3_z=[b3_var_PSD_Pwelch_theta, b3_var_PSD_Pwelch_alpha,
b3_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,b3_z);
        end

        % Button pushed function: Subject4Button_2
        function Subject4Button_2Pushed(app, event)
             app.SubjectEditField.Value = '4';
            app.ConditionEditField.Value = 'DRIVING';
            app.NoDetectLamp.Color = [0 1 0]; %green
            app.DetectLamp.Color = [0.9 0.9 0.9]; %grey

            b4_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s4_condition_active.csv");
            b4_eeg_raw_data_active=b4_eeg_data_active(:,3);
            plot(app.UIAxes,[b4_eeg_raw_data_active]);


b4_filtered_eeg_data_active=b4_eeg_raw_data_active(b4_eeg_raw_data_act
ive <= 300 & b4_eeg_raw_data_active >= -600);
            [b4_Raw_X, b4_Raw_fft]=my_fft(b4_eeg_raw_data_active);
            [b4_Filtered_eeg_X,
b4_Filtered_eeg_fft]=my_fft(b4_filtered_eeg_data_active);
            plot(app.UIAxes_7, b4_Filtered_eeg_X,
abs(b4_Filtered_eeg_fft));


b4_Theta_time=my_Theta_filter(b4_filtered_eeg_data_active);

b4_Alpha_time=my_Alpha_filter(b4_filtered_eeg_data_active);
            b4_Beta_time=my_Beta_filter(b4_filtered_eeg_data_active);

            [b4_Raw_X_Theta, b4_Raw_fft_Theta]=my_fft(b4_Theta_time);
            plot(app.UIAxes_2,b4_Raw_X_Theta, abs(b4_Raw_fft_Theta));

            [b4_Raw_X_Alpha, b4_Raw_fft_Alpha]=my_fft(b4_Alpha_time);
            plot(app.UIAxes_3,b4_Raw_X_Alpha, abs(b4_Raw_fft_Alpha));

            [b4_Raw_X_Beta, b4_Raw_fft_Beta]=my_fft(b4_Beta_time);
            plot(app.UIAxes_4,b4_Raw_X_Beta, abs(b4_Raw_fft_Beta));

            b4_PSD_Pwelch_theta=pwelch(b4_Theta_time);
            b4_mean_PSD_Pwelch_theta=mean(b4_PSD_Pwelch_theta);
            b4_SD_PSD_Pwelch_theta=std(b4_PSD_Pwelch_theta);
            b4_var_PSD_Pwelch_theta=var(b4_PSD_Pwelch_theta);

            b4_PSD_Pwelch_alpha=pwelch(b4_Alpha_time);
            b4_mean_PSD_Pwelch_alpha=mean(b4_PSD_Pwelch_alpha);
            b4_SD_PSD_Pwelch_alpha=std(b4_PSD_Pwelch_alpha);
            b4_var_PSD_Pwelch_alpha=var(b4_PSD_Pwelch_alpha);

            b4_PSD_Pwelch_beta=pwelch(b4_Beta_time);
            b4_mean_PSD_Pwelch_beta=mean(b4_PSD_Pwelch_beta);
```

```
            b4_SD_PSD_Pwelch_beta=std(b4_PSD_Pwelch_beta);
            b4_var_PSD_Pwelch_beta=var(b4_PSD_Pwelch_beta);

            b4_x=[b4_mean_PSD_Pwelch_theta; b4_mean_PSD_Pwelch_alpha;
b4_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,b4_x);

            b4_y=[b4_SD_PSD_Pwelch_theta, b4_SD_PSD_Pwelch_alpha,
b4_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,b4_y);

            b4_z=[b4_var_PSD_Pwelch_theta, b4_var_PSD_Pwelch_alpha,
b4_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,b4_z);
        end

        % Button pushed function: Subject5Button_2
        function Subject5Button_2Pushed(app, event)
             app.SubjectEditField.Value = '5';
            app.ConditionEditField.Value = 'DRIVING';
            app.NoDetectLamp.Color = [0.9 0.9 0.9]; %green
            app.DetectLamp.Color = [1 0 0]; %grey

            b5_eeg_data_active=xlsread("D:\6_UMP\DEGREE\SEMESTER
6\FYP1\MATLAB\Version 3\s5_condition_active.csv");
            b5_eeg_raw_data_active=b5_eeg_data_active(:,3);
            plot(app.UIAxes,[b5_eeg_raw_data_active]);


b5_filtered_eeg_data_active=b5_eeg_raw_data_active(b5_eeg_raw_data_act
ive <= 300 & b5_eeg_raw_data_active >= -600);
            [b5_Raw_X, b5_Raw_fft]=my_fft(b5_eeg_raw_data_active);
            [b5_Filtered_eeg_X,
b5_Filtered_eeg_fft]=my_fft(b5_filtered_eeg_data_active);
            plot(app.UIAxes_7, b5_Filtered_eeg_X,
abs(b5_Filtered_eeg_fft));


b5_Theta_time=my_Theta_filter(b5_filtered_eeg_data_active);

b5_Alpha_time=my_Alpha_filter(b5_filtered_eeg_data_active);
            b5_Beta_time=my_Beta_filter(b5_filtered_eeg_data_active);

            [b5_Raw_X_Theta, b5_Raw_fft_Theta]=my_fft(b5_Theta_time);
            plot(app.UIAxes_2,b5_Raw_X_Theta, abs(b5_Raw_fft_Theta));

            [b5_Raw_X_Alpha, b5_Raw_fft_Alpha]=my_fft(b5_Alpha_time);
            plot(app.UIAxes_3,b5_Raw_X_Alpha, abs(b5_Raw_fft_Alpha));

            [b5_Raw_X_Beta, b5_Raw_fft_Beta]=my_fft(b5_Beta_time);
            plot(app.UIAxes_4,b5_Raw_X_Beta, abs(b5_Raw_fft_Beta));

            b5_PSD_Pwelch_theta=pwelch(b5_Theta_time);
            b5_mean_PSD_Pwelch_theta=mean(b5_PSD_Pwelch_theta);
            b5_SD_PSD_Pwelch_theta=std(b5_PSD_Pwelch_theta);
            b5_var_PSD_Pwelch_theta=var(b5_PSD_Pwelch_theta);
```

```
            b5_PSD_Pwelch_alpha=pwelch(b5_Alpha_time);
            b5_mean_PSD_Pwelch_alpha=mean(b5_PSD_Pwelch_alpha);
            b5_SD_PSD_Pwelch_alpha=std(b5_PSD_Pwelch_alpha);
            b5_var_PSD_Pwelch_alpha=var(b5_PSD_Pwelch_alpha);

            b5_PSD_Pwelch_beta=pwelch(b5_Beta_time);
            b5_mean_PSD_Pwelch_beta=mean(b5_PSD_Pwelch_beta);
            b5_SD_PSD_Pwelch_beta=std(b5_PSD_Pwelch_beta);
            b5_var_PSD_Pwelch_beta=var(b5_PSD_Pwelch_beta);

            b5_x=[b5_mean_PSD_Pwelch_theta; b5_mean_PSD_Pwelch_alpha;
b5_mean_PSD_Pwelch_beta];
            bar(app.UIAxes_5,b5_x);

            b5_y=[b5_SD_PSD_Pwelch_theta, b5_SD_PSD_Pwelch_alpha,
b5_SD_PSD_Pwelch_beta];
            bar(app.UIAxes_6,b5_y);

            b5_z=[b5_var_PSD_Pwelch_theta, b5_var_PSD_Pwelch_alpha,
b5_var_PSD_Pwelch_beta];
            bar(app.UIAxes_8,b5_z);
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are
created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Color = [0.7098 0.9608 0.9608];
            app.UIFigure.Position = [40 40 1297 720];
            app.UIFigure.Name = 'UI Figure';

            % Create Subject1Button
            app.Subject1Button = uibutton(app.UIFigure, 'push');
            app.Subject1Button.ButtonPushedFcn =
createCallbackFcn(app, @Subject1ButtonPushed, true);
            app.Subject1Button.BackgroundColor = [1 1 0.0667];
            app.Subject1Button.Position = [11 642 137 32];
            app.Subject1Button.Text = 'Subject 1';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'RAW DATA')
            xlabel(app.UIAxes, 'TIME')
            ylabel(app.UIAxes, 'AMPLITUDE')
            app.UIAxes.PlotBoxAspectRatio = [1.86363636363636 1 1];
            app.UIAxes.FontSize = 9;
            app.UIAxes.Position = [1 289 310 212];

            % Create UIAxes_2
            app.UIAxes_2 = uiaxes(app.UIFigure);
            title(app.UIAxes_2, 'THETA BAND')
```

```
            xlabel(app.UIAxes_2, 'FREQUENCY')
            ylabel(app.UIAxes_2, 'AMPLITUDE')
            app.UIAxes_2.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_2.FontSize = 9;
            app.UIAxes_2.XLim = [0 50];
            app.UIAxes_2.ColorOrder = [1 0 0;0.9294 0.6941
0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_2.Position = [327 289 310 212];

            % Create UIAxes_3
            app.UIAxes_3 = uiaxes(app.UIFigure);
            title(app.UIAxes_3, 'ALPHA BAND')
            xlabel(app.UIAxes_3, 'FREQUENCY')
            ylabel(app.UIAxes_3, 'AMPLITUDE')
            app.UIAxes_3.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_3.FontSize = 9;
            app.UIAxes_3.XLim = [0 50];
            app.UIAxes_3.ColorOrder = [0.9686 0.4902 0.0078;0.9294
0.6941 0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_3.Position = [650 289 310 212];

            % Create UIAxes_4
            app.UIAxes_4 = uiaxes(app.UIFigure);
            title(app.UIAxes_4, 'BETA BAND')
            xlabel(app.UIAxes_4, 'FREQUENCY')
            ylabel(app.UIAxes_4, 'AMPLITUDE')
            app.UIAxes_4.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_4.FontSize = 9;
            app.UIAxes_4.XLim = [0 50];
            app.UIAxes_4.ColorOrder = [0 1 0;0.9294 0.6941
0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_4.Position = [975 289 310 212];

            % Create UIAxes_5
            app.UIAxes_5 = uiaxes(app.UIFigure);
            title(app.UIAxes_5, 'PSD MEAN')
            xlabel(app.UIAxes_5, 'BAND')
            ylabel(app.UIAxes_5, 'AMPLITUDE')
            app.UIAxes_5.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_5.FontSize = 9;
            app.UIAxes_5.ColorOrder = [1 0 1;0.851 0.3255 0.098;0.9294
0.6941 0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_5.XTick = [0 0.5 1];
            app.UIAxes_5.XTickLabel = {'THETA'; 'ALPHA'; 'BETA'};
            app.UIAxes_5.Position = [327 62 310 212];

            % Create UIAxes_6
            app.UIAxes_6 = uiaxes(app.UIFigure);
            title(app.UIAxes_6, 'PSD STANDARD DEV')
            xlabel(app.UIAxes_6, 'BAND')
            ylabel(app.UIAxes_6, 'AMPLITUDE')
            app.UIAxes_6.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_6.FontSize = 9;
```

```
            app.UIAxes_6.ColorOrder = [1 1 0;0.9294 0.6941
0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_6.XTick = [0 0.5 1];
            app.UIAxes_6.XTickLabel = {'THETA'; 'ALPHA'; 'BETA'};
            app.UIAxes_6.BackgroundColor = [0.9412 0.9412 0.9412];
            app.UIAxes_6.Position = [650 64 310 209];

            % Create Image
            app.Image = uiimage(app.UIFigure);
            app.Image.Position = [700 558 212 162];
            app.Image.ImageSource = '01-Logo UMP_Full Color.png';

            % Create
EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel =
uilabel(app.UIFigure);

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel.Fon
tName = 'Arial';

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel.Fon
tSize = 18;

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel.Fon
tWeight = 'bold';

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel.Pos
ition = [455 559 729 22];

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel.Tex
t = 'EEG MECHANISM INTERACTION TO EVALUATE VEHICLE''S DRIVER
MICROSLEEP';

            % Create DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label
            app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label =
uilabel(app.UIFigure);

app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.HorizontalAlignmen
t = 'center';

app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.FontName =
'Arial';

app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.FontSize = 18;

app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.FontWeight =
'bold';

app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.Position = [533
538 574 22];
            app.DESIGNEDBYMUHAMMADHAZIMBINMOHDYUSOFEA18158Label.Text =
'DESIGNED BY MUHAMMAD HAZIM BIN MOHD YUSOF (EA18158)';

            % Create SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel
```

```
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel =
uilabel(app.UIFigure);

app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.HorizontalAlignment =
'center';
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.FontName =
'Arial';
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.FontSize =
18;
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.FontWeight =
'bold';
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.Position =
[581 517 477 22];
            app.SUPERVISEDBYIRTSDRNORIZAMBINSULAIMANLabel.Text =
'SUPERVISED BY IR.TS. DR. NORIZAM BIN SULAIMAN';

            % Create DetectLamp
            app.DetectLamp = uilamp(app.UIFigure);
            app.DetectLamp.Position = [717 2 26 26];
            app.DetectLamp.Color = [0.902 0.902 0.902];

            % Create NoDetectLamp
            app.NoDetectLamp = uilamp(app.UIFigure);
            app.NoDetectLamp.Position = [717 30 26 26];
            app.NoDetectLamp.Color = [0.902 0.902 0.902];

            % Create MICROSLEEPAPPEARANCELabel
            app.MICROSLEEPAPPEARANCELabel = uilabel(app.UIFigure);
            app.MICROSLEEPAPPEARANCELabel.BackgroundColor = [0.9686
0.302 0.302];
            app.MICROSLEEPAPPEARANCELabel.FontWeight = 'bold';
            app.MICROSLEEPAPPEARANCELabel.Position = [525 20 176 22];
            app.MICROSLEEPAPPEARANCELabel.Text = 'MICROSLEEP
APPEARANCE:';

            % Create Subject2Button
            app.Subject2Button = uibutton(app.UIFigure, 'push');
            app.Subject2Button.ButtonPushedFcn =
createCallbackFcn(app, @Subject2ButtonPushed, true);
            app.Subject2Button.BackgroundColor = [1 1 0.0667];
            app.Subject2Button.Position = [11 611 137 32];
            app.Subject2Button.Text = 'Subject 2';

            % Create Subject3Button
            app.Subject3Button = uibutton(app.UIFigure, 'push');
            app.Subject3Button.ButtonPushedFcn =
createCallbackFcn(app, @Subject3ButtonPushed, true);
            app.Subject3Button.BackgroundColor = [1 1 0.0667];
            app.Subject3Button.Position = [11 580 137 32];
            app.Subject3Button.Text = 'Subject 3';

            % Create RELAXEDSTATELabel
            app.RELAXEDSTATELabel = uilabel(app.UIFigure);
            app.RELAXEDSTATELabel.BackgroundColor = [0.949 0.7765
0.0863];
            app.RELAXEDSTATELabel.FontWeight = 'bold';
            app.RELAXEDSTATELabel.Position = [28 687 104 22];
```

```
            app.RELAXEDSTATELabel.Text = 'RELAXED STATE';

            % Create Subject4Button
            app.Subject4Button = uibutton(app.UIFigure, 'push');
            app.Subject4Button.ButtonPushedFcn =
createCallbackFcn(app, @Subject4ButtonPushed, true);
            app.Subject4Button.BackgroundColor = [1 1 0.0667];
            app.Subject4Button.Position = [11 549 137 32];
            app.Subject4Button.Text = 'Subject 4';

            % Create Subject5Button
            app.Subject5Button = uibutton(app.UIFigure, 'push');
            app.Subject5Button.ButtonPushedFcn =
createCallbackFcn(app, @Subject5ButtonPushed, true);
            app.Subject5Button.BackgroundColor = [1 1 0.0667];
            app.Subject5Button.Position = [11 518 137 32];
            app.Subject5Button.Text = 'Subject 5';

            % Create UIAxes_7
            app.UIAxes_7 = uiaxes(app.UIFigure);
            title(app.UIAxes_7, 'FILTERED EEG IN FREQ.')
            xlabel(app.UIAxes_7, 'FREQUENCY')
            ylabel(app.UIAxes_7, 'AMPLITUDE')
            app.UIAxes_7.PlotBoxAspectRatio = [1.86363636363636 1 1];
            app.UIAxes_7.FontSize = 9;
            app.UIAxes_7.XLim = [1 50];
            app.UIAxes_7.Position = [1 61 310 212];

            % Create Subject1Button_2
            app.Subject1Button_2 = uibutton(app.UIFigure, 'push');
            app.Subject1Button_2.ButtonPushedFcn =
createCallbackFcn(app, @Subject1Button_2Pushed, true);
            app.Subject1Button_2.BackgroundColor = [0.302 0.7451
0.9333];
            app.Subject1Button_2.Position = [174 641 137 32];
            app.Subject1Button_2.Text = 'Subject 1';

            % Create Subject2Button_2
            app.Subject2Button_2 = uibutton(app.UIFigure, 'push');
            app.Subject2Button_2.ButtonPushedFcn =
createCallbackFcn(app, @Subject2Button_2Pushed, true);
            app.Subject2Button_2.BackgroundColor = [0.302 0.7451
0.9333];
            app.Subject2Button_2.Position = [174 610 137 32];
            app.Subject2Button_2.Text = 'Subject 2';

            % Create Subject3Button_2
            app.Subject3Button_2 = uibutton(app.UIFigure, 'push');
            app.Subject3Button_2.ButtonPushedFcn =
createCallbackFcn(app, @Subject3Button_2Pushed, true);
            app.Subject3Button_2.BackgroundColor = [0.302 0.7451
0.9333];
            app.Subject3Button_2.Position = [174 579 137 32];
            app.Subject3Button_2.Text = 'Subject 3';

            % Create Subject4Button_2
            app.Subject4Button_2 = uibutton(app.UIFigure, 'push');
```

```
            app.Subject4Button_2.ButtonPushedFcn =
createCallbackFcn(app, @Subject4Button_2Pushed, true);
            app.Subject4Button_2.BackgroundColor = [0.302 0.7451
0.9333];
            app.Subject4Button_2.Position = [174 548 137 32];
            app.Subject4Button_2.Text = 'Subject 4';

            % Create Subject5Button_2
            app.Subject5Button_2 = uibutton(app.UIFigure, 'push');
            app.Subject5Button_2.ButtonPushedFcn =
createCallbackFcn(app, @Subject5Button_2Pushed, true);
            app.Subject5Button_2.BackgroundColor = [0.302 0.7451
0.9333];
            app.Subject5Button_2.Position = [174 517 137 32];
            app.Subject5Button_2.Text = 'Subject 5';

            % Create DRIVINGSTATELabel
            app.DRIVINGSTATELabel = uilabel(app.UIFigure);
            app.DRIVINGSTATELabel.BackgroundColor = [0 0 1];
            app.DRIVINGSTATELabel.FontWeight = 'bold';
            app.DRIVINGSTATELabel.FontColor = [1 1 1];
            app.DRIVINGSTATELabel.Position = [195 687 96 22];
            app.DRIVINGSTATELabel.Text = 'DRIVING STATE';

            % Create UNDETECTEDLabel
            app.UNDETECTEDLabel = uilabel(app.UIFigure);
            app.UNDETECTEDLabel.BackgroundColor = [0.7098 0.9608
0.9608];
            app.UNDETECTEDLabel.FontWeight = 'bold';
            app.UNDETECTEDLabel.Position = [755 32 88 22];
            app.UNDETECTEDLabel.Text = 'UNDETECTED';

            % Create DETECTEDLabel
            app.DETECTEDLabel = uilabel(app.UIFigure);
            app.DETECTEDLabel.BackgroundColor = [0.7098 0.9608
0.9608];
            app.DETECTEDLabel.FontWeight = 'bold';
            app.DETECTEDLabel.Position = [755 4 70 22];
            app.DETECTEDLabel.Text = 'DETECTED';

            % Create SubjectLabel
            app.SubjectLabel = uilabel(app.UIFigure);
            app.SubjectLabel.HorizontalAlignment = 'right';
            app.SubjectLabel.FontWeight = 'bold';
            app.SubjectLabel.Position = [318 37 53 22];
            app.SubjectLabel.Text = 'Subject:';

            % Create SubjectEditField
            app.SubjectEditField = uieditfield(app.UIFigure, 'text');
            app.SubjectEditField.HorizontalAlignment = 'center';
            app.SubjectEditField.FontWeight = 'bold';
            app.SubjectEditField.Position = [386 37 100 22];

            % Create ConditionLabel
            app.ConditionLabel = uilabel(app.UIFigure);
            app.ConditionLabel.HorizontalAlignment = 'right';
            app.ConditionLabel.FontWeight = 'bold';
```

```
            app.ConditionLabel.Position = [306 5 65 22];
            app.ConditionLabel.Text = 'Condition:';

            % Create ConditionEditField
            app.ConditionEditField = uieditfield(app.UIFigure,
'text');
            app.ConditionEditField.HorizontalAlignment = 'center';
            app.ConditionEditField.FontWeight = 'bold';
            app.ConditionEditField.Position = [386 5 100 22];

            % Create
EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2 =
uilabel(app.UIFigure);

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2.F
ontName = 'Arial';

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2.F
ontWeight = 'bold';

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2.P
osition = [129 184 486 22];

app.EEGMECHANISMINTERACTIONTOEVALUATEVEHICLESDRIVERMICROSLEEPLabel_2.T
ext = 'EEG MECHANISM INTERACTION TO EVALUATE VEHICLE''S DRIVER
MICROSLEEP';

            % Create Image2
            app.Image2 = uiimage(app.UIFigure);
            app.Image2.Position = [478 584 129 119];
            app.Image2.ImageSource = 'paskal.png';

            % Create UIAxes_8
            app.UIAxes_8 = uiaxes(app.UIFigure);
            title(app.UIAxes_8, 'PSD VARIANCE')
            xlabel(app.UIAxes_8, 'BAND')
            ylabel(app.UIAxes_8, 'AMPLITUDE')
            app.UIAxes_8.PlotBoxAspectRatio = [1.81159420289855 1 1];
            app.UIAxes_8.FontSize = 9;
            app.UIAxes_8.ColorOrder = [0.0275 0.5294 0.0275;0.9294
0.6941 0.1255;0.4941 0.1843 0.5569;0.4667 0.6745 0.1882;0.302 0.7451
0.9333;0.6353 0.0784 0.1843];
            app.UIAxes_8.XTick = [0 0.5 1];
            app.UIAxes_8.XTickLabel = {'THETA'; 'ALPHA'; 'BETA'};
            app.UIAxes_8.BackgroundColor = [0.9412 0.9412 0.9412];
            app.UIAxes_8.Position = [975 64 310 209];

            % Create Image3
            app.Image3 = uiimage(app.UIFigure);
            app.Image3.Position = [1029 585 100 100];
            app.Image3.ImageSource = 'drnorizam-removebg-preview.png';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
```

```
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = fyp_v1

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

**CLASSIFIER MATLAB CODE**

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates
the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or
to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and
response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier.
The
%       struct contains various fields with information about the
trained
%       classifier.
%
```

```
%       trainedClassifier.predictFcn: a function to make predictions on
new
%        data.
%
%       validationAccuracy: a double containing the accuracy in
percent. In
%        the app, the History list displays this overall accuracy score
for
%        each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your
original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data
set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new
data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as
used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 13-Jun-2022 19:10:23


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'eegRawValue', 'attention', 'meditation',
'blinkStrength'};
predictors = inputTable(:, predictorNames);
response = inputTable.Condition;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the
classifier.
classificationKNN = fitcknn(...
    predictors, ...
    response, ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', categorical({'Active'; 'Drowsy'}));

% Create the result struct with predict function
```

```
predictorExtractionFcn = @(t) t(:, predictorNames);
knnPredictFcn = @(x) predict(classificationKNN, x);
trainedClassifier.predictFcn = @(x)
knnPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'attention', 'blinkStrength',
'eegRawValue', 'meditation'};
trainedClassifier.ClassificationKNN = classificationKNN;
trainedClassifier.About = 'This struct is a trained model exported
from Classification Learner R2019b.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the
name of the variable that is this struct, e.g. ''trainedModel''. \n
\nThe table, T, must contain the variables returned by: \n
c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype)
must match the original training data. \nAdditional variables are
ignored. \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.
inputTable = trainingData;
predictorNames = {'eegRawValue', 'attention', 'meditation',
'blinkStrength'};
predictors = inputTable(:, predictorNames);
response = inputTable.Condition;
isCategoricalPredictor = [false, false, false, false];

% Set up holdout validation
cvp = cvpartition(response, 'Holdout', 0.25);
trainingPredictors = predictors(cvp.training, :);
trainingResponse = response(cvp.training, :);
trainingIsCategoricalPredictor = isCategoricalPredictor;

% Train a classifier
% This code specifies all the classifier options and trains the
classifier.
classificationKNN = fitcknn(...
    trainingPredictors, ...
    trainingResponse, ...
    'Distance', 'Euclidean', ...
    'Exponent', [], ...
    'NumNeighbors', 1, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', categorical({'Active'; 'Drowsy'}));

% Create the result struct with predict function
knnPredictFcn = @(x) predict(classificationKNN, x);
validationPredictFcn = @(x) knnPredictFcn(x);

% Add additional fields to the result struct
```

```
% Compute validation predictions
validationPredictors = predictors(cvp.test, :);
validationResponse = response(cvp.test, :);
[validationPredictions, validationScores] =
validationPredictFcn(validationPredictors);

% Compute validation accuracy
correctPredictions = (validationPredictions == validationResponse);
isMissing = ismissing(validationResponse);
correctPredictions = correctPredictions(~isMissing);
validationAccuracy =
sum(correctPredictions)/length(correctPredictions);
```

**FFT FUNCTION CODE**

```
function [xfft,fff_1] = my_fft(data)
Fs=512;
N=length(data);
nfft=2^nextpow2(N);
xfft=(Fs*(0:nfft/2-1)/nfft);
fft_1=fft(data,nfft);
fff_1=fft_1(1:nfft/2);
end
```

**THETA FUNCTION CODE**

```
function [Filtered_data] = my_Theta_filter(data) %#ok<*FNDEF>
Fs=512;
order=5;
F_low=4;
F_high=7;
[b,a]=butter(order,[F_low,F_high]/(Fs/2),'bandpass');
Filtered_data=filter(b,a,data);
end
```

**ALPHA FUNCTION CODE**

```
function [Filtered_data] = my_Alpha_filter(data) %#ok<*FNDEF>
Fs=512;
order=5;
F_low=8;
F_high=12;
[b,a]=butter(order,[F_low,F_high]/(Fs/2),'bandpass');
Filtered_data=filter(b,a,data);
end
```

**BETA FUNCTION CODE**

```
function [Filtered_data] = my_Beta_filter(data) %#ok<*FNDEF>
Fs=512;
order=5;
F_low=13;
F_high=30;
[b,a]=butter(order,[F_low,F_high]/(Fs/2),'bandpass');
Filtered_data=filter(b,a,data);
```