A PROJECT REPORT
ON

"Hand Movements Prediction System"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING
BY

(Master Farheen Adam Shabnam)	17CO05
(Karia Janvi Vinay Seema)	18CO02
(Shaikh Misbah Karim Shabana)	18CO11
(Chimaokar Shifa Shahnawaz Shaheen)	18CO12

UNDER THE GUIDANCE OF
(Prof. Samreen Kazi)

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector -16, Near Thana Naka,
Khanda Gaon, New Panvel - 410206

2021-2022
AFFILIATED TO
UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"Hand Movements Prediction System"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

(Master Farheen Adam Shabnam)	17CO05
(Karia Janvi Vinay Seema)	18CO02
(Shaikh Misbah Karim Shabana)	18CO11
(Chimaokar Shifa Shahnawaz Shaheen)	18CO12

UNDER THE GUIDANCE OF
(Prof. Samreen Kazi)

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 & 3, Sector -16, Near Thana Naka,
Khanda Gaon, New Panvel - 410206

2021-2022
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-I-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is to certify that the project entitled

"Hand Movements Prediction System"

submitted by

(Master Farheen Adam Shabnam)	17CO05
(Karia Janvi Vinay Seema)	18CO02
(Shaikh Misbah Karim Shabana)	18CO11
(Chimaokar Shifa Shahnawaz Shaheen)	18CO12

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during the year 2021-2022, under our guidance.

Date: / /

Prof. Samreen Kazi
Project Supervisor

Prof. Nusrat Jahan
Project Coordinator

Prof. Tabrez Khan
Head of Department

Dr. Abdul Razak Honnutagi
Director

External Examiner

Acknowlegments

We would like to take the opportunity to express our sincere thanks to our guide **Prof. Samreen Kazi**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for her invaluable support and guidance throughout our project research work. Without her kind guidance & support this was not possible.

We are grateful to her for her timely feedback which helped us track and schedule the process effectively. Her time, ideas and encouragement that she gave helped us to complete our project efficiently.

We would like to express our deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Nusrat Jahan**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of the Computer Engineering Department who helped us directly or indirectly during this course of work.

Master Farheen Adam Shabnam
17CO05

Karia Janvi Vinay Seema
18CO02

Shaikh Misbah Karim Shabana
18CO11

Chimaokar Shifa Shahnawaz Shaheen
18CO12

Project I Approval for Bachelor of Engineering

This project entitled ***Hand Movements Prediction System*** by ***Master Farheen Adam, Karia Janvi Vinay, Shaikh Misbah Karim and Chimaokar Shifa Shahnawaz*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. ....

2. ....

Supervisors

1. ....

2. ....

Chairman

....

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Master Farheen Adam Shabnam
Roll Number: 17CO05

Karia Janvi Vinay Seema
Roll Number: 18CO02

Shaikh Misbah Karim Shabana
Roll Number: 18CO11

Chimaokar Shifa Shahnawaz Shaheen
Roll Number: 18CO12

ABSTRACT

Title : Hand Movements Prediction System

Electroencephalography (EEG) is a non-invasive method of recording brain activity right from the scalp using electrodes that are attached to an EEG cap. Analysis of these signals has many promising applications such as making EEG-based Brain Computer Interfaces (BCI). EEG-based BCIs allow patients with disabilities to be able to control the movements of prosthetic limbs and other devices.

In a healthy person, the motor cortex is crucial for the planning, control, and execution of voluntary movements. But in case of people with disabilities either the brain is not able to communicate with the muscles or has lost control of them. Moreover, for performing everyday rituals, the patient needs to have control of the movements(of limbs) involved in these tasks.

To aid these people, we propose to build a machine learning model to classify and predict hand movements based on EEG (electroencephalography) readings of able bodied people performing these hand movements.

Keywords : Electroencephalography (EEG), non-invasive, Brain Computer Interfaces (BCI), prosthetic, motor cortex, voluntary, EEG cap, machine learning, disabilities, signal processing, model, feature extraction, classification, training set, testing set, decision tree, logistic regression, multi-label classification.

Glossary :

A:

API - a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

Application - a software program that runs on your computer.

Analysis - detailed examination of the elements or structure of something.

Architecture - The complex or carefully designed structure of something.

Artifact - Irregularities in an EEG recording due to internal or external stimuli.

Automation - the use or introduction of automatic equipment in a manufacturing or other process or facility.

B:

BCI - A brain computer interface is a direct neural pathway between the brain and an external device.

C:

Classification - A supervised machine learning algorithm used to make prediction.

Covariance - A statistical measure of the relationship between two random variables.

D:

Database - A structured set of data held in a computer, especially one that is accessible in various ways.

Denoising - The act of separating the signal from a mix of noise and signal.

Deployment - The process of running an application on a server or device.

Dimensionality - The attributes in a dataset.

E:

EEG - An electroencephalogram measures brain activity (in ranges of μV) from the surface of the scalp.

Eigenvectors - An eigenvector, corresponding to a real nonzero eigenvalue, points in a direction in which it is stretched by the transformation.

Electrical impulses - The small electrical activity generated by the brain while performing some task.

Energy-entropy - The state of relative uncertainty of energy.

F:

Feature - Attribute of a dataset.

fMRI - Functional magnetic resonance imaging (fMRI) measures the small changes in blood flow that occur with brain activity.

Framework - A basic structure underlying a system, concept, or text.

Frontal - Region of brain located directly behind the forehead.

I:

Integrate - combine (one thing) with another to form a whole.

M:

Machine Learning - It is a subset of artificial intelligence which enables processes to predict outcome without being explicitly programmed to do so.

Maintenance - the process of preserving a condition or situation or the state of being preserved.

Model - The output of the training function of a machine learning algorithm.

Motor Cortex - The part of cerebral cortex of the brain responsible for action.

Motor Imagery - The imagination of motor function.

MLP - Multi-layer perceptron is a neural network with multiple intermediate layers.

MRI - Magnetic Resonance Imaging is a brain imaging technique that gives information about the structure of the brain.

MVC - The model-views-controller framework of a web application.

N:

Non-Invasive - Superficial processes that do not pose a threat to the user.

P:

PCA - Principle component analysis is a technique for dimensionality reduction of data.

Posterior Cortex - It refers to the back of the cerebral cortex.

Prosthetic - An artificial limb.

PSD - Power Spectral Density is a measure of the distribution of power in a signal and is often used in denoising and as well as feature selection of a signal.

R:

RMS - It is the root of the mean square value.

S:

Sensorimotor Cortex - It is a subset of the motor cortex in the frontal lobe and is responsible for sensory-motor rhythms.

Signal-Processing - It refers to manipulating signals and applying different processing techniques on the signal.

Somatosensory Cortex - The somatosensory cortex is a region of the brain which is responsible for receiving and processing sensory information from across the body.

SVM - Support Vector Machines are a supervised machine learning algorithm use to classify data.

T:

Temporal Resolution - it simply refers to the amount of times data was collected and or recorded from the same region.

V:

Validation - the action of checking or proving the validity or accuracy of something.

W:

Web application - an application program that is stored on a remote server and delivered over the Internet through a browser interface.

Web services - a software service used to communicate between two devices on a network.

Contents

Acknowledgments . . . iii
Project I Approval for Bachelor of Engineering . . . iv
Declaration . . . v
Abstract . . . vi
Table of Contents . . . xi

1 Introduction **1**
1.1 Purpose . . . 1
1.2 Project Scope . . . 1
1.3 Project Goals and Objectives . . . 2
1.3.1 Goals . . . 2
1.3.2 Objectives . . . 2
1.4 Organisation of Report . . . 2

2 Literature Survey **4**
2.1 Comparative Analysis of Various Filtering Techniques for Denoising EEG Signals 4
2.1.1 Advantages of Paper . . . 4
2.1.2 Disadvantages of Paper . . . 4
2.1.3 How to Overcome . . . 4
2.2 Wavelet Transform Use for Feature Extraction and EEG Signal Segments Classification . . . 5
2.2.1 Advantages of Paper . . . 5
2.2.2 Disadvantages of Paper . . . 5
2.2.3 How to Overcome . . . 5
2.3 EEG Based Motor Imagery Classification Using SVM and MLP . . . 5
2.3.1 Advantages of Paper . . . 5
2.3.2 Disadvantages of Paper . . . 6
2.3.3 How to Overcome . . . 6
2.4 Technical Review . . . 6
2.4.1 Advantages of Technology . . . 7
2.4.2 Reasons to use this Technology . . . 8

3 Project Planning **9**
3.1 Members and Capabilities . . . 9
3.2 Roles and Responsibilities . . . 9
3.3 Assumptions and Constraints . . . 9
3.3.1 Assumptions . . . 9
3.3.2 Constraints . . . 10

3.4 Project Management Approach 10
3.5 Ground Rules for the Project 11
3.6 Project Budget 12
3.7 Project Timeline 12

4 Software Requirements Specification **14**
4.1 Overall Description 14
4.1.1 Product Perspective 14
4.1.2 Product Features 15
4.1.3 User Classes and Characteristics 15
4.1.4 Operating Environment 15
4.1.5 Design and Implementation Constraints 16
4.2 System Features 16
4.3 External Interface Requirements 17
4.3.1 User Interfaces 17
4.3.2 Hardware Interfaces 17
4.3.3 Software Interfaces 17
4.3.4 Communications Interfaces 17
4.4 Nonfunctional Requirements 18
4.4.1 Performance Requirements 18
4.4.2 Safety Requirements 18
4.4.3 Security Requirements 18

5 System Design **19**
5.1 System Requirements Definition 19
5.1.1 Functional requirements 19
5.1.2 System requirements (non-functional requirements) 19
5.2 System Architecture Design 20
5.3 Sub-system Development 21
5.3.1 Module 1 : Front-end Development Module 21
5.3.2 Module 2 : Denoising and Feature Extraction Module 21
5.3.3 Module 3 : Model Training and Back-end Development Module 22
5.4 Systems Integration 32
5.4.1 Class Diagram 32
5.4.2 Sequence Diagram 32

6 Implementation **33**
6.1 The Manage.py File 33
6.2 The Views.py File 33
6.3 The MainPage.html File 36

7 System Testing **44**
7.1 Test Cases and Test Results 44
7.2 Sample of a Test Case 44

7.2.1 Software Quality Attributes . . . 47

8 Screenshots of Project **48**
8.1 Landing Page . . . 48
8.2 About Us . . . 49
8.2.1 Motivation . . . 49
8.2.2 Application of EEG . . . 49
8.2.3 Types of EEG . . . 50
8.2.4 Uses of EEG . . . 50
8.3 Main Section . . . 51
8.3.1 Uploads . . . 51
8.3.2 Output . . . 52
8.4 Our Team . . . 53

9 Conclusion and Future Scope **54**
9.1 Conclusion . . . 54
9.2 Future Scope . . . 54

References **55**

Appendix A **56**

A Electrode Placement System for EEG **56**
A.1 International 10-20 System . . . 56
A.1.1 Electrode Nomenclature . . . 56
A.1.2 Measurement & Positioning . . . 57
A.2 10-10 System . . . 58
A.2.1 Modified Combinatorial Nomenclature (MCN) . . . 58

Appendix B **60**

B EEG Waveforms **60**

List of Figures

3.1 Project Timeline Spreadsheet 12
3.2 Project Timeline Gantt Chart 13

5.1 System Architecture 20
5.2 Framework of Proposed System 20
5.3 Class Diagram of System 32
5.4 Sequence Diagram of System 32

8.1 Landing Page 48
8.2 Motivation 49
8.3 Application of EEG 49
8.4 Types of EEG 50
8.5 Uses of EEG 50
8.6 Upload Page 51
8.7 File Dialogue for accepting only .csv files 51
8.8 Uploaded EEG file 51
8.9 Sample Output 1 52
8.10 Sample Output 2 52
8.11 Sample Output 3 52
8.12 Sample Output 4 53
8.13 Sample Output 5 53
8.14 Our Team 53

A.1 International 10-20 System for Electrode Placement(1) 56
A.2 Modified 10-10 System for Electrode Placement(1) 58

B.1 EEG Waveform for Delta Waves (< 4Hz)(2) 60
B.2 EEG Waveform for Theta Waves (4 - 7Hz)(2) 60
B.3 EEG Waveform for Alpha Waves (8 - 15Hz)(2) 60
B.4 EEG Waveform for Beta Waves (16 - 31Hz)(2) 61
B.5 EEG Waveform for Gamma Waves (> 32Hz)(2) 61
B.6 EEG Waveform for Sensorimotor Rhythm or Mu Waves (8 - 12Hz)(2) 61

List of Tables

2.1 Summary of Literature Review 6
2.2 Frequency Bands of EEG signals(2)(3) 7

3.1 Table of Capabilities 9
3.2 Table of Responsibilities 9

4.1 Summary of EEG-based BCI robotic arm studies(3) 15

7.1 Test Cases for Hand Movements Prediction System 44

Chapter 1

Introduction

1.1 Purpose

To perform some motor action like walking, picking objects, throwing, or moving in general, our brain sends some directions to the body part involved in said action in the form of electrical signals which leads to us performing said action. But due to defects in the limb(s) or connection between the limb(s) and our brain, these actions or movements cannot be carried out. This makes the world inaccessible for the one affected by this. Although prosthetic devices exist, they slowly become less and less useful in more intricate tasks. The solution here is making these devices more interactive and functional but this often comes at the price of invasiveness and low affordability.

EEG-based Brain Computer Interfaces allow for patients with disabilities to control the movements of artificial limbs or exoskeletons.

EEG signals record brain activity from the scalp using an EEG cap with minimal invasiveness. Analysing these signals with respect to motor functions will allow us to predict motor functions just from the EEG recording.

To aid these people, we propose to build a machine learning model to classify and predict hand movements based on EEG (electroencephalography) readings of able bodied people performing these hand movements.

1.2 Project Scope

Our proposed solution to the problem statement is to build a multi-label classification model using machine learning and deep learning algorithms to accurately predict the different motions of the upper limb.

Our model will be able to quickly and accurately predict (different and coexisting) hand movements given the electroencephalography readings of a subject.

This will also help in improving our understanding of EEG signals themselves.

An online, accurate and inexpensive system will allow for the development of various technologies and applications to build more functional BCI based prosthetic devices.

Based on our system other types of interactive BCI can also be developed leading us one step further into human machine interaction.

1.3 Project Goals and Objectives

1.3.1 Goals

- To build a machine learning model to predict multiple actions that a person performs with their hands given the EEG recording of said person

- To build a website using Django framework to deploy our model and present the predictions to the user

- To maximize accuracy and minimise latency

1.3.2 Objectives

- Selecting a suitable dataset(4)

- Preparing the data for further operations

- Channel Selection to make it easier to work with the data

- Feature Extraction to help in classification of the data

- Constructing a model to classify the data to predict various hand movements like grasping, lifting, replacing, etc.

1.4 Organisation of Report

The material presented in this report is organized into 9 chapters.

After this introductory chapter, Chapter-2 gives us the literature survey of the project,i.e., the research papers that were referenced while developing the project.

Chapter-3 informs about the complete planning of the project, the work distribution, the prerequisites etc.

Chapter-4 specifies the software requirements of the project.

Chapter-5 discusses about the system design of the project.

Chapter-6 gives an overview of the implementation of the project.

Chapter-7 shows how the project was tested.

Chapter-8 contains the screenshots of the project.

Chapter-9 concludes the report and states the future scope of our project.

At the very end of our report are a list of references used throughout our work and two appendices to supplement our report.

Chapter 2

Literature Survey

2.1 Comparative Analysis of Various Filtering Techniques for Denoising EEG Signals

EEG signals suffer from the curse of high dimensionality so it becomes extremely important to remove noise and unnecessary data like blinking artifacts, etc.. This paper compares the butterworth filter, adaptive least mean square filter and the wavelet transform for removing artifacts from the eeg recording. All three filters remove artifacts very well. The wavelet transform method yields the best result based on the peak signal-noise ratio and mean square error.(5)

2.1.1 Advantages of Paper

- All three filters remove artifacts very well
- The wavelet transform method yields the best result based on the peak signal-noise ratio and mean square error

2.1.2 Disadvantages of Paper

- This paper uses sleep eeg and epileptic eeg recordings.

2.1.3 How to Overcome

For our project we will be using wavelet transform to denoise eeg recording of motor functions as it is effective on non-stationary signals.

2.2 Wavelet Transform Use for Feature Extraction and EEG Signal Segments Classification

Segmentation, feature extraction and classification of signal components belong to very common problems in various engineering, economical and biomedical applications. The paper is devoted to the use of discrete wavelet transform (DWT) both for signal preprocessing and signal segments feature extraction as an alternative to the commonly used discrete Fourier transform (DFT). Feature vectors belonging to separate signal segments are then classified by a competitive neural network as one of methods of cluster analysis and processing. The paper provides a comparison of classification results using different methods of feature extraction most appropriate for EEG signal components detection. Problems of multichannel segmentation are mentioned in this connection as well.(6)

2.2.1 Advantages of Paper

- DWT can be used for selecting features in the time and frequency domain efficiently.

2.2.2 Disadvantages of Paper

- The main drawback comes due to the high dimensionality of the eeg signals.

2.2.3 How to Overcome

We can select channels relevant to motor function, like those directly over the motor cortex, for dimensionality reduction so our model is more efficient.

2.3 EEG Based Motor Imagery Classification Using SVM and MLP

This paper focuses on the classification of motor imagery of the left-right hand movements from a healthy subject. Elliptic Band-pass filters are used to discard the unwanted signals. Our study was on C3 and C4 electrodes particularly for the left-right limb movements. We deployed various feature extraction techniques on the EEG data. Statistical-based, wavelet-based energy-entropy & RMS, PSD based average power and bad power were performed to form the desired feature vectors. Variants of Support Vector Machines (SVM) were employed for classification and the results were also compared with Multi-layered Perceptron (MLP). Empirical results show that both SVM and MLP were suitable for such motor imagery classifications with the accuracy of 85% and 85.71% respectively. Among all employed feature extraction techniques wavelet-based methods specifically the energy-entropy feature set, gave promising results for both the classifiers.(7)

2.3.1 Advantages of Paper

- Both the SVM and MLP give good classification.

- The energy-entropy feature is very efficient for classification.

- MLP is found to be better than SVM based on accuracy.

2.3.2 Disadvantages of Paper

- Higher dimensionality is a matter of concern for EEG data classification.

- SVMs cannot be used for multi label classification.

2.3.3 How to Overcome

For the purpose of multi label classification we can train models separately on each label and combine the predictions for each label while predicting the result.

Summary of Literature Review

Let us have a summarised recap of our literature review by referring the following table :

Table 2.1: Summary of Literature Review

Features	Paper 1	Paper 2	Paper 3	Proposed System
Noise Reduction	yes	yes	no	yes
Low Dimensionality	no	yes	no	yes
More Functionality	no	no	yes	yes

Apart from the features mentioned in our literature survey we also aim at making our system robust, cost-effective, user-friendly and effective.

2.4 Technical Review

The execution of any voluntary action involves and is heavily dependent on the directions sent by the brain to the actuators (limbs / body parts). These directions are sent in the form of electrical impulses that travel through the nervous system via neurons. The method of monitoring or capturing the electrical activity generated by the brain is called electroencephalography (EEG). It measures the slight voltage fluctuations (of amplitude in μV) on the surface layer of the brain by sensing it through electrodes placed on the scalp.

EEG electrodes are generally placed according to the 10-20 system or the 10-10 system where the numbers represent the ratio of distance between adjacent electrodes to the total front-back or left-right distance.(see Appendix 9)

The recorded brain activity is generally related to two types of processes - rhythmic function or event related potentials (ERPs).

The rhythmic activity is categorized into the following frequency bands:(see Appendix A.2.1 for waveform)

Table 2.2: Frequency Bands of EEG signals(2)(3)

Band	Frequency (Hz)	Location	Activity
Delta	< 4	Frontal in adults Posterior in children	Deep Sleep
Theta	4 - 7	Random	Drowsiness Deep Meditation
Alpha	8 - 15	Posterior	Relaxed
Beta	16 - 31	Frontal	Active Alert
Gamma	> 32	Somatosensory Cortex	Hyperactivity Simultaneous Information Processing
Mu	8 - 12	Sensorimotor Cortex	Rest State Motor Neurons

The ERPs are responses of the brain which are a direct result of a sensory, cognitive, or motor event. The ERPs with respect to motor events are again categorized as either Event Related Synchronization or Event Related Desynchronization.

Performing any motor activity or even just imagining (motor imagery (MI)) / preparing to perform these activities produces oscillations in the motor cortex of the brain, this is known as Sensorimotor Rhythms (SMR). The increase and decrease of this activity in a particular frequency band is what is the event related synchronization and event related desynchronization. The alpha and beta bands are the most influential sensorimotor rhythms when it comes to observing motor function or motor imagery in a subject.

The motor imagery/ function relating to the movement of the upper limbs is observed in the C3 and C4 regions of the brain.(2)

Analysis of these rhythms helps in predicting the various movements of the hand.

2.4.1 Advantages of Technology

As compared to all the other techniques employed for brain imaging like MRI and fMRI, EEG provides many technical advantages.(2)(8)

- As EEG is recorded from the surface of the scalp is a **non-invasive** technique to measure brain activity.

- The hardware costs are also significantly lower as compared to the setup required for other techniques making it easily **affordable** for people who may require it.

- It also avoids the hassle associated with any bulky setup by being **semi-fully portable** and also **does not require a high level of expertise** to record.

- As EEG is normally recorded at a sampling rate of 250 - 2000 Hz it also provides an **impeccable temporal resolution**.

- Deploying our system on the internet makes it **accessible** to a wide range of people.

2.4.2 Reasons to use this Technology

- Currently, there are no realistic, affordable, most importantly non-invasive, or low-risk options for neurologically and/or physically disabled patients to directly control arm prosthetics with their brain activity.

- So far we only have a few laboratory tests to detect and understand EEG signals.

- Exploring the relationship between EEG signals and hand movements can prove to be crucial in developing prosthetic devices that can be controlled through brain activity with minimal invasiveness.

- This may also allow us to further our understanding of EEG signals themselves.

- What grants our system its competitive edge however is its accessibility, affordability coupled with its user-friendliness.

- And it also has the ability to predict overlapping actions improving its functionality as compared to other systems.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

Sr. No.	Name of Member	Capabilities
1	Master Farheen Adam	Design, Hosting
2	Karia Janvi Vinay	Python, Django
3	Shaikh Misbah Karim	HTML, CSS, JS, Bootstrap
4	Chimaokar Shifa Shahnawaz	Python, ML, Signal Processing

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

Sr. No.	Name of Member	Role	Responsibility
1	Master Farheen Adam	Team Member	Design, Deployment
2	Karia Janvi Vinay	Team Leader	Backend, Program Integration
3	Shaikh Misbah Karim	Team Member	Frontend Development
4	Chimaokar Shifa Shahnawaz	Team Member	Machine Learning

3.3 Assumptions and Constraints

3.3.1 Assumptions

1. All needed resources will be available.

2. All members have all the required skills.

3. The EEG will be recorded by an EEG cap using the 10-10 EEG montage (see Appendix A part A.2)

4. The user will have a set of sampled EEG recording.

5. The user will have EEG recordings in the form of a csv file.

3.3.2 Constraints

1. The entire project must be managed within the team of developers.

2. The work must be completed with the available set of resources.

3. Due to constraints on time and expertise this project is limited to a classification system, applications of which being a future scope of our project.

3.4 Project Management Approach

To manage our project we employed the waterfall model. This model divides the entire process into separate phases with the beginning of each phase dependent on the last phase. Due its rigidity, it is simple to use and we have more control over our project as the project is majorly dependent on the team with very little interaction with clients. The approach is very linear and can be managed easily by forming a schedule and setting objectives and deadlines at every stage of the development process.

We began by identifying a real world problem that could be solved through a software solution. We learned that the efficiency of normal limb prosthetics incrementally reduces as the complexity of the tasks they are employed for increases. For instance, while a patient may be able to push doors or pick up larger / lighter objects with their artificial hand they might not be able to grasp or lift smaller objects or say be able to write with a pen etc. To find the solution to this problem we searched for existing systems that help combat this issue. The existing solutions however had a few drawbacks like in-affordability and/or invasiveness.

EEG is a non-invasive , portable , relatively inexpensive way of measuring and recording brain activity with high temporal resolution. By analysing these signals and their relationship with motor function we can use this as a preferable substitute for the expensive and invasive existing systems. Hence, we decided to build our project on the topic “Hand Movements Prediction System”.

Having established our problem statement and objectives for our final year project, we now had the responsibility of constructing a solution. To aid us in the construction of the various modules of our project we collected and studied a handful of IEEE papers relevant to the problem at hand. Out of these we selected three papers for the purposes of gaining some understanding of data preprocessing , signal processing and creating a machine learning model for classification of the EEG data. With this knowledge we began working on designing our web application to deploy our classification model on. After finalising the design of our website we began the development of the same. The next stage of our project would be implementing the model, training the model and testing the model after which we can move on to integrating the front-end and back-end of our project and finally deploying and hosting it on the internet.

Throughout the process, the guidance from our supervisor became the foundation for solving issues whenever they happened to be beyond our scope.

3.5 Ground Rules for the Project

To meet our requirements and for the smooth operation of our project, we have set up a few ground rules to be followed by every member in our team :

1. We treat each other with respect.
2. We intend to develop personal relationships to enhance trust and open communication.
3. We value constructive dialogue. We will avoid being defensive and give feedback in a constructive manner.
4. Each side will come to the table as prepared as possible to expedite the process.
5. One person talks at a time; there are no side discussions
6. We emphasize open and honest communication - there are no hidden agendas.
7. We depersonalize discussion of issues - no attacks on people.
8. We will listen, be non-judgmental and keep an open mind on issues until it is time to decide.
9. You are encouraged to ask genuine ”questions of clarification.” Please avoid asking ”questions of attack.”
10. Please use each other’s first names, not the pronouns ”he” or ”she.”
11. Speak for yourself only.
12. Appeals and attempts to convince should be made to each other and not to the mediator.
13. If something is not working for you, speak up.

14. Try to avoid establishing hard positions, expressing yourself instead in terms of your interests, intentions, and the outcomes that you would like to create.

3.6 Project Budget

This project is very cost-effective. Almost no expenses were spent during the development of our project.

i. The hosting was done on Heroku free of any costs.

ii. The website was made using Django which is an open-source framework.

iii. The datasets for the purpose of building our model were also available on the internet for free.

Hence, our project involving the building of our model and website and the subsequent deployment of the same was done for next to no costs.

3.7 Project Timeline

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Project : Hand Movement Prediction System**	**58.667 day...**	**12/2/21 8:00 AM**	**2/22/22 2:20 PM**		
2		**Data Collection**	**7.75 days?**	**12/2/21 8:00 AM**	**12/13/21 3:00 P...**		
3		Team Meeting	0.25 days?	12/2/21 8:00 AM	12/2/21 10:00 AM		Farheen Master;Janvi Karia;...
4		Finding suitable datasets	3.5 days	12/2/21 10:00 AM	12/7/21 3:00 PM	3	Shifa Chimaokar;Farheen M...
5		Comparing datasets	3.5 days	12/7/21 3:00 PM	12/13/21 10:00 AM	4	Shifa Chimaokar;Farheen M...
6		Dataset Selection	0.5 days?	12/13/21 10:00 AM	12/13/21 3:00 PM	5	Farheen Master;Shifa Chima...
7		**Coding**	**44.917 days**	**12/13/21 3:00 PM**	**2/14/22 2:20 PM**	**6**	
8		**Data Preprocessing**	**13 days**	**12/13/21 3:00 PM**	**12/30/21 3:00 P...**	**6**	
9		Data Cleaning	1.75 days	12/13/21 3:00 PM	12/15/21 1:00 PM	6	Shifa Chimaokar;Janvi Karia
10		Denpoising	5 days	12/15/21 1:00 PM	12/22/21 1:00 PM	9	Shifa Chimaokar
11		Signal Processing	5 days	12/22/21 1:00 PM	12/29/21 1:00 PM	10	Shifa Chimaokar
12		Filtering	1.25 days	12/29/21 1:00 PM	12/30/21 3:00 PM	11	Janvi Karia;Shifa Chimaokar...
13		**Modelling**	**14 days**	**12/30/21 3:00 PM**	**1/19/22 3:00 PM**	**12**	
14		Selecting Algorithm	3.5 days	12/30/21 3:00 PM	1/5/22 10:00 AM	12	Janvi Karia;Shifa Chimaokar
15		Training Model	7 days	1/5/22 10:00 AM	1/14/22 10:00 AM	14	Shifa Chimaokar
16		Testing Model	3.5 days	1/14/22 10:00 AM	1/19/22 3:00 PM	15	Janvi Karia;Misbah Shaikh
17		**Testing**	**14.5 days**	**1/19/22 3:00 PM**	**2/9/22 10:00 AM**	**16**	
18		Functionality	4 days	1/19/22 3:00 PM	1/25/22 3:00 PM	16	Misbah Shaikh
19		Accuracy	4 days	1/25/22 3:00 PM	1/31/22 3:00 PM	18	Janvi Karia
20		Improvisation of Model	5 days	1/31/22 3:00 PM	2/7/22 3:00 PM	19	Shifa Chimaokar
21		Final Backend	1.5 days	2/7/22 3:00 PM	2/9/22 10:00 AM	20	Misbah Shaikh;Janvi Karia
22		Integrate Frontend and Backend	2.667 days	2/9/22 10:00 AM	2/11/22 4:20 PM	21	Misbah Shaikh[50%];Janvi K...
23		Testing final program	0.75 days	2/11/22 4:20 PM	2/14/22 2:20 PM	22	Farheen Master;Janvi Karia;...
24		**Deployment**	**5 days**	**2/14/22 2:20 PM**	**2/21/22 2:20 PM**	**23**	
25		Model Deployment	2.5 days	2/14/22 2:20 PM	2/17/22 9:20 AM	23	Misbah Shaikh;Farheen Master
26		Hosting	2.5 days	2/17/22 9:20 AM	2/21/22 2:20 PM	25	Farheen Master;Janvi Karia
27		Milestone :Project Deployed	1 day?	2/21/22 2:20 PM	2/22/22 2:20 PM	26	

Hand Movements Prediction System

Figure 3.1: Project Timeline Spreadsheet

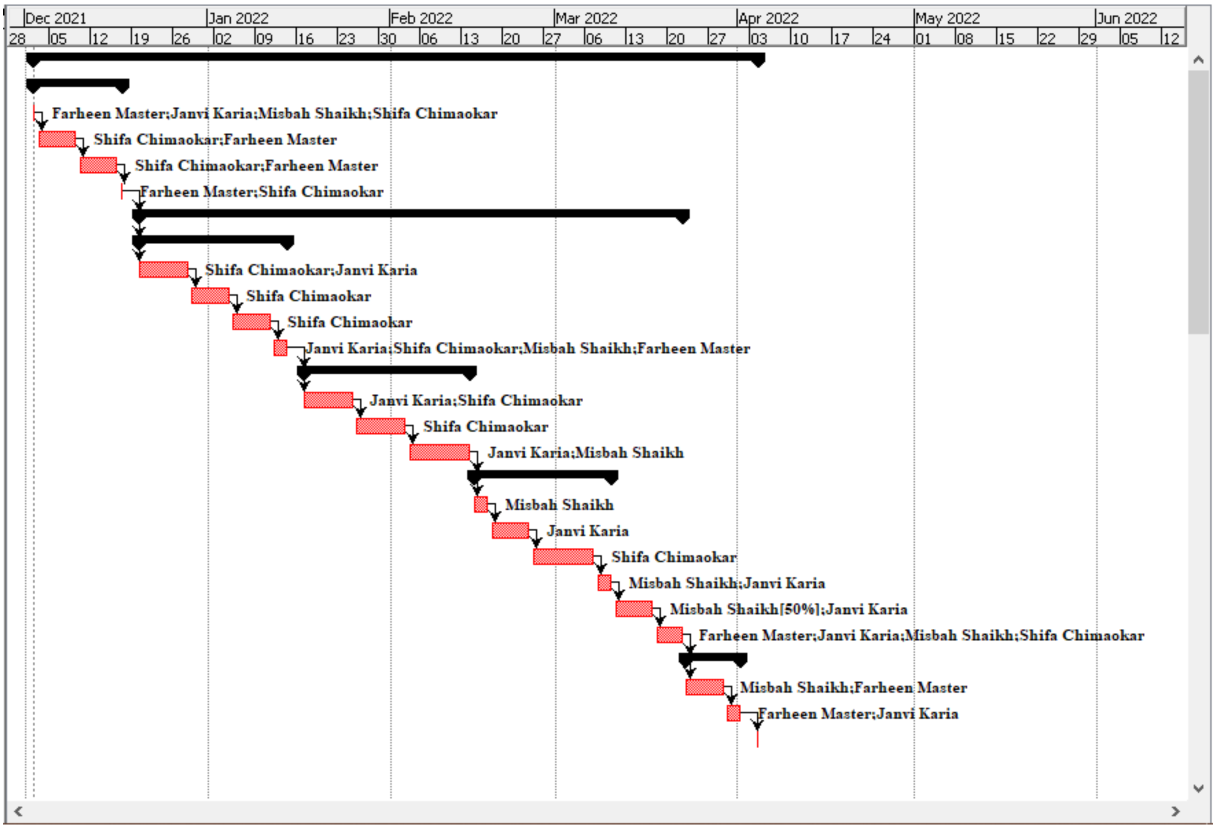


Figure 3.2: Project Timeline Gantt Chart

Chapter 4

Software Requirements Specification

4.1 Overall Description

For our final year project we have built a Hand Movements Prediction System. This is a classification system to predict the various motions performed by the upper limb of any user. This classification system is built by training machine learning models on EEG data to predict movements like touching, grasping, lifting or replacing any object.

The motivation behind building this model is to provide disabled people with better assistive prosthetics which may provide them with a greater sense of control on the limb and subsequently their own lives.

Our choice to use EEG data for this purpose is informed by the fact that EEG records brain activity and is completely safe to use. Analysing this brain activity and its relation to motor function will help us detect patterns and be able to predict the anticipated action. This prediction can help in building prosthetic devices that would give better command over their use.

4.1.1 Product Perspective

Due to the advent in robot control systems many studies have been conducted on EEG-based BCI for a robotic arm, the summary of a few of which has been illustrated in the table below.(3)

Table 4.1: Summary of EEG-based BCI robotic arm studies(3)

References	No. of subjects	Control signal	EEG feature	Classification algorithm	Performance evaluation
Yang C. et al. (2018)	2	SSVEP	FFT	CCA	Five tasks performed
Hortal et al. (2015)	2	MI	FFT	SVM	CA: 70%
Bousseta et al. (2018)	4	MI	FFT	RBF SVM	Success rate: 85.45%
Roy et al. (2016)	5	MI	WT, PSD	SVM	CA: 75.77%

4.1.2 Product Features

The following are the main features of our system :

- Our system is deployed on the internet which makes it easily accessible.
- Our product is able to make multi-label predictions.
- Our product also gives a good accuracy.
- The user will be able to use our model to predict hand movements using his/her EEG recording.
- The user will be educated about EEG and its relation to motor function.

4.1.3 User Classes and Characteristics

This project is made for any user that wishes to be able to predict hand movement artefacts in a recording of their EEG. Our project has only one kind of user. To be able to use our product the user must have relevant EEG data which is already sampled and stored in a csv file. The user has full access to our entire website and no user information is being stored.

4.1.4 Operating Environment

The software and hardware requirements that formed our operating environment for developing our project are listed below :

- **Python :**
- **Important Libraries :**
  - **Numpy & Pandas** for data handling and processing

  - **Matplotlib** for data visualisation
  - **PyWavelets** for signal processing
  - **scikit-learn** for machine learning

- **IDE :**
  - **Jupyter Lab**

- **RAM :** RAM is needed for your system to be able to handle large amounts of data. While 8 GB would suffice, 16 GB will make the process much smoother as well as faster.

- **Processor :** intel i5 or higher or amd ryzen 5 or higher.

- **Cores :** dual or quad core (More the number of cores, higher is the degree of parallelism that is achieved and hence, faster the operations can be performed.)

- **Storage :** SSD is very important while working with huge datasets such as the EEG dataset.

4.1.5 Design and Implementation Constraints

- This system can only work when the data provided to it is relevant both in the contents and its format(csv).

- Irrelevant data may lead to erroneous or no prediction.

- Due constraints on time and the lack of expertise in the field of robotics we will not be able to implement a brain-computer interface and are only limited to a deploying our model on a website using Django.

4.2 System Features

- **Django :** Django is the best framework for web applications, as it allows developers to use modules for faster development. As a developer, you can make use of these modules to create apps, websites from an existing source. It speeds up the development process greatly, as you do not have to code everything from scratch.

- **HTML, CSS, JS, Bootstrap :** HTML, CSS, Bootstrap and JavaScript are the backbone of a website.HTML provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript. CSS is used to control presentation, formatting, and layout. JavaScript is used to control the behavior of different elements. Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional

JavaScript plugins. Bootstrap also gives you the ability to easily create responsive designs.

- **Python :** Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances. Python is highly productive as compared to other programming languages like C++ and Java. ... Python is also very famous for its simple programming syntax, code readability and English-like commands that make coding in Python lot easier and efficient.

- **Discrete Wavelet Transform :** The DWT is a signal processing method that we have employed for denoising our EEG signals as well as extracting key features from them which were then used to classify the signals.

4.3 External Interface Requirements

4.3.1 User Interfaces

Our Model is deployed on a website for anyone to access online. The front-end is built in HTML, CSS, JS & Bootstrap and the back-end uses the Python Django framework.

4.3.2 Hardware Interfaces

- Laptop or Desktop

- Device should support internet connectivity

- *Optional* An EEG cap to record brain activity.

4.3.3 Software Interfaces

- Any Operating System (Windows / MacOs / Ubuntu)

- A browser supporting HTML5 and JS

- A stable internet connection.

4.3.4 Communications Interfaces

No explicit communication interface is required to interact with our system. The user can simply navigate to our website and access it through a browser.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

For building our website we have used the Django web application framework. It is great for our use-case as our website is simple. Also it provides excellent support for machine learning as it is a Python framework and our machine learning module is also created in Python.

4.4.2 Safety Requirements

For the purpose of recording the electrical activity from the user's scalp, an EEG cap is used. EEG is a completely safe, painless and harmless technique as it is recorded from the surface and is non-invasive unlike MRIs or fMRIs.

4.4.3 Security Requirements

There are no explicit security requirements for our project as it does not store the user's data or EEG files but only uses it to feed to the model to get a prediction.

Chapter 5

System Design

5.1 System Requirements Definition

5.1.1 Functional requirements

Our system should offer the following required services for each component :

Front-end The front-end should be visually pleasing and user friendly with ease of navigation.

Back-end The back-end should sufficiently connect the machine learning model to the front-end

Data Preprocessing The data should be properly cleaned and irrelevant data be dropped.

Signal Processing The signals should be denoised and sampled properly.

Model The model(s) should provide ample accuracy of prediction.

5.1.2 System requirements (non-functional requirements)

The following is the specification for non-functional requirements that our system must meet :

Accessibility The system is easily accessible given a good internet connection.

Reliability The system is reliable and will exhibit expected behaviour.

Security Since the system does not record personal user data, there is no vulnerability associated with it.

Time Efficiency Barring an unavoidable latency the execution time is significantly low.

Performance The system will work robustly under daily use.

5.2 System Architecture Design

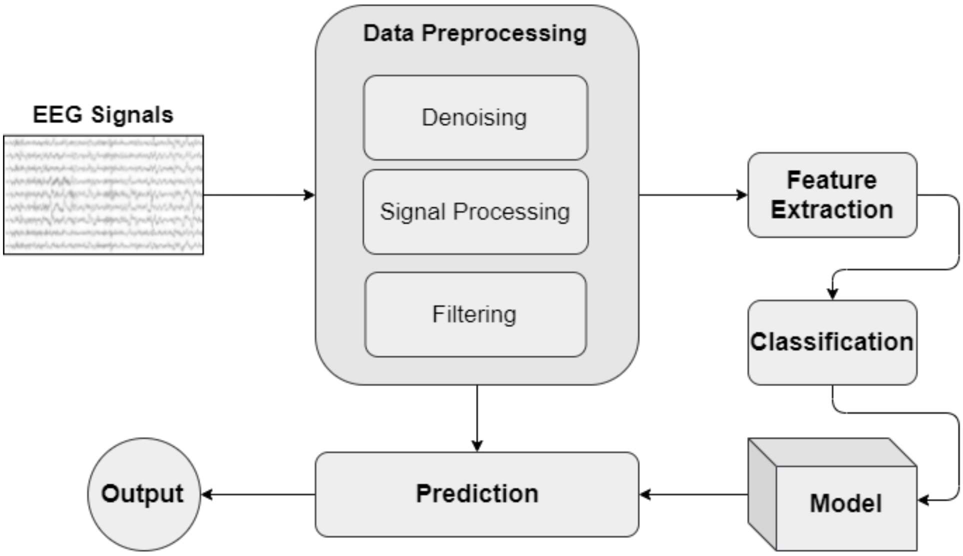


Figure 5.1: System Architecture

The user can access our model through the services page on our website. As an input we will be taking the sampled EEG signals through our web interface, this input will then be fed to our data preprocessing unit where techniques like denoising, signal processing and filtering will be performed on our data so as to get clean data. The important features will be extracted from this data and used for training a classification model. This model will then be tested for accuracy. After finalising, our model will be used to predict the various motions of the upper limb as output using the data from the user.

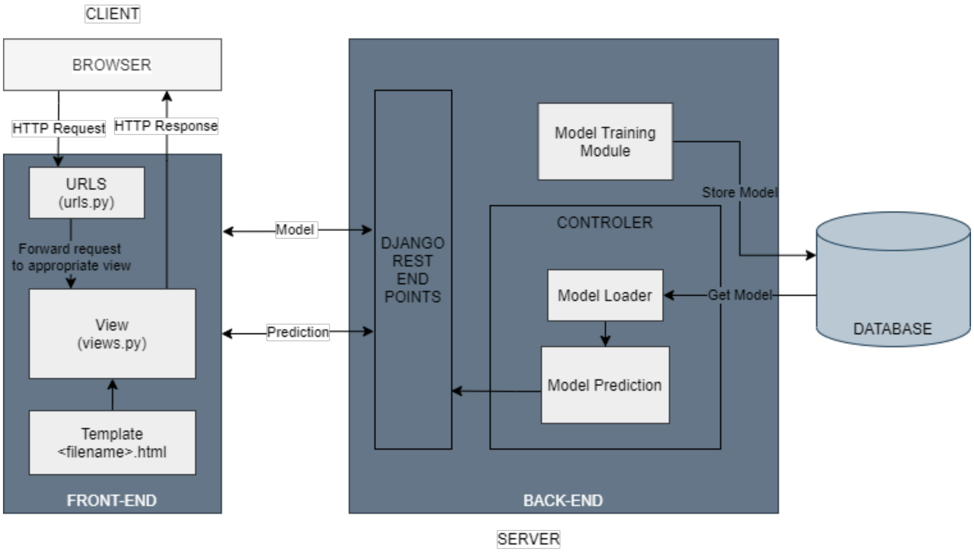


Figure 5.2: Framework of Proposed System

Framework selection is an important part of building any project. After the creation of our classification model, we plan to deploy it as a web application using python's Django framework so as to make it more dynamic.

The Django framework is a high- level web framework based on the model-views-controller (MVC) architecture. It helps in quick, easy and orderly development of web applications without having to worry about the structure and hassle involved in the integration of front-end and back-end.(9)
Since our system sits on top of the Django framework, we can understand it in terms of the following modules:

- **Controller Module :** This involves two processes - Training & Inference. The training process is responsible for training as well as testing and finally generating the model. The inference engine is responsible for loading the model and using it to output a prediction.

- **Views Module :** This module is responsible for handling Request/Response from the client browser and loading appropriate views with the help of templates. It is the front-end part of the system. Django supports the use of HTML, CSS, JS as well as Bootstrap for designing the web - UI.

- **Model Management Module :** This is the most important module and is responsible for handling the classification model generated by the controller module. This involves storage, retrieval, updation and deletion. Django uses object-relational mapping (ORM) for coordinating with the database to store, load and manage the models.

Other processes involve data cleaning and signal processing and happen before the training / model generation process. Python also provides exception libraries and packages like pandas, SciPy etc., and TensorFlow etc. respectively for data preparation, signal processing and machine learning tasks.

5.3 Sub-system Development

Our System can be divided into the following sub-systems:

5.3.1 Module 1 : Front-end Development Module

The front-end of our project was developed using HTML, CSS, JavaScript and Bootstrap.This module was used to build the front-end of the project which is the GUI of our website where the user can visit and interact with our project.

5.3.2 Module 2 : Denoising and Feature Extraction Module

The denoising and feature extraction module are responsible for data preprocessing and extracting features from that data.

There are two options for denoising signals : *The Fourier Transform* or *The Wavelet Transform*. The Fourier Transform works well for stationary signals, but since EEG signals are changing with time and on performing various tasks, we want to use the wavelet transform as it will give us information about the signals localised in time as well as frequency domain.

The Wavelet Transform works by taking a wavelet and performing several scaling and shifting operations on our signal with it. This results in the signal being separated into low pass and high pass sub-bands. We can use this to denoise the EEG data by passing it through these filters.

Similarly it can also be used to extract features from the signal by recursively decomposing it till we obtain the required frequencies (features).

5.3.3 Module 3 : Model Training and Back-end Development Module

The data along with features were then used to train decision tree classifiers on each label separately. These separate predictions were then combined to give the final prediction of our system.

The back-end was developed in Python using Django framework. It is a very efficient framework for machine learning tasks like ours and it is also completely open-sourced which provides an ease of debugging. Since the back-end was in python it was very easy ti integrate the various modules which were also built using python.

Imports

In [1]:
```
import pywt

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from math import sqrt,log10
from sklearn.metrics import mean_squared_error

import warnings
warnings.filterwarnings('ignore')
```

Denoising EEG Signals using Discrete Wavelet Transform

Steps to denoise signals :

1. Perform multilevel wavelet decomposition using the wavedec() function from PyWavelets module.
2. Select a thresholding technique.
3. Apply thresholding and reconstruct the signal using the waverec() function from the PyWavelets module.

The Discrete Wavelet Transform splits the input signal into low pass and high pass subbands. The low pass subband is also called the *approximation coefficient* and the high pass subband is also called the *detail coefficient* . The approximation coefficient can be further split up on multiple levels. This is known as *Multilevel Wavelet Decomposition*.

Once we perform the multilevel wavelet decomposition we want to select a suitable thresholding technique. For our purposes we will use the universal thresholding technique as it is very convienient and the universal threshold is easy to compute.

$$\text{Universal Threshold} = \frac{\left(\sqrt{2\log(\text{length}(\text{X}))}\right)\text{median}(\text{abs}(\text{D}))}{0.6745}$$

where,

- **X** is the signal
- **D** is the set of first level detail coefficients

Using the Universal threshold we will apply hard thresholding to the signal, which is a type of thresholding technique where all coefficients below the calculated threshold are reduced to 0 and all the coefficients above the threshold value are left unchanged.

After thresholding the signal is reconstructed using the waverec() function.

In [2]:
```
def madev(d, axis=None):
    """ Median absolute deviation of a signal """
    return np.median(np.absolute(d))

def wavelet_denoising(x):
    c = pywt.wavedec(x,"sym18", mode="per",level = 4)

    sigma = (1/0.6745) * madev(c[-1])

    univ_thresh = sigma * np.sqrt(2 * np.log(len(x)))
```

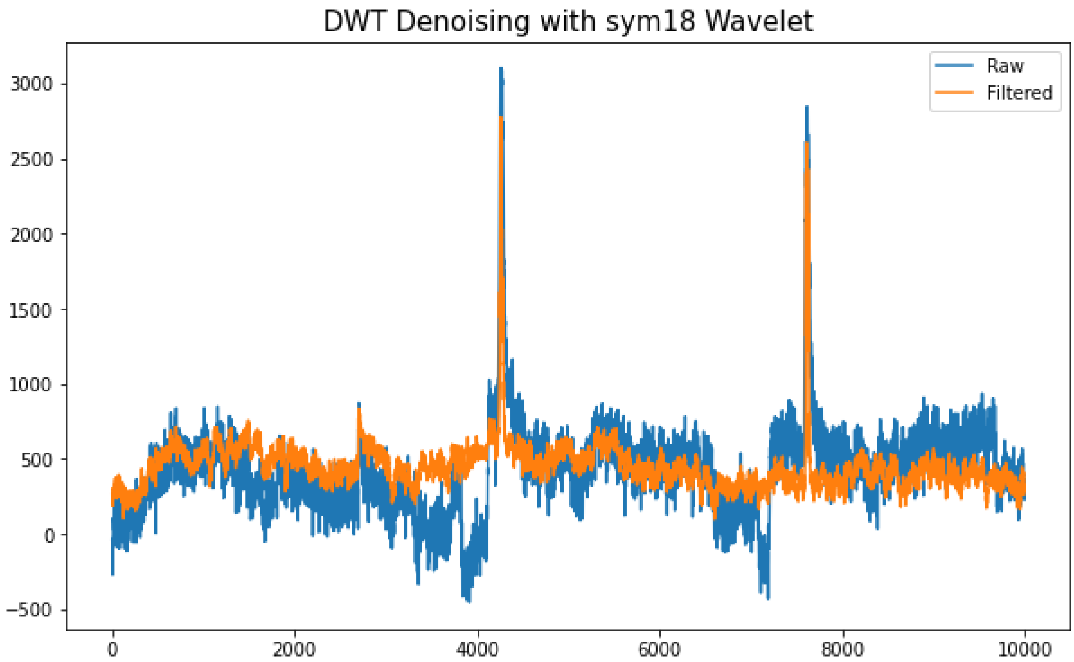
```
    c[1:] = (pywt.threshold(i, value=univ_thresh, mode='hard') for i in c[1:])

    return pywt.waverec(c, "sym18", mode='per')


signal = pd.read_csv('data/train/subj1_data.csv')
signal = signal.drop("id", axis=1)

filtered = pd.DataFrame(wavelet_denoising(signal))

###PLOT###
plt.figure(figsize=(10, 6))
plt.plot(signal.iloc[:10000,0], label='Raw')
plt.plot(filtered.iloc[:10000,0], label='Filtered')
plt.legend()
plt.title("DWT Denoising with sym18 Wavelet", size=15)
plt.show()
```



Feature Extraction of EEG Signals using Discrete Wavelet Transform

The wavedec() function performs 1D multilevel Discrete Wavelet Transform decomposition of a given signal and returns an ordered list of coefficients arrays in the form : [cA_n, cD_n, cD_n-1, ..., cD2, cD1]

where n denotes the level of decomposition. The first element (cA_n) of the result is the approximation coefficients array and the following elements (cD_n - cD_1) are detailed coefficients arrays.

Now coming to the point of different frequency bands

Discrete wavelet transform will always return only one approximation coefficient.

If starting frequency band of the eeg waveform is 0-64 Hz then at level =1 we will get 0-32 Hz which gives approximation coefficients & another band is 32-64Hz which gives the detail coefficient of the wavelet.

At level = 2, the discrete wavelet transform will return 3 frequency bands:

1. 0-16 Hz i.e. approximation coefficients
2. 16-32 Hz i.e. detail coefficients
3. 32-64 Hz i.e. detail coefficients. and so on

Since the sampling frequency of our eeg data is 500Hz, therefore by Nyquist-Shanon Theorem, the highest frequency in our waveform will be 250 Hz then at level =1 we will get 0-125 Hz which gives approximation coefficients & another band is 125-250Hz which gives the detail coefficient of the wavelet.

Therefore at level = 5, the discrete wavelet transform will return 3 frequency bands:

1. 0 - 7.8125 Hz i.e. approximation coefficients
2. 7.8125 - 15.625 Hz ~ Alpha Frequency Band
3. 15.625 -31.25 Hz ~ Beta Frequency Band
4. 31.25 - 62.5 Hz
5. 62.5 - 125 Hz
6. 125 - 250 Hz

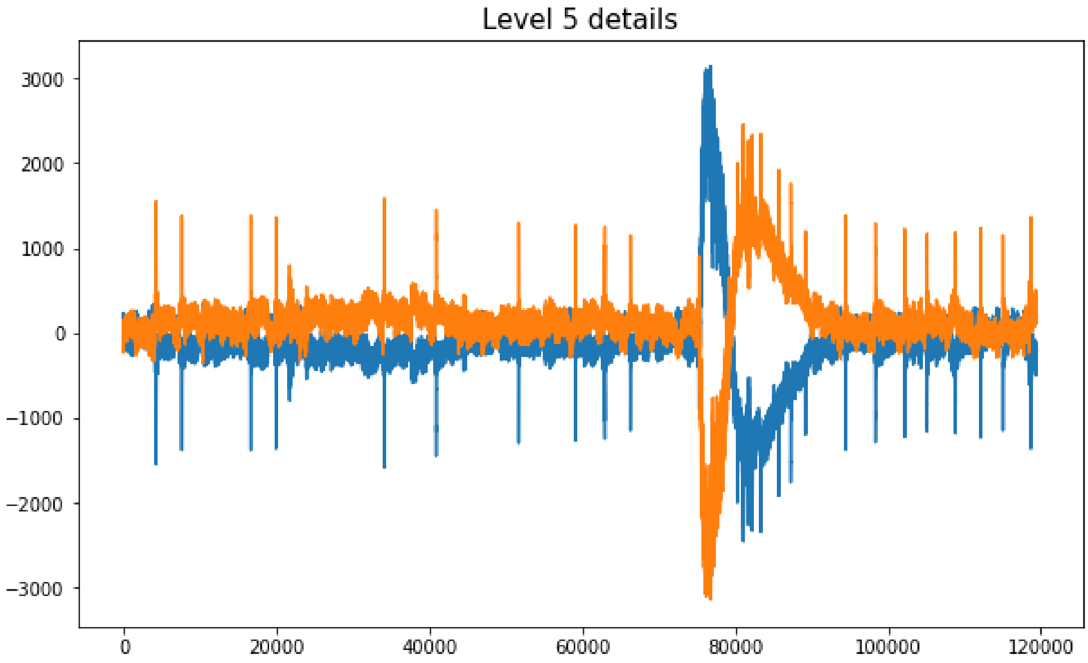
The aim is to extract Alpha and Beta frequencies as they correspond to motor function.

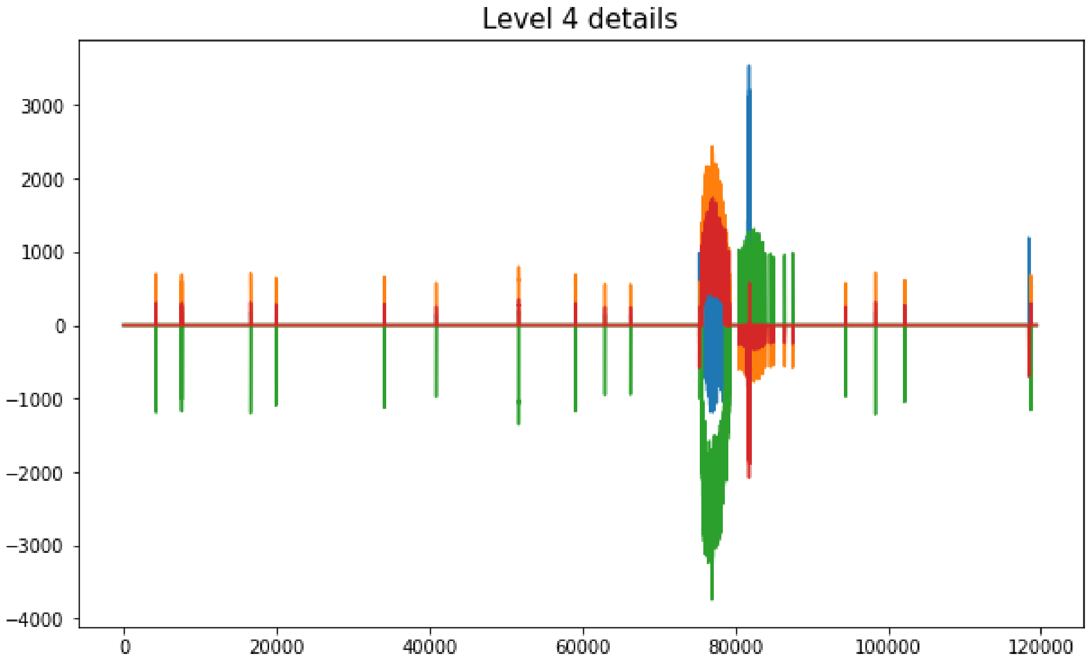
In [3]:
```
def FEdwt(s):
    coefli = pywt.wavedec(s,"sym18", mode="per", level=5)
    return coefli

cli = FEdwt(filtered)

i = 5
for c in cli[1:3:]:
    d = pd.DataFrame(pywt.idwt(None, c,"sym18", mode="per"))

    ###PLOT###
    plt.figure(figsize=(10, 6))
    plt.plot(d.iloc[:], label='')
    plt.title(f"Level {i} details", size=15)
    plt.show()
    i-=1
```





Level 4 details
3000
2000
1000
0
−1000
−2000
−3000
−4000
0
20000
40000
60000
80000
100000
120000

MODEL TRAINING

Imports

In [1]:
```
# To use the wavelet decomposition function in the PyWavelets module
import pywt

# For data handling
import numpy as np
import pandas as pd

# To handle file path
from glob import glob

# To use mathematical functions
from math import sqrt,log10

# To save and load our model
import joblib

# To train a decision tree classification model
from sklearn.tree import DecisionTreeClassifier

# To ignore any warnings that may be prompted
import warnings
warnings.filterwarnings('ignore')
```

Denoising EEG Signals using Discrete Wavelet Transform

In [2]:
```
def madev(d, axis=None):
    '''
    Median absolute deviation of a signal
    '''
    return np.median(np.absolute(d))


def wavelet_denoising(x):
    '''
    Function to denoise the EEG signals using the discrete wavelet transform.
    '''
    # Using wavelet decomposition to apply DWT on 4 levels
    c = pywt.wavedec(x,"sym18", mode="per",level = 4)

    # Calculation for universal threshold
    sigma = (1/0.6745) * madev(c[-1])
    univ_thresh = sigma * np.sqrt(2 * np.log(len(x)))

    # Applying hard thresholding using the universal threshold
    # calculated in the previous step
    c[1:] = (pywt.threshold(i, value=univ_thresh, mode='hard') for i in c[1:])

    return pywt.waverec(c, "sym18", mode='per')
```

Feature Extraction of EEG Signals using Discrete Wavelet Transform

In [3]:
```
def FEdwt(s):
    '''
    Function to extract features (namely information relating to the alpha(8-12Hz) and
    beta(13-30Hz) frequency bands that corespond to the performance of motor functions.)
    from EEG signals.
```

```
    '''
    # Using wavelet decomposition to apply DWT on 5 levels
    # to get frequencies upto approx. the alpha band
    coefli = pywt.wavedec(s,"sym18", mode="per", level=5)

    # Making a list of features and appending the original signal
    features = []
    features.append(s)

    # Appending only the decompositions coresponding to the alpha and beta bands
    for c in coefli[1:3:]:
        features.append(pd.DataFrame(pywt.idwt(None, c,"sym18", mode="per")))

    # Final dataframe of signal and features extracted from it
    featr = pd.concat([feature for feature in features])
    return featr
```

Data PreProcessing

Train - Test split : 75% - 25%

In [4]:
```
# Get the file path to all files present in the train and test folders
train_fnames =  glob('data/train/subj*_data.csv')
test_fnames =  glob('data/test/subj*_data.csv')
```

In [5]:
```
def preprocess_data(fname):

    # Store data in a dataframe and drop the id column
    datax = pd.read_csv(fname)
    datax.drop(['id'], axis = 1, inplace=True)

    # Denoise the signal and re-attach the header to the result
    cols = datax.columns

    datax = pd.DataFrame(wavelet_denoising(datax))
    datax.columns = cols

    # Get output filename from data filename
    events = fname.replace('_data','_events')
    datay = pd.read_csv(events)

    # Drop the id column
    datay.drop(['id'], axis = 1, inplace=True)

    # Channel Selection to reduce dimensionality and preserve raw data
    # Drop all channels that are away from the central lobe
    datax.drop([x for x in datax.columns if 'C' not in x], axis = 1, inplace=True)

    # Concatenating all labels to end of the data
    for col in datay.columns:
        datax[f'{col}_output'] = datay[col]

    return datax
```

In [6]:
```
# Concatenating all training data
train_data = pd.concat([preprocess_data(fname) for fname in train_fnames])
train_data
```

Out[6]:

	FC5	FC1	FC2	FC6	C3	Cz	C4	CP5	
0	292.183972	300.131051	308.161367	315.968030	329.734439	335.162066	339.322013	342.841195	340.8
1	288.927679	296.780747	304.716068	312.430381	326.033940	331.397360	335.508097	338.985647	337.0
2	271.703353	279.387597	287.152326	294.700799	308.011911	313.260029	317.282394	320.685185	318.7
3	249.053309	260.469669	272.005604	283.220250	302.996359	310.793405	316.769369	321.824842	319.0
4	249.304860	265.001491	280.862528	296.281817	323.472469	334.192817	342.409316	349.360209	345.4
...	...	...	...	...	...	...	...	...	
141421	81.553617	55.054955	28.278747	2.248287	-43.654296	-61.752120	-75.623009	-87.357333	-80.8
141422	107.748201	81.887391	55.755716	30.351838	-14.445819	-32.108009	-45.645010	-57.096876	-50.7
141423	120.161328	96.299704	72.188155	48.748139	7.413594	-8.883209	-21.373723	-31.940296	-26.0
141424	130.386958	104.762738	78.870133	53.698665	9.310845	-8.189760	-21.602916	-32.950013	-26.6
141425	149.150528	125.297457	101.194552	77.762938	36.443210	20.152249	7.666212	-2.896573	2.9

1807279 rows × 17 columns

In [7]:

```
# Concatenating all testing data
test_data = pd.concat([preprocess_data(fname) for fname in test_fnames])
test_data
```

Out[7]:

	FC5	FC1	FC2	FC6	C3	Cz	C4	CP5
0	-373.220094	-380.495103	-387.846311	-394.992779	-407.594989	-412.563611	-416.371760	-419.593331
1	-376.130104	-383.550627	-391.048872	-398.338283	-411.192561	-416.260565	-420.144883	-423.430891
2	-386.955898	-393.351202	-399.813490	-406.095796	-417.174128	-421.541937	-424.889598	-427.721612
3	-397.889565	-403.693940	-409.559108	-415.260926	-425.315613	-429.279833	-432.318169	-434.888503
4	-379.715488	-385.088507	-390.517802	-395.795886	-405.103353	-408.772971	-411.585510	-413.964828
...	...	...	...	...	...	...	...	...
151118	123.425156	135.066750	146.830278	158.266179	178.432451	186.383325	192.477190	197.632402
151119	138.938076	144.035821	149.186960	154.194634	163.025256	166.506870	169.175316	171.432735
151120	137.225714	138.581369	139.951223	141.282926	143.631274	144.557147	145.266773	145.867094
151121	124.134817	124.646762	125.164069	125.666968	126.553790	126.903434	127.171415	127.398118
151122	116.835845	113.426805	109.982060	106.633255	100.727911	98.399635	96.615153	95.105538

636910 rows × 17 columns

Training the Model on Train Data

AIM :: To make multi-label prediction

To acheive multi-label prediction we train a binary classification model on our data seperately for each label and concatenate the predictions while giving the output.

Training Decision Tree Classifier on the label HandStart

In [8]:

```
clf1 = DecisionTreeClassifier()
clf1.fit(train_data.iloc[:,:11] , train_data.iloc[:,11])
```

Out[8]: DecisionTreeClassifier()

Training Decision Tree Classifier on the label FirstDigitTouch

In [9]:
```
clf2 = DecisionTreeClassifier()
clf2.fit(train_data.iloc[:,:11] , train_data.iloc[:,12])
```

Out[9]: DecisionTreeClassifier()

Training Decision Tree Classifier on the label BothStartLoadPhase

In [10]:
```
clf3 = DecisionTreeClassifier()
clf3.fit(train_data.iloc[:,:11] , train_data.iloc[:,13])
```

Out[10]: DecisionTreeClassifier()

Training Decision Tree Classifier on the label LiftOff

In [11]:
```
clf4 = DecisionTreeClassifier()
clf4.fit(train_data.iloc[:,:11] , train_data.iloc[:,14])
```

Out[11]: DecisionTreeClassifier()

Training Decision Tree Classifier on the label Replace

In [12]:
```
clf5 = DecisionTreeClassifier()
clf5.fit(train_data.iloc[:,:11] , train_data.iloc[:,15])
```

Out[12]: DecisionTreeClassifier()

Training Decision Tree Classifier on the label BothReleased

In [13]:
```
clf6 = DecisionTreeClassifier()
clf6.fit(train_data.iloc[:,:11] , train_data.iloc[:,16])
```

Out[13]: DecisionTreeClassifier()

Accuracy Scores for each Label

WARNING :: The high accuracy only applies to each label seperately

In [14]:
```
score1 = clf1.score(test_data.iloc[:,:11], test_data.iloc[:,11]) * 100
print("HandStart Acc          :: %.2f" % score1,"%")

score2 = clf2.score(test_data.iloc[:,:11], test_data.iloc[:,12]) * 100
print("FirstDigitTouch Acc    :: %.2f" % score2,"%")

score3 = clf3.score(test_data.iloc[:,:11], test_data.iloc[:,13]) * 100
print("BothStartLoadPhase Acc :: %.2f" % score3,"%")

score4 = clf4.score(test_data.iloc[:,:11], test_data.iloc[:,14]) * 100
print("LiftOff Acc            :: %.2f" % score4,"%")


score5 = clf5.score(test_data.iloc[:,:11], test_data.iloc[:,15]) * 100
print("Replace Acc            :: %.2f" % score5,"%")


score6 = clf6.score(test_data.iloc[:,:11], test_data.iloc[:,16]) * 100
print("BothReleased Acc       :: %.2f" % score6,"%")
```

```
HandStart Acc          :: 95.53 %
FirstDigitTouch Acc    :: 95.36 %
BothStartLoadPhase Acc :: 95.36 %
LiftOff Acc            :: 95.23 %
Replace Acc            :: 95.01 %
BothReleased Acc       :: 95.10 %
```

Saving models to pickle files using Joblib

In [15]:

```
joblib.dump(clf1, 'pickle1.pkl')
joblib.dump(clf2, 'pickle2.pkl')
joblib.dump(clf3, 'pickle3.pkl')
joblib.dump(clf4, 'pickle4.pkl')
joblib.dump(clf5, 'pickle5.pkl')
joblib.dump(clf6, 'pickle6.pkl')
```

Out[15]:

```
['pickle6.pkl']
```

5.4 Systems Integration

System integration is an extremely crucial step in the design of any system as the result of this will be one cohesive infrastructure consisting of the various modules all integrated together.

5.4.1 Class Diagram

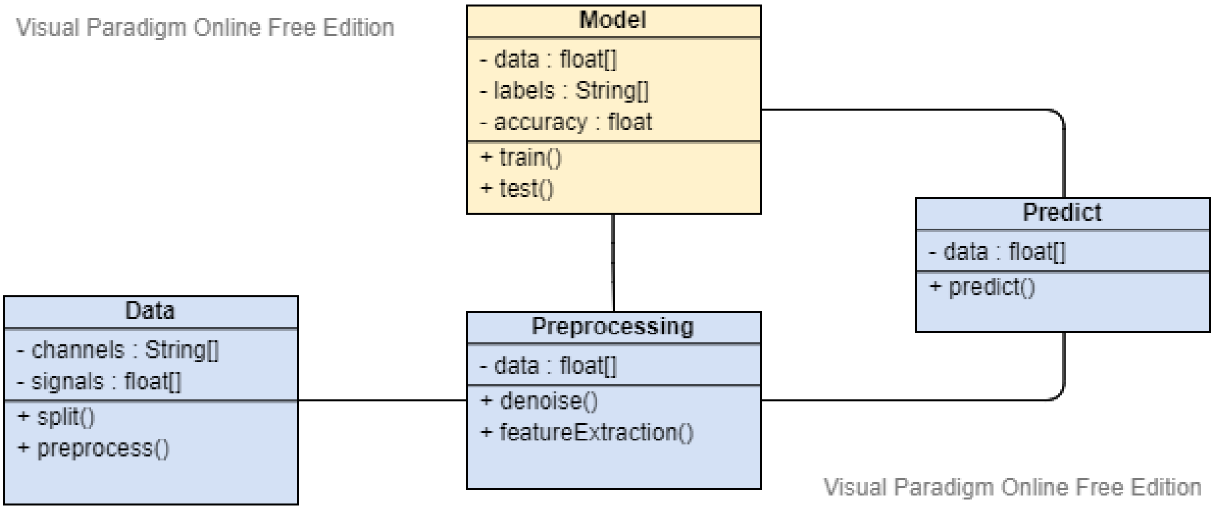


Figure 5.3: Class Diagram of System

5.4.2 Sequence Diagram

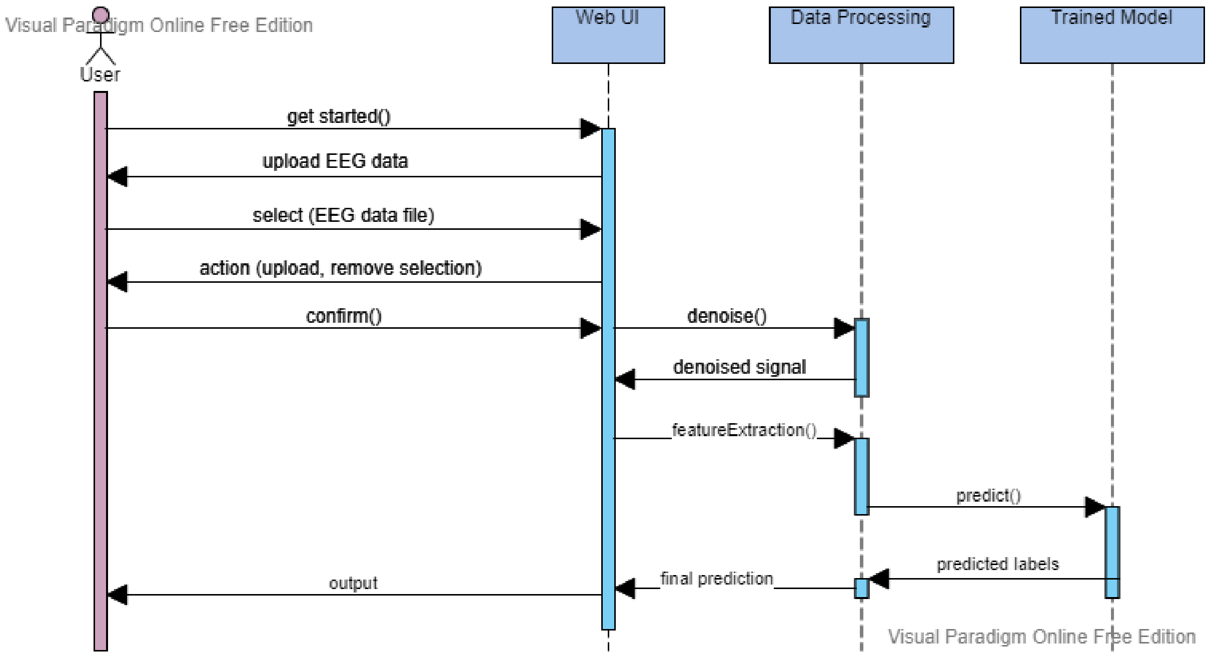


Figure 5.4: Sequence Diagram of System

Chapter 6

Implementation

6.1 The Manage.py File

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'helpinghands.
                                                    settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you
                                                       "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

6.2 The Views.py File

```
from django.shortcuts import render
from django.http import HttpResponse
from django.shortcuts import render
from django.conf import settings
from django.core.files.storage import FileSystemStorage
from django.core.files.storage import default_storage
```

```
from datetime import datetime
import csv
import pywt
import numpy as np
import pandas as pd
from math import sqrt,log10
from sklearn.tree import DecisionTreeClassifier
import joblib

def madev(d, axis=None):
    """ Median absolute deviation of a signal """
    return np.median(np.absolute(d))

def wavelet_denoising(x):
    c = pywt.wavedec(x,"sym18", mode="per",level = 4)

    sigma = (1/0.6745) * madev(c[-1])

    univ_thresh = sigma * np.sqrt(2 * np.log(len(x)))

    c[1:] = (pywt.threshold(i, value=univ_thresh, mode='hard') for i in
                                                        c[1:])

    return pywt.waverec(c, "sym18", mode='per')


def index(request):
    return render(request,"index.html")

def mainpage(request):
    return render(request,"mainpage.html")

def page(request):
    return render(request,"page.html")

def upload_csv(request):
    context = {}
    if request.method == 'POST':
        print(request.FILES['file'] )
        file_name = default_storage.save(datetime.now().strftime("%Y-%m-
                                                %d-%H-%M-%S-")+str(request.
                                                FILES['file'].name), request
                                                .FILES['file'])
        print("media/"+file_name)
        datax = pd.read_csv("media/"+file_name)
        datax.drop(['id'], axis = 1, inplace=True)
        cols = datax.columns
        datax = pd.DataFrame(wavelet_denoising(datax))
        datax.columns = cols
        datax.drop([x for x in datax.columns if 'C' not in x], axis = 1,
                                                 inplace=True)
        model = [
            joblib.load('pickle/pickle1.pkl'),
            joblib.load('pickle/pickle2.pkl'),
            joblib.load('pickle/pickle3.pkl'),
            joblib.load('pickle/pickle4.pkl'),
            joblib.load('pickle/pickle5.pkl'),
            joblib.load('pickle/pickle6.pkl'),
        ]
```

```
    results = list()
    for i in range(6):
        results.append(model[i].predict(datax))

    t_results = zip(*results)

    default_storage.delete(file_name)

    sentences = [
        "Subject has started  i t s  arm to touch object.",
        "Subject has touched the object.",
        "Subject has grasped the object.",
        "Subject has lifted the object.",
        "Subject has replaced the object back.",
        "Subject has released the object from  i t s  grasp.",
        "No changes."
        ]

    rows = []
    mat = []
    li = []
    count = 0
    for i in t_results:
        if count % 10 == 0:
            t_matrix = zip(*rows)
            for row in t_matrix:
                mat.append(list(row))

            temp = []
            string = ""
            flag = 1
            c = 0
            for j in mat:

                if 1 in j:

                    string += sentences[c]
                    flag = 0
                else:
                    pass
                c+=1
            if flag:
                string += sentences[-1]

            li.append(string)
            rows = []
            mat = []
            rows.append(i[1:])
        else:
            rows.append(i[1:])

        count += 1


    t_matrix = zip(*rows)
    for row in t_matrix:
        mat.append(list(row))

    temp = []
    string = ""
```

```
    flag = 1
    c = 0
    for j in mat:
        if 1 in j:

            string += sentences[c]
            flag = 0
        else:
            pass
        c+=1
    if flag:
        string += sentences[-1]

    li.append(string)
    rows = []
    mat = []
    print(len(li))

    li = li[1:]

    context = {
        'output' : li,

    }


    return render(request,"mainpage.html",context)

return render(request,"mainpage.html",context)
```

6.3 The MainPage.html File

```
<!DOCTYPE html>
<html lang="en">

<head>
  {% load static %}
  <title>Helping Hands</title>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="{% static 'assets/img/favicon.png' %}" rel="icon">
  <link href="{% static 'assets/img/apple-touch-icon.png' %}" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Krub:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{% static 'assets/vendor/aos/aos.css' %}" rel="stylesheet">
```

```
  <link href="{% static 'assets/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap-icons/bootstrap-icons.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/boxicons/css/boxicons.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/glightbox/css/glightbox.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/swiper/swiper-bundle.min.css' %}" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="{% static 'assets/css/style.css' %}" rel="stylesheet">

  <script type="text/javascript">
    function readURL(input) {
  if (input.files && input.files[0]) {

    var reader = new FileReader();

    reader.onload = function(e) {
      $('.image-upload-wrap').hide();

      $('.file-upload-image').attr('src', e.target.result);
      $('.file-upload-content').show();

      $('.image-title').html(input.files[0].name);
    };

    reader.readAsDataURL(input.files[0]);

  } else {
    removeUpload();
  }
}

function removeUpload() {
  $('.file-upload-input').replaceWith($('.file-upload-input').clone());
  $('.file-upload-content').hide();
  $('.image-upload-wrap').show();
}
$('.image-upload-wrap').bind('dragover', function () {
    $('.image-upload-wrap').addClass('image-dropping');
  });
  $('.image-upload-wrap').bind('dragleave', function () {
    $('.image-upload-wrap').removeClass('image-dropping');
});
  </script>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>services</title>
  <meta content="" name="description">

  <meta content="" name="keywords">
<style type="text/css">
  body {
  font-family: sans-serif;
```

```
  background-color: #eeeeee;
}

.file-upload {
  background-color: #ffffff;
  width: 600px;
  margin: 0 auto;
  padding: 20px;
}

.file-upload-btn {
  width: 100%;
  margin: 0;
  color: #fff;
  background: #4154F1;
  border: none;
  padding: 10px;
  border-radius: 4px;
  border-bottom: 4px solid #012970;
  transition: all .2s ease;
  outline: none;
  text-transform: uppercase;
  font-weight: 700;
}

.file-upload-btn:hover {
  background: #4FC3F7  ;
  color: #ffffff;
  transition: all .2s ease;
  cursor: pointer;
}

.file-upload-btn:active {
  border: 0;
  transition: all .2s ease;
}

.file-upload-content {
  display: none;
  text-align: center;
}

.file-upload-input {
  position: absolute;
  margin: 0;
  padding: 0;
  width: 100%;
  height: 100%;
  outline: none;
  opacity: 0;
  cursor: pointer;
}

.image-upload-wrap {
  margin-top: 20px;
  border: 4px dashed #012970;
  position: relative;
}
.output-wrap {
  margin-top: 20px;
```

```
  border: 4px solid #012970;
  position: relative;
}
.image-dropping,
.image-upload-wrap:hover {
  background-color: #4FC3F7 ;
  border: 4px dashed #ffffff;
}

.image-title-wrap {
  padding: 0 15px 15px 15px;
  color: #222;
}

.drag-text {
  text-align: center;
}

.drag-text h3 {
  font-weight: 100;
  text-transform: uppercase;
  color: #4154F1;
  padding: 60px 0;
}
.output-text {
  background-color: rgb(12, 59, 59);
  text-align: center;
}

.output-text h3 {
  font-weight: 100;
  text-transform: uppercase;
  color: #d6d7dc;
  padding: 60px 0;
}
.file-upload-image {
  max-height: 200px;
  max-width: 200px;
  margin: auto;
  padding: 20px;
}

.remove-image {
  width: 200px;
  margin: 0;
  color: #fff;
  background: #1A237E  ;
  border: none;
  padding: 10px;
  border-radius: 4px;
  border-bottom: 4px solid #1A237E ;
  transition: all .2s ease;
  outline: none;
  text-transform: uppercase;
  font-weight: 700;
}

.remove-image:hover {
  background: #0D47A1 ;
  color: #ffffff;
```

```
  transition: all .2s ease;
  cursor: pointer;
}

.remove-image:active {
  border: 0;
  transition: all .2s ease;
}
</style>
  <!-- Favicons -->
  <link href="{% static 'assets/img/favicon.png' %}" rel="icon">
  <link href="{% static 'assets/img/apple-touch-icon.png' %}" rel="
      apple-touch-icon">

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300
      ,300i,400,400i,600,600i,700,700i|Nunito:300,300i,400,400i
      ,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i
      ,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{% static 'assets/vendor/bootstrap/css/
      bootstrap.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap-icons/
      bootstrap-icons.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/aos/aos.css' %}" rel="
      stylesheet">
  <link href="{% static 'assets/vendor/remixicon/remixicon.css' %}"
      rel="stylesheet">
  <link href="{% static 'assets/vendor/swiper/swiper-bundle.min.css'
       %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/glightbox/css/
      glightbox.min.css' %}" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="{% static 'assets/css/style.css' %}" rel="stylesheet">
</head>

<body>

  <!-- ======= Header ======= -->
  <header id="header" class="fixed-top">
    <div class="container d-flex align-items-center
        justify-content-between">

      <a href="index.html" class="logo"><img src="{% static 'assets/
          img/logo.png' %}" alt="" class="img-fluid">Helping Hands</a
          >
      <!-- <h1 class="logo"><a href="index.html">EEGLAB</a></h1> -->


      <nav id="navbar" class="navbar">
        <ul>

        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
      </nav><!-- .navbar -->

    </div>
  </header><!-- End Header -->
```

```
  <!-- form section -->

 <br>
  <br>
<section id="service">
<div class="container">
    <h1 style="text-align: center;">Upload Your EEG Data</h1>
    <p style="text-align: center;">(In the form of CSV file)</p>
<script class="jsbin" src="https://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<div class="file-upload">
  <button class="file-upload-btn" type="button" onclick="$('.file-upload-input').trigger( 'click' )">Upload EEG Data</button>
  <form method="post" enctype="multipart/form-data" action="{% url 'upload_csv' %}">
    {% csrf_token %}
  <div class="image-upload-wrap">
    <input class="file-upload-input" type='file' onchange="readURL(this);" name="file" accept=".csv" />
    <div class="drag-text">
      <h3>Drag and drop your CSV File</h3>
    </div>
  </div>
  <div class="container pt-3">
  <div class="file-upload-content">
    <div class="image-title-wrap">
      <button type="button" onclick="removeUpload()" class="remove-image col-12 responsive-width">Remove <span class="image-title">Uploaded File</span></button><br><br>
       <input class="remove-image" type="submit" value="Upload" />
        <div ng-show="progressVisible">
            <div class="percent"></div>
            <div class="progress-bar">
                <div class="uploaded" ng-style="{'width': progress+'%'}"></div>
            </div>
        </div>

    </div>
  </div>
  </div>
</form>
</div>
{% if output %}
<div>
  <div class="file-upload">
    <img class="file-upload-image" src="#" alt="" />
    <h1 style="text-align: center;">Output of your EEG Data</h1>
    <form method="post" enctype="multipart/form-data" action="{% url 'upload_csv' %}">
      {% csrf_token %}
    <div class="output-wrap">
      <div class="output-text">
        <p>
          <center><h3 name="out" id="out"></h3></center>
        </p>
      </div>

    </div>
```

```
  </form>

{% endif %}
</div>
</section>

<!-- End Form  -->

  <!-- ======= Footer ======= -->
  <footer id="footer">

    <div class="footer-top">
      <div class="container">
        <div class="row">

          <div class="col-lg-3 col-md-6 footer-contact">
            <h3>Helping Hands</h3>
            <p>
              Plot No. 2 & 3, Sector - 16,<br>
              Navi Mumbai, Maharashtra 410206
              India<br><br>

              <strong>Phone:</strong> +1 123 567 890<br>
              <strong>Email:</strong> eeglab@gmail.com<br>
            </p>
          </div>

          <div class="col-lg-2 col-md-6 footer-links">
            <h4>Useful Links</h4>
            <ul>
              <li><i class="bx bx-chevron-right"></i> <a href="#">Home</a></li>
              <li><i class="bx bx-chevron-right"></i> <a href="#aboutus">About us</a></li>
              <li><i class="bx bx-chevron-right"></i> <a href="#">Services</a></li>

          </div>

        </div>
      </div>
    </div>

    <div class="container d-md-flex py-4">

        </div>
      </div>
      <div class="social-links text-center text-md-right pt-3 pt-md-0">
        <a href="#" class="twitter"><i class="bx bxl-twitter"></i></a>
        <a href="#" class="facebook"><i class="bx bxl-facebook"></i></a>
        <a href="#" class="instagram"><i class="bx bxl-instagram"></i></a>
        <a href="#" class="google-plus"><i class="bx bxl-skype"></i></a>
        <a href="#" class="linkedin"><i class="bx bxl-linkedin"></i></a>
      </div>
```

```
      </div>
    </footer><!-- End Footer -->

    <div id="preloader"></div>
    <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

    <!-- Vendor JS Files -->
    <script src="{% static 'assets/vendor/aos/aos.js' %}"></script>
    <script src="{% static 'assets/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>
    <script src="{% static 'assets/vendor/glightbox/js/glightbox.min.js' %}"></script>
    <script src="{% static 'assets/vendor/isotope-layout/isotope.pkgd.min.js' %}"></script>
    <script src="{% static 'assets/vendor/php-email-form/validate.js' %}"></script>
    <script src="{% static 'assets/vendor/swiper/swiper-bundle.min.js' %}"></script>

    <!-- Template Main JS File -->
    <script src="{% static 'assets/js/main.js' %}"></script>
    <script>
      var text = {{ output | safe}};
      var counter = 0;
      var elem = document.getElementById("out");
      var inst = setInterval(change, 1500);

      function change() {
        elem.innerHTML = text[counter];
        counter++;

      }

    </script>

  </body>
  </html>
```

Chapter 7

System Testing

7.1 Test Cases and Test Results

Table 7.1: Test Cases for Hand Movements Prediction System

Test ID	Test Case Title	Expected Result	System Behaviour	Pass/ Fail
T01	Loading main landing page	Site should open	As Expected	Pass
T02	Open File Dialogue for CSV files only	File dialogue should contain only csv files	As Expected	Pass
T03	Remove File Button	Selected file should be removed	As Expected	Pass
T04	Upload Button	Output should be shown	As Expected	Pass
T05	Output Prediction	Accuracy $>$ 70 %	Accuracy $\approx$ 75 %	Pass

7.2 Sample of a Test Case

Title : Loading main landing page

Description : When we open our website the user should be able to land on the home page of our website.

Precondition : The user must have stable internet connectivity.

Assumption : A supported browser will be used.

Test Steps :

1. Navigate to http://aiktchelpinghands.herokuapp.com

Expected Result : User should be able to see the landing page

Actual Result : We are able to see the landing page of our website.

Title : Open File Dialogue for CSV files only

Description : Dialog boxes allow the user to navigate within their local directories to choose any file or folder for upload. Narrowing down the options in the dialog box by opening one dedicated to a specific type of file helps in locating the needed file faster.

Precondition : The user must have stable internet connectivity.

Assumption : A supported browser will be used.

Test Steps :

1. Navigate to http://aiktchelpinghands.herokuapp.com
2. Click on the *Get Started* button
3. Click on the *Upload EEG Data* button
4. A file dialog box should open where only csv files are visible

Expected Result : The file dialog box should only contain .csv files.

Actual Result : We can only select .csv files in the file navigator

Title : Remove File Button

Description : Sometimes the user might accidentally upload or select the wrong csv file, in this case the user should have the option to remove the selection and select the desired file. We have provided a *Remove File* button to serve this very purpose.

Precondition : The user must have stable internet connectivity.

Assumption : A supported browser will be used.

Test Steps :

1. Navigate to http://aiktchelpinghands.herokuapp.com
2. Click on the *Get Started* button

3. Click on the *Upload EEG Data* button

4. Select a file from the file dialog box

5. Click on the *Remove File* button

Expected Result : The selected file should be removed from selections.

Actual Result : We have successfully deselected the file.

Title : Upload Button

Description : Once the user has selected a valid EEG file and uploaded the same, the model should be able to make predictions on the kind of motor function being performed

Precondition : The user should upload a valid csv file containing EEG data.

Assumption : A supported browser will be used.

Test Steps :

1. Navigate to http://aiktchelpinghands.herokuapp.com

2. Click on the *Get Started* button

3. Click on the *Upload EEG Data* button

4. Select a file from the file dialog box

5. Click on the *Upload* button

Expected Result : The predictions are outputted on the web page.

Actual Result : We can see what the user is doing being displayed on the site.

Title : Output Prediction

Description : We have trained a machine learning model to predict the actions of the hand of a user given the user's EEG recordings sampled and stored in a csv file format.

Precondition : Model should already be trained and dumped in a model file.

Assumption : Relevant data is being used.

Test Steps :

1. Load the models trained for each label into separate variables.

2. Use the loaded models to predict the labels of a test dataset.

3. Calculate the Accuracy of the predictions

Expected Result : The accuracy of the predictions should be greater than 70%.

Actual Result : After testing the model(s) on our test data the accuracy for each label comes to be equal to approximately 95%, the overall accuracy of the predictions is approximately 75%.

7.2.1 Software Quality Attributes

- **Robust :** The website is stable and robust under daily use.

- **Speed of operation :** The system is able to reduce the latency (but not eliminate) with regards to the execution time.

- **Cost-effective :** Affordability is key when designing an EEG-based classification system. Our system has next to no external costs associated with it.

- **User-friendly :** The users can navigate the system with ease.

- **Effective :** The system offers an accuracy as high as 75%.

Chapter 8

Screenshots of Project

8.1 Landing Page

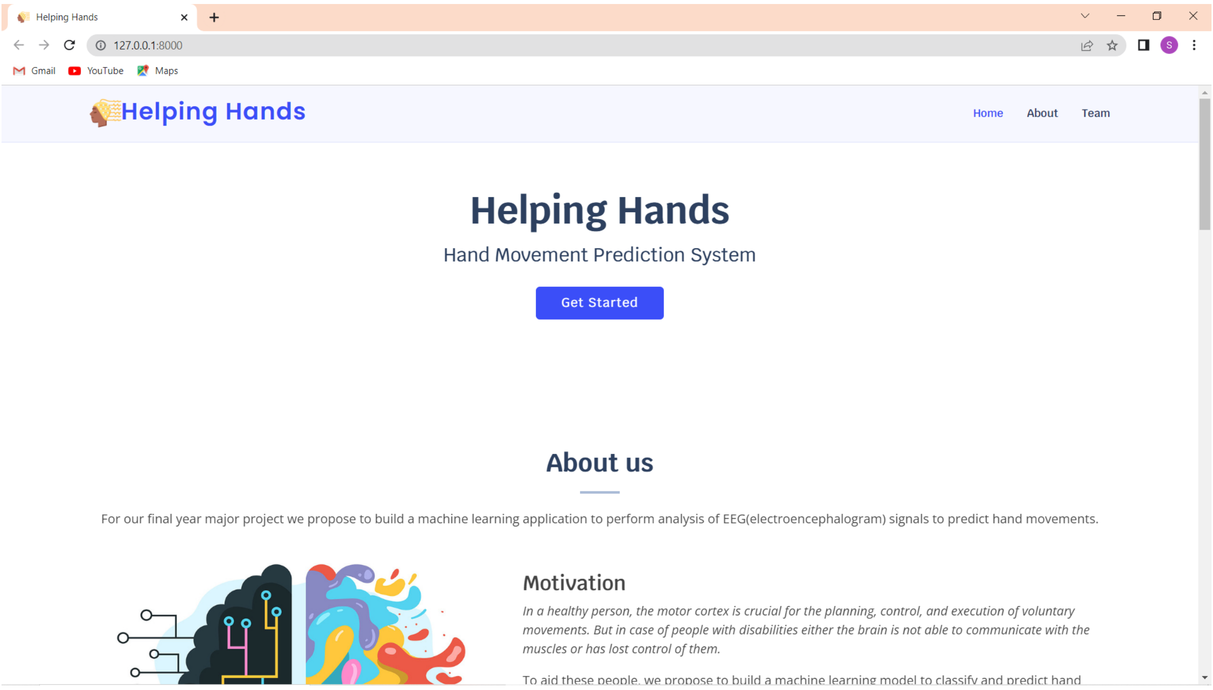


Figure 8.1: Landing Page

8.2 About Us

8.2.1 Motivation

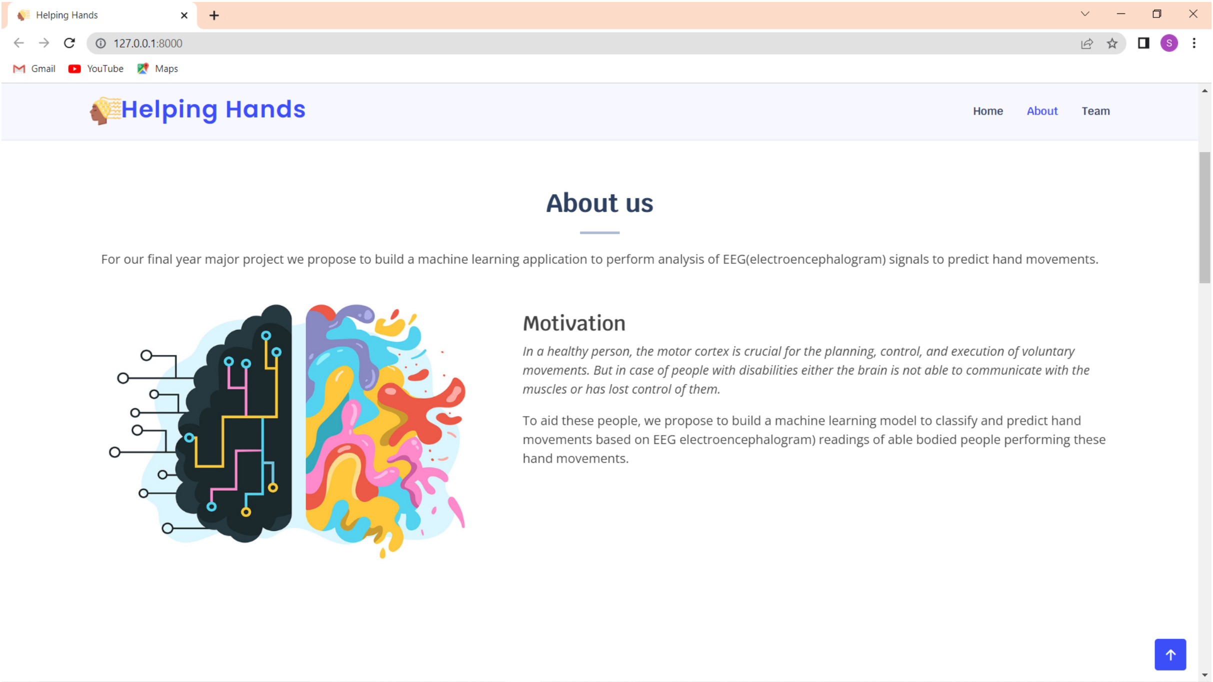


Figure 8.2: Motivation

8.2.2 Application of EEG

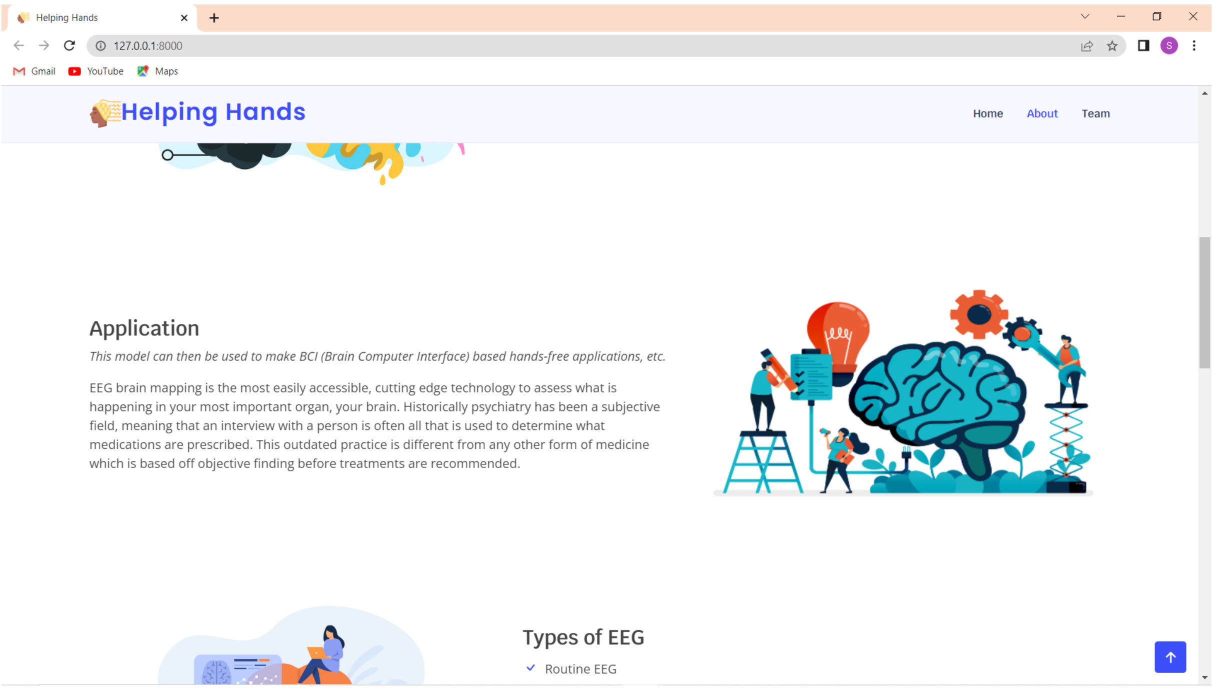


Figure 8.3: Application of EEG

8.2.3 Types of EEG

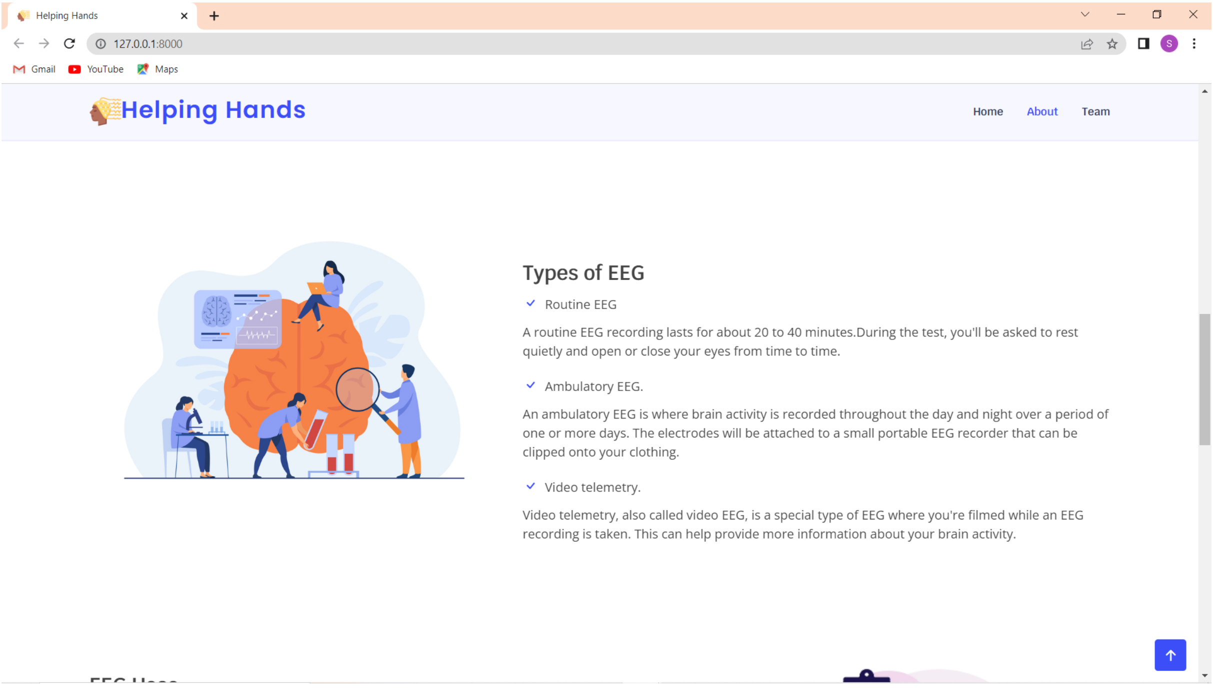


Figure 8.4: Types of EEG

8.2.4 Uses of EEG

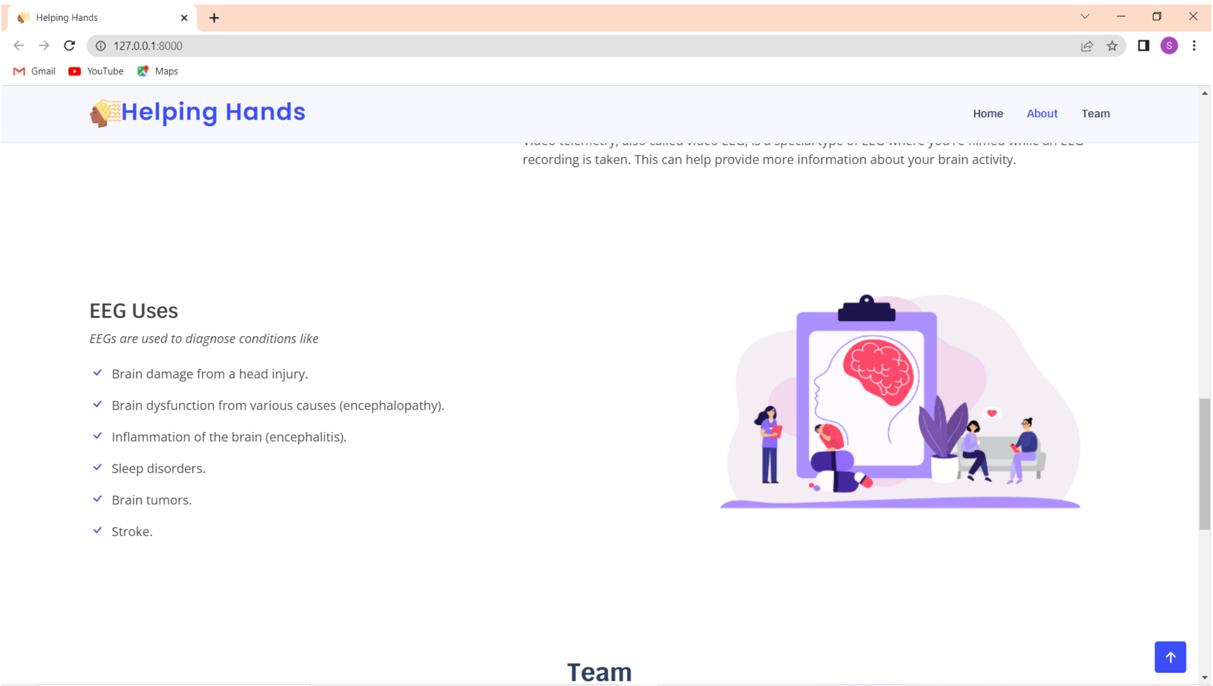


Figure 8.5: Uses of EEG

8.3 Main Section

8.3.1 Uploads

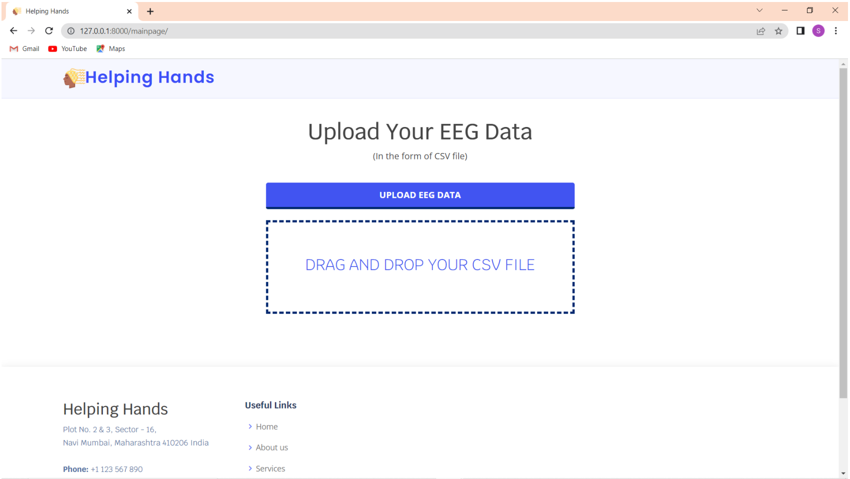


Figure 8.6: Upload Page

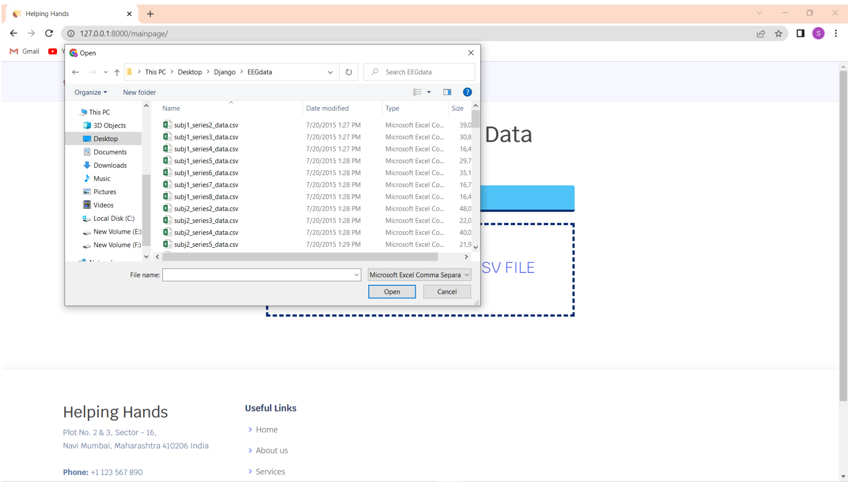


Figure 8.7: File Dialogue for accepting only .csv files

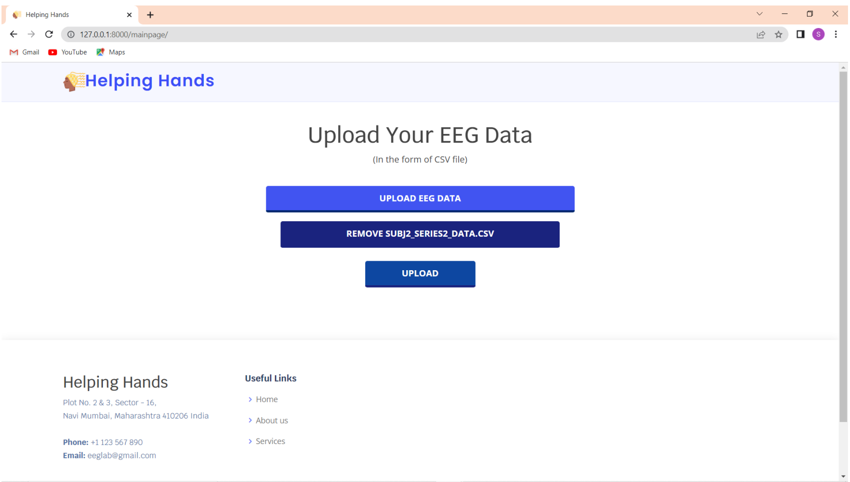


Figure 8.8: Uploaded EEG file

8.3.2 Output

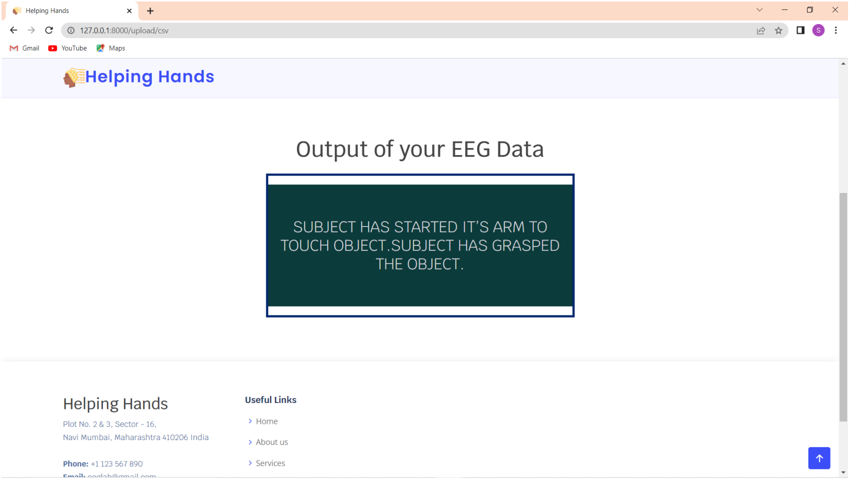


Figure 8.9: Sample Output 1

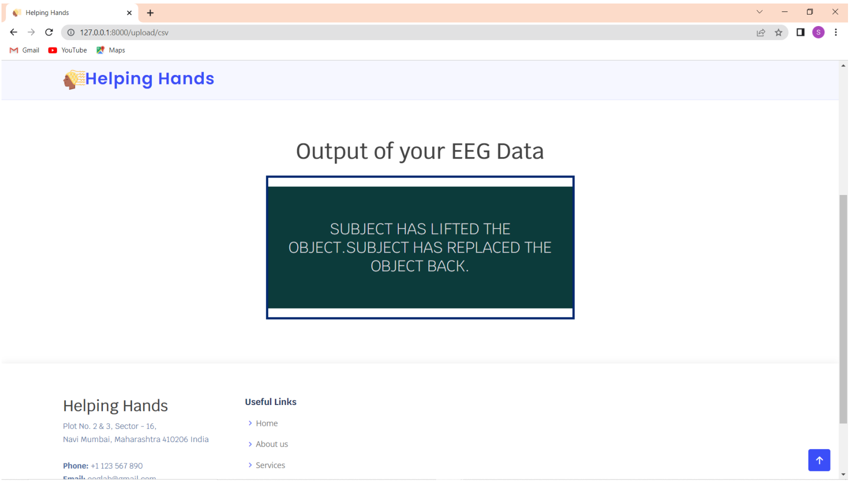


Figure 8.10: Sample Output 2

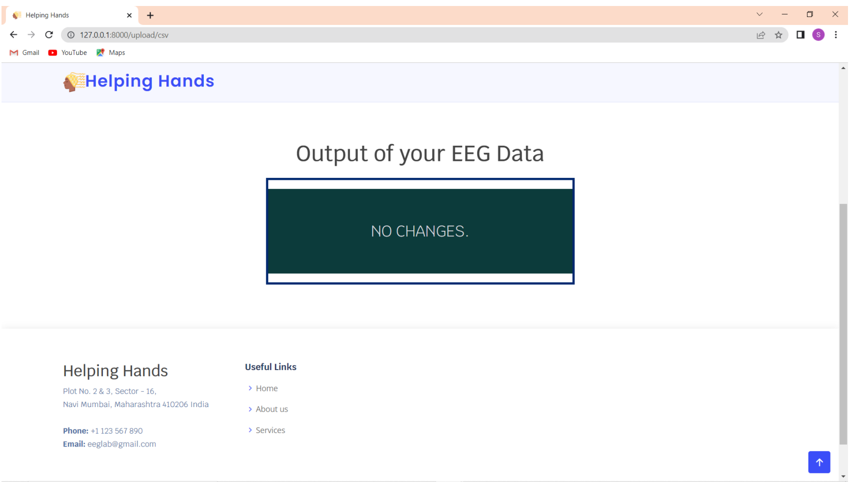


Figure 8.11: Sample Output 3

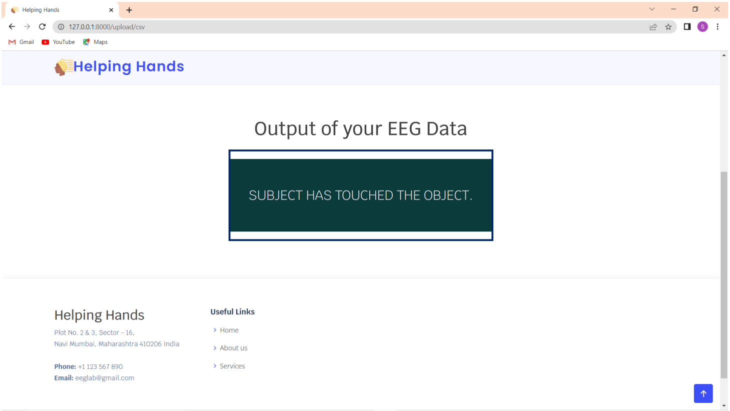


Figure 8.12: Sample Output 4

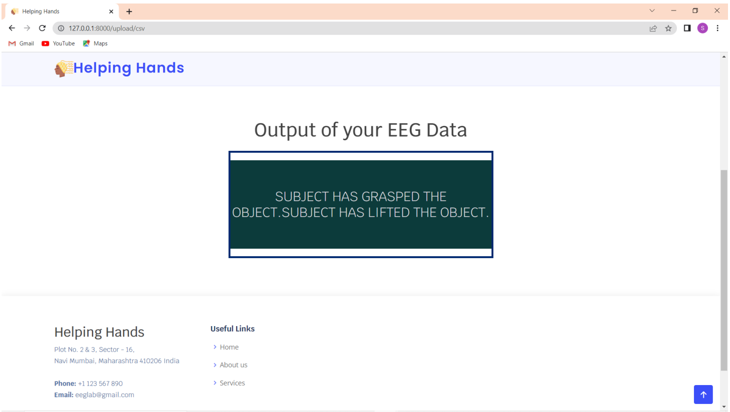


Figure 8.13: Sample Output 5

8.4 Our Team

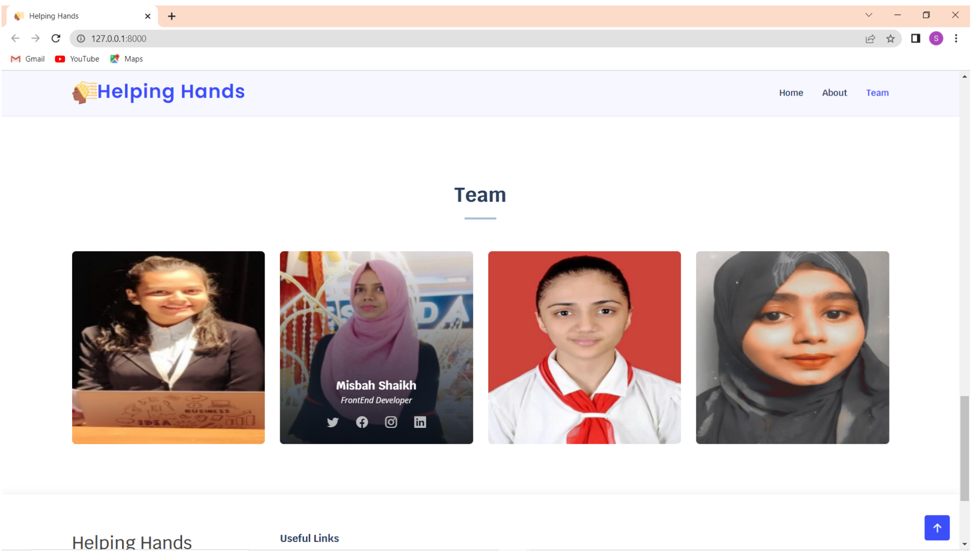


Figure 8.14: Our Team

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Using signal processing, and machine learning we can help disabled people overcome their disabilities and bring some normalcy into their day to day lives. Returning a patient's capacity to undertake everyday tasks with the help of an EEG-based BCI prosthetic device would positively contribute to their independence as well as improve their quality of life.

EEG is a non-invasive way of measuring or observing brain activity. It is also more affordable and portable as compared to methods like MRIs and fMRIs.

Analysis of EEG signals is therefore very important in developing such devices.

We have thus trained a machine learning model for this in python using various machine learning, data processing and signal processing libraries like NumPy, Pandas, Matplotlib, scikit-learn, etc. and special packages like PyWavelets and deployed the model as a web application using the Django framework.

9.2 Future Scope

- Our model can become the basis of developing various brain computer interfaces for various purposes.

- This classification system can be later used to develop better and more interactive prosthetics granting disable people more control over their arm.

- It may also be used to develop a robotic arm.

References

[1] Wikipedia contributors, “10–20 system (eeg) — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=10%E2%80%9320_system_(EEG)&oldid=1040419178, 2021. [Online; accessed 2-November-2021].

[2] Wikipedia contributors, “Electroencephalography — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=Electroencephalography&oldid=1047805792, 2021.

[3] M. Rashid, N. Sulaiman, A. P. P. Abdul Majeed, R. M. Musa, A. F. Ab. Nasir, B. S. Bari, and S. Khatun, “Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review,” *Frontiers in Neurorobotics*, vol. 14, p. 25, 2020.

[4] M. D. Luciw, E. Jarocka, and B. B. Edin, “Multi-channel eeg recordings during 3,936 grasp and lift trials with varying weight and friction,” *Scientific Data*, vol. 1, no. 1, 2014.

[5] A. W. Pise and P. P. Rege, “Comparative analysis of various filtering techniques for denoising eeg signals,” in *2021 6th International Conference for Convergence in Technology (I2CT)*, pp. 1–4, IEEE, 2021.

[6] A. Prochazka, J. Kukal, and O. Vysata, “Wavelet transform use for feature extraction and eeg signal segments classification,” in *2008 3rd International symposium on communications, control and signal processing*, pp. 719–722, IEEE, 2008.

[7] R. Chatterjee and T. Bandyopadhyay, “Eeg based motor imagery classification using svm and mlp,” in *2016 2nd international conference on Computational Intelligence and Networks (CINE)*, pp. 84–89, IEEE, 2016.

[8] B. Farnsworth, “Eeg vs. mri vs. fmri - what are the differences?.” https://imotions.com/blog/eeg-vs-mri-vs-fmri-differences/, Feb 2021.

[9] https://www.djangoproject.com/.

Appendix A

Electrode Placement System for EEG

A.1 International 10-20 System

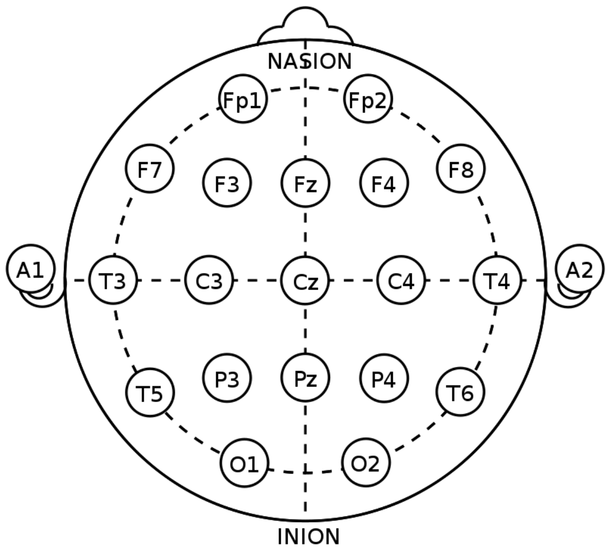


Figure A.1: International 10-20 System for Electrode Placement(1)

The International 10-20 system is a standard way to place and apply electrodes on the surface of the scalp to record EEG signals for various reasons.This method was developed to maintain standardization of testing methodologies to ensure coherent results using the scientific method.The system is based on the relationship between the location of an electrode and the underlying area of the brain, specifically the cerebral cortex.Here, the "10" and "20" refer to the fact that the actual distances between adjacent electrodes are either 10% or 20% of the total front–back or right–left distance of the skull.(1)

A.1.1 Electrode Nomenclature

Each electrode placement site has a letter to identify the lobe, or area of the brain it is reading from :

- pre-frontal (Fp)
- frontal (F)
- temporal (T)
- parietal (P)
- occipital (O)
- central (C).

There are also (Z) sites: A "Z" (zero) refers to an electrode placed on the mid-line of the skull, (FpZ, Fz, Cz, Oz) and are often utilized as 'grounds' or 'references' as they don't represent either hemisphere adequately.

Even-numbered electrodes (2,4,6,8) refer to electrode placement on the right side of the head, whereas odd numbers (1,3,5,7) refer to those on the left.This is a convention followed for the measurement of EEG, EOG, ECG as well as EMG. More commonly electrodes are referred to as either right, left, or reference or common.

The "A" (sometimes referred to as "M" for mastoid process) refers to the prominent bone process usually found just behind the outer ear (less prominent in children and some adults).(1)

A.1.2 Measurement & Positioning

Certain prominent structural features of the skull are used to locate and position the EEG electrodes. These are found and marked.

- **Nasion to Inion :** the nasion is the distinctly depressed area between the eyes, just above the bridge of the nose, and the inion, is the crest point at the back of the skull, often indicated by a bump. The Z electrodes are placed on the line joining these two at intervals of 10%, 20%, 20%, 20%, 20% and 10%.

- **Preauricular to preauricular :** The preauricular point is in front of each ear, and can be more easily located with mild palpation, and if necessary, requesting the patient to open mouth slightly. The T3, C3, Cz, C4, and T4 electrodes are placed at marks made at intervals of 10%, 20%, 20%, 20%, 20% and 10%, respectively, measured across the top of the head.

- **Skull circumference** is measured just above the ears (T3 and T4), just above the bridge of the nose (at Fpz), and just above the occipital point (at Oz). The Fp, F, T, T, and O electrodes are placed at intervals of 5%, 10%, 10%, 10%, 10%, and 5%, respectively, measured above the ear, from front (Fpz) to back (Oz) on each side.

- **Measurement methods for placement of the F and P points varry.** If measured front-to-back , they can be 25% "up" from the front and back points. If measured side-to-side, they can be 25% "up" from the side points. If measured diagonally, from Nasion to Inion through the C3 and C4 points, they will be 20% in front of and behind the C3 and C4 points. Each of these measurement methods results in different nominal electrode placements.

- When placing the A (or M) electrodes, palpation is often necessary to determine **the most pronounced point of the mastoid process** behind either ear.

Not placing electrodes at the proper positions would lead to various artifacts in the recording.(1)

A.2 10-10 System

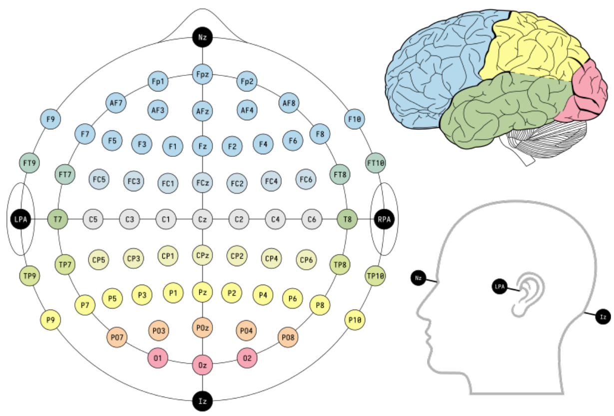


Figure A.2: Modified 10-10 System for Electrode Placement(1)

When recording a more detailed EEG with more electrodes, the new electrodes are added using the 10% division, which fills in intermediate locations halfway between those of the existing 10–20 system. This new electrode-naming-system is more complicated giving rise to the Modified Combinatorial Nomenclature (MCN).

A.2.1 Modified Combinatorial Nomenclature (MCN)

This MCN system uses 1, 3, 5, 7, 9 for the left hemisphere which represents 10%, 20%, 30%, 40%, 50% of the inion-to-nasion distance respectively. The introduction of extra letter codes allows the naming of intermediate electrodes.

The new letter codes of the MCN for intermediate electrode places are:

- AF – between Fp and F
- FC – between F and C
- FT – between F and T

- CP – between C and P
- TP – between T and P
- PO – between P and O

Also, the MCN system renames four electrodes of the 10–20 system:

- T3 is now T7
- T4 is now T8
- T5 is now P7
- T6 is now P8

An even higher-resolution nomenclature has been suggested and called the "5% system" or the "10–5 system".

Appendix B

EEG Waveforms

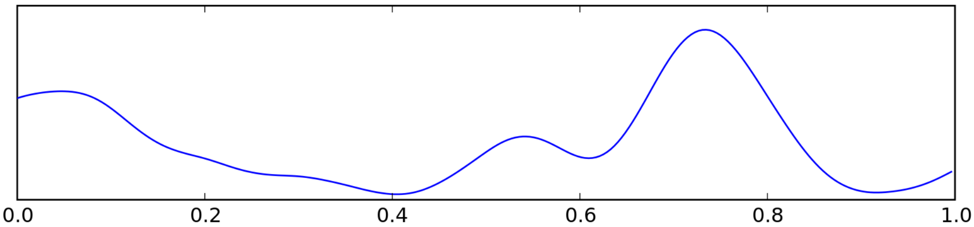


Figure B.1: EEG Waveform for Delta Waves (< 4Hz)(2)

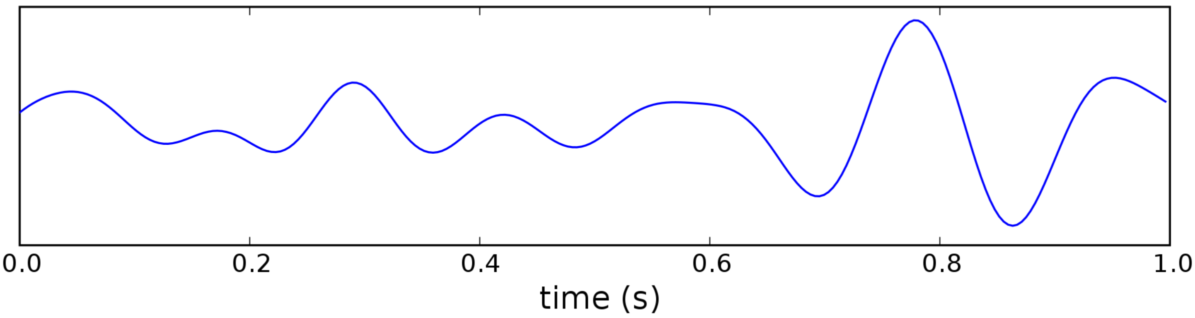


Figure B.2: EEG Waveform for Theta Waves (4 - 7Hz)(2)

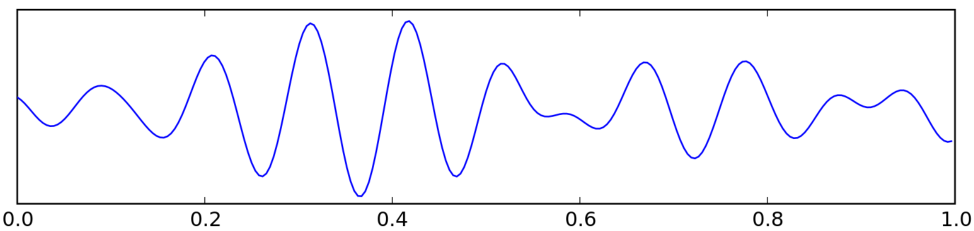


Figure B.3: EEG Waveform for Alpha Waves (8 - 15Hz)(2)

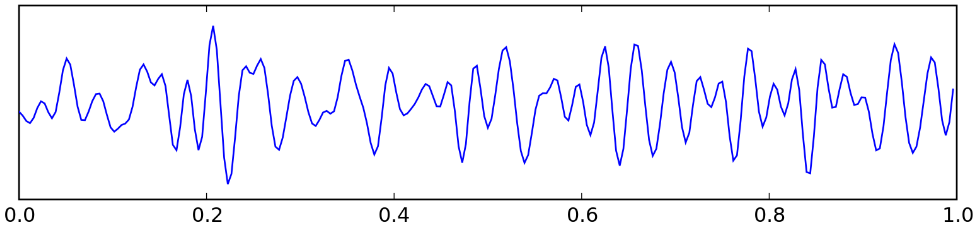


Figure B.4: EEG Waveform for Beta Waves (16 - 31Hz)(2)

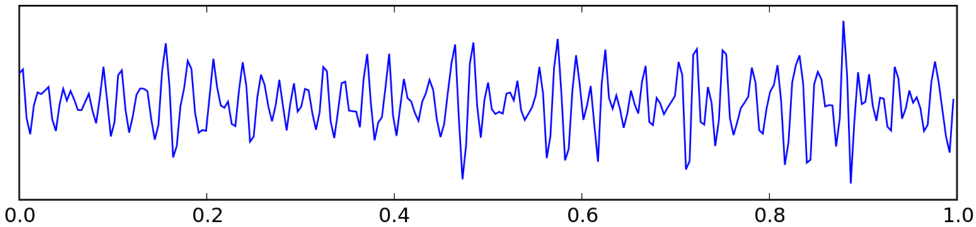


Figure B.5: EEG Waveform for Gamma Waves (> 32Hz)(2)

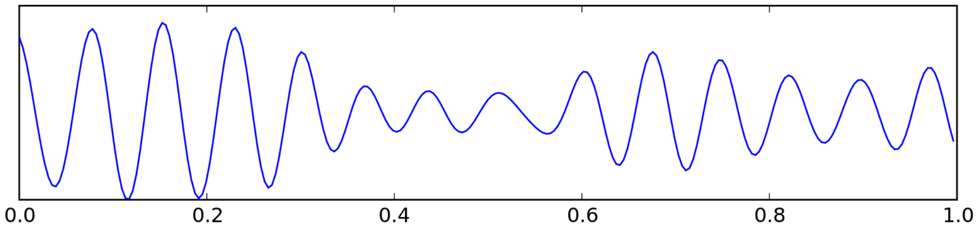


Figure B.6: EEG Waveform for Sensorimotor Rhythm or Mu Waves (8 - 12Hz)(2)