CLASSIFICATION OF EEG SIGNALS INTO DESIRED OUTPUT FOR A BIONIC ARM OF AN UPPER ARM AMPUTEE

Report submitted to the SASTRA Deemed to be University
as the requirement for the course

MEC300 MINI PROJECT WORK

Submitted by

Ravi Teja E
(Reg. No.: 123009051)
N S S Murali Prabhakar
(Reg. No.: 12009148)
Venkat Varun P
(Reg. No.: 123009170)

June 2022

SCHOOL OF MECHANICAL ENGINEERING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SCHOOL OF MECHANICAL ENGINEERING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled "**CLASSIFICATION OF EEG SIGNALS INTO DESIRED OUTPUT FOR A BIONIC ARM OF AN UPPER ARM AMPUTEE**" submitted as a requirement for the course, **MEC 300 Mini-project Work** for B.Tech. Mechanical Engineering, is a bonafide record of the work done by **Mr. Ravi Teja E (Reg. No.123009051**), **Mr. N S S Murali Prabhakar (Reg. No.123009148**), **Mr. Venkat Varun P**

(Reg. No.123009170) during the academic year 2021-22, in the School of Mechanical Engineering, under my supervision.

Signature of Project Supervisor **:**

Name with Affiliation **:** Dr. Anjan Kumar Dash, Professor, SoME

Date **:** 26.06.2022

Mini-project Viva-voce held on _________________

Examiner 1 **Examiner 2**

SCHOOL OF MECHANICAL ENGINEERING

THANJAVUR – 613 401

Declaration

We declare that the report titled "**CLASSIFICATION OF EEG SIGNALS INTO DESIRED OUTPUT FOR A BIONIC ARM OF AN UPPER ARM AMPUTEE**" submitted by us is an original work done by us under the guidance of **Dr. Anjan Kumar Dash, Professor, School of Mechanical Engineering, SASTRA Deemed to be University** during the sixth semester of the academic year 2021-22, in the **School of Mechanical Engineering**. The work is original and wherever we have used materials from other sources, we have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

Signature of the candidate(s) : E. Ravi Teja , N. Siva ,

Name of the candidate(s) : **Ravi Teja E, N S S Murali Prabhakar, Venkat Varun P**

Date : **26.06.2022**

Acknowledgment

We are very much grateful to our guide Dr. Anjan Kumar Dash, Professor, School of Mechanical Engineering, SASTRA Deemed to be University, Thanjavur for his valuable guidance and support during the entire course of our mini project.

We also feel grateful to Dr. Pugazhenthi, Dean, School of Mechanical Engineering for his cooperation during the entire project work.

Our gratefulness extends to the members of the mini project review committee and staffs of School of Mechanical Engineering for the constant support during the entire execution of our project.

Our final words of gratitude are to authors of research articles and our friends whose valuable support and encouragement helped us during our project work.

Table of Contents

List of Figures

List of Tables

Nomenclature

API **-** Application Programming Interface
BCI **-** Brain Control interface
CSV - Comma-Separated Values file
EEG – Electroencephalogram
FFT **-** Fast Fourier Transform
HTTP- Hyper Text Transform Protocol
KNN – K- Nearest Neighbours
ML **-** Machine Learning

Abstract

Ravi Teja E

Because so many people are losing their hands or legs owing to various causes such as accidents, surgeries, etc., "amputated persons" is the most widely heard word these days. We focused on a group of people who had their hands amputated and learned about their issues in everyday life, such as holding objects, making nonverbal gestures, writing, and so on. We've taken on two half-finished projects, one of which predicts our actions based on brain signal frequencies and the other of which is a bionic arm controlled by an electromagnetic band. We considered connecting the EEG data from the brain sensor to the bionic arm so that the arm could respond to the brain impulses and act accordingly.

Specific Contribution

- Developed Machine Learning Algorithm for classifying the actions based on data acquired from the Node-Red using Emotiv Insight Sensor
- Sending data from Node-Red to the microcontroller wirelessly and actuating the bionic arm
- Obtained EEG frequency values using Node-RED and sensor

Specific Learning

- Learnt the basic operation of the software Node-RED Toolbox
- Learnt about basic usage of python programming language in developing an efficient machine learning algorithm
- Programming a microcontroller to receive the data from Node-RED wirelessly

Technical Limitations & Ethical Challenges faced

- Machine learning model used to give wring predictions due to data insufficiency
- Bluetooth mode of data transfer from Node-RED to microcontroller haven’t worked due to improper functionality of Bluetooth node in Node-RED
- Emotiv Insight used to get disconnected frequently

Keywords: Bionic arm, EEG data, Electromagnetic Band, Brain Impulses

N S S Murali Prabhakar

Amputation is the frequently heard problem that is being faced by many people nowadays because of accidents, blood vessel disease, etc. Of those amputated people majority of them are facing hand amputation problems. The main aim of our project is to help amputated people do some daily work like holding and placing objects, hand gestures, etc. by actuating a bionic arm. EEG values of the brain are transferred to the computer when thinking about an action and these values are stored as a dataset. The datasets are trained and tested with the help of the K-Nearest Neighbour machine learning model and 94% efficiency is obtained. The connection between Node-RED and a microcontroller is achieved with the wireless data transfer through internet. The microcontroller is programmed and servo motors rotate in the bionic arm according to the action input received.

Specific contributions:

- Obtained frequency values for corresponding actions from the EEG sensor
- Made connections in Node-RED to obtain EEG values of actions into a dataset
- Obtained actions as output in the Node-RED with the help of ML model by configuring Python shell node

Specific Learning:

- Learned the operation, configuration, and connections of nodes in the Node-RED toolbox
- Learned some basic models of machine learning, and working of KNN algorithm

Technical Limitations & Ethical Challenges faced:

- It was difficult to maintain a good EEG quality while obtaining frequency values as quality reduces when there is a slight change in the position of head
- It was hard to put the probes of the EEG sensor at the same spot on the head every time

Venkat Varun P

Technology today is evolving in a rapid pace. It can be used to solve many problems. One main problem we have focused is on ‘AMPUTEE’. An Amputee is a person who lost part of his limb. Usually, an amputee uses Prosthesis as a replacement for missing limb for support, better mobility and to manage activities to stay independent. Our Project aims to develop a Bionic arm which can do things beyond Prosthesis can achieve. The arm is controlled by obtaining signals from Emotiv Brain sensor through interface called Node-red which helps in sending the information to the Machine Learning Algorithm. The Machine Learning Algorithm decides the action and sends information to the micro-controller which will make the bionic arm work. The accuracy of our Machine Learning Model is 94% based on the data sets given. The Node-red and micro-controller are connected via internet.

Specific Contribution

- Setting up the Emotiv Insight Sensor for controlling the Bionic arm
- Gone through literature survey on several journals and came up with some ideas and understanding

Specific Learning

- Understanding the working of EEG signals and Emotiv Insight Sensor for obtaining the data sets
- Learnt how to establish connection with Emotiv Insight Sensor and Node-red without any external subscription needed.

Technical Limitations & Ethical Challenges faced

- In spite of proper connections made, the Emotiv node in the Node-red interface becomes inactive and no datasets are stored.
- Couldn’t use the headset for more than 1 hour because of battery limitation. We had to face frequent work interruption for charging the headset

1. Introduction

The upper arm amputation is the problem which we majorly observe in the amputated people. There is no prosthetic (bionic) arm completely developed till now for those people to use and perform their daily activities like holding and releasing objects, showing some gestures, moving elbow up, down, right and even left. This is still in development stage because we need to control a prosthetic arm using our brain

One way to achieve this is recording the values of brain when thinking of some action continuously for a span of time. This can be achieved by EEG (electronic encephalogram) sensor. our brain has trillions of neurons which produce current or potential difference corresponding to the activities we perform and these are the signals which are transferred from one neuron to another. The EEG sensor measures the frequency of that current and these readings can be stored. These frequency values tend to be nearly in the same range for a particular activity we perform and this is the way we can differentiate between various activities.

1.1. Electroencephalogram (EEG)

An EEG records the electrical activity of the brain in terms of frequencies via electrodes affixed to the scalp. An EEG device's electrodes record electrical activity that is expressed at different EEG frequencies. These unprocessed EEG signals can be recognized as discrete waves with various frequencies using a technique known as the Fast Fourier Transform (FFT). One Hertz (Hz), which stands for one cycle per second, is used to measure frequency, which is the rate of electrical oscillations. According to frequency, there are four primary types of brainwaves: beta, alpha, theta, and delta.

Frequency Bands	**Frequency (Hz)**
Beta	13-30
Alpha	8-13
Theta	4-7
Delta	0.5-4
Mu	8-12
Gamma	35 and above

Table 1.1.1 EEG Frequency Range

1.2. Emotiv INSIGHT

It is a 5-channel wireless EEG system that measures brain activity in the frontal, temporal, and parieto-occipital regions. It is intended for daily usage by anyone who want to learn more about and develop their own brains. Any number of probes, from 5 to any number, may be present. The accuracy rises with the number of probes. Figure 1.2.1 shows Emotiv Insight sensor used for the project.

Figure 1.2.1 Emotiv Insight Sensor

1.3. Node-RED Toolbox

A programming tool called Node-RED is used to connect physical components and APIs in various ways within an online service. It provides an editor which can be run in the local server which makes it easier for usage. It is an open-source software. Emotiv has its own application for acquiring the data, but it's not an open-source which made us to choose Node-RED as our basic requirement. Figure 1.3.1 shows Node-RED toolbox interface.

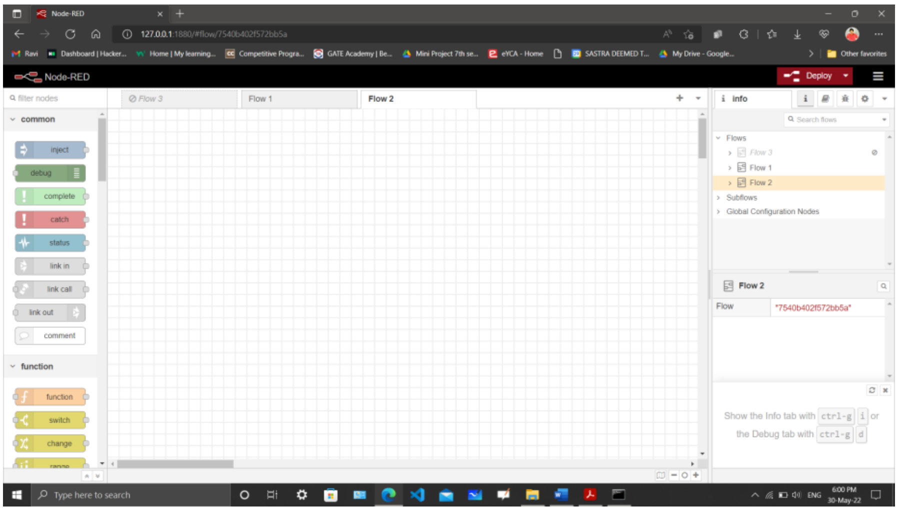


Figure 1.3.1 Node-RED Toolbox

1.4. Brain Computer Interface (BCI)

Communication has to be established between the Emotiv INSIGHT sensor and computer to transfer the data from sensor and store in the computer, so we have selected Node-RED as our BCI for visualizing the values which are obtained for specific actions.

1.5. Python & Machine Learning

Machine learning is used for classifying the actions based on the Emotiv Data, so for developing an algorithm basic knowledge on python programming language is required. Training a proper

Machine Learning algorithm can help us in achieving a particular action for its respective frequency values obtained from the Emotiv INSIGHT.

1.6. Microcontroller

Microcontroller is the heart of any electronic device. It is small and low-cost microcomputer which can perform specific tasks depending on different variables. The CPU, memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters, etc.), and other components make up a generic microcontroller. In our case we require a microcontroller which can receive data wirelessly from Node-RED and actuate the robotic arm accordingly.

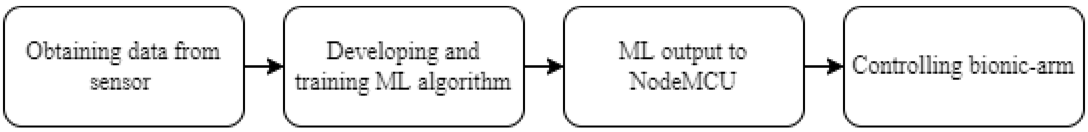
We have selected NodeMCU (ESP8266) IoT based microcontroller for free flow of data between Node-RED and Robotic arm.

1.7. Blynk IOT

Blynk is a Platform with IOS and Android apps along with cloud support to control Arduino, Raspberry Pi and the likes over the Internet. You can drag and drop widgets to build a graphical user interface for your project on a digital dashboard. In our project we have used it for establishing communication between microcontroller and Node-RED for sending data to microcontroller.

1.8. Workflow

These are the steps involved in achieving the solution for the problem statement selected.



1.9. Objective

There are many upper arm amputees in the world facing number of challenges in their day-to-day life. The objective of our project is to develop a bionic arm which can be controlled by the EEG frequency values obtained from the brain and actuate the servo motors in the arm to perform some basic actions like hold, release, acceleration, deceleration.

2. Literature Review

We learned about the many brain wave frequencies, including beta, alpha, gamma, theta, and mu, and each one's significance dependent on its frequency. For instance, the brain releases beta waves at frequencies between 13 and 60 Hertz when a person is anxious or fearful. When a person is calm both emotionally and physically, they emit alpha waves, which have frequencies between 7 and 13 Hz. Delta waves, on the other hand, are produced while a person is unconscious. In order to interpret electrical impulses, we discovered surgical implantation techniques where the arm is surgically attached and connections are also created to the nerves. These signals can later be filtered and transformed into instructions. However, this is a pricey choice, ranging from 10,000 to 120,000 USD. Although the installation is permanent, the readings are quite precise [1]. We came across projects which used 14 biopotential Emotiv probes in the sensor with gold plated connectors which offer optimal positioning for accurate spatial resolution. In addition, gyroscope generates optimal positional information. The computer system receives the brain's digital data via Bluetooth in the 2.4 GHz range. The output of EEG will be in the range of 0.2 to 45 Hz [2]. The Brain Computer Interface became known to us (BCI). A communication channel between the brain and an object that has to be controlled outside is made possible by this interface between the brain and the computer. Numerous ground-breaking uses for it exist, including cancer detection, entertainment and gaming, training and education, etc. [3]. Emotiv Insight data can be obtained using different methods and one of it is EEGLAB a MATLAB’s tool. Emotiv software is used for getting the data from sensor [4]. EEG datasets can be classified using different algorithms and for visual attention test related data Convolutional Neural Network can be used [5]. EEG signals are non-stationary, have a low signal-to-noise ratio, and are polluted with numerous disturbances and artefacts (SNR). Furthermore, high dimensional feature vector is used in the majority of EEG applications. The use of kNN and SVM in EEG classification has been demonstrated to be successful in differentiating features in EEG datasets. Nevertheless, several EEG applications produced various findings. Determining their benefits and drawbacks as well as their overall performances, this research examines the use of kNN and SVM classifier on diverse EEG applications [6]. The EEG values are recorded for actions by integrating EEG signal analyzer, EEG detector system and recognition system. The recognition of pattern is done with the help of BP neural networks (BPNN) and the multi-classification algorithms based on support vector machine respectively. The actions are transferred to a wearable intelligent mind-controlled prosthetic hand (IMCP Hand) with the help of circuit containing microprocessor [7]. The implementation of human computer interface (HCI) is based on opened and closed eye movements. According to the eye opening and closing the actions are performed. This will be used to enhance the daily life quality of disabled people. An embedded code is implemented for the transfer of data from EEG sensor to Arduino. The output is projected on LEDs or diodes [8]. To overcome the limitations of conventional wet sensors, the dry spring-loaded sensors are made of metal conductors. The main advantage of spring-loaded sensors is, it can measure EEG values without ski preparation and gel application on probes. It consists of a size adjustable soft cap which can fit for any user. This helps us by making it easy to use and adjust for obtaining EEG frequency data [9]. The electrodes are sufficiently flexible and conformable to allow for prolonged wear on the head without tiring the user. After prolonged testing, the current sensing electrodes on the market are stiff and give users headaches. When compared to commercial electrodes, the 3D printed sensing electrodes produced equivalent signal quality while offering a significantly higher level of comfort [10]. The Brain-to-Brain Interface (BTBI), which allowed for the real-time exchange of behavioral data between the brains of two rats, was another new concept to us. We have two rats in this BTBI. An "encoder" rat was given a task in which it had to decide between two visual stimuli. Cortical activity samples were taken while the encoder rat was working on the task. Later, they are sent via intracortical macrostimulation to corresponding cortical regions of a rat used as a "decoder" (ICMS). The decoder rat acquired a fair amount of accuracy in choosing behaviour that were comparable to those of the encoder rat [11]. In some instances, the electromagnetic band utilized to control the bionic arm was utilized. The arm is where the electromagnetic band is worn. Unfortunately, electromagnetic bands now exist in regular sizes and can be uncomfortable depending on the wearer. Additionally, it costs more to customize the band for better fit [12].

3. Interfacing EEG Sensor to Computer

For recording the EEG values of brain for a particular activity, we need to connect the emotiv sensor with node red. For achieving this we need to first install node red and then install some extensions of emotiv sensor in the Node-RED server. This installation of extension as adds a set of nodes called Emotiv-BCI into the Node-RED tool box. In this Emotiv-BCI, we require EMOTIV node and a Frequency Band Power node as shown in Figure 3.1.

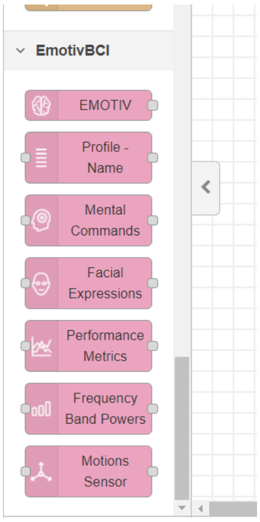


Figure 3.1 Emotiv nodes

We have 5 frequency bands namely alpha, beta-low, beta-high, gamma and theta. The emotiv insight sensor consists of 5 electrodes or probes. Hence for each frequency band we get 5 frequency readings and totally we get 25 frequency values recorded as single dataset

They are alpha1, alpha2, alpha3, alpha 4, alpha5, beta-low1, gamma1, theta1…., theta5. In the node red we can drag and drop the nodes, configure them and there by connect the nodes.

Initially "EMOTIV" node is connected with 5 "frequency band power" nodes namely alpha, beta-low, beta-high, gamma and theta. These frequency band power nodes store the frequency values of all electrodes falling in that frequency band. Each frequency band gives 5 readings in the form of a list. All the 5 lists are joined by "," using a node called "join". Then all the comma separated values are split using ",". Finally using node "join" we join a set of 25 values as a dataset and move next 25 values to the next rows respectively. Figure 3.3 depicts all frequency nodes connected to obtain the dataset. We can view the frequency set of readings by connecting all nodes and the final join node with node called "debug". For getting good frequency data we require 92-100% EEG signal quality as shown in Figure 3.2.

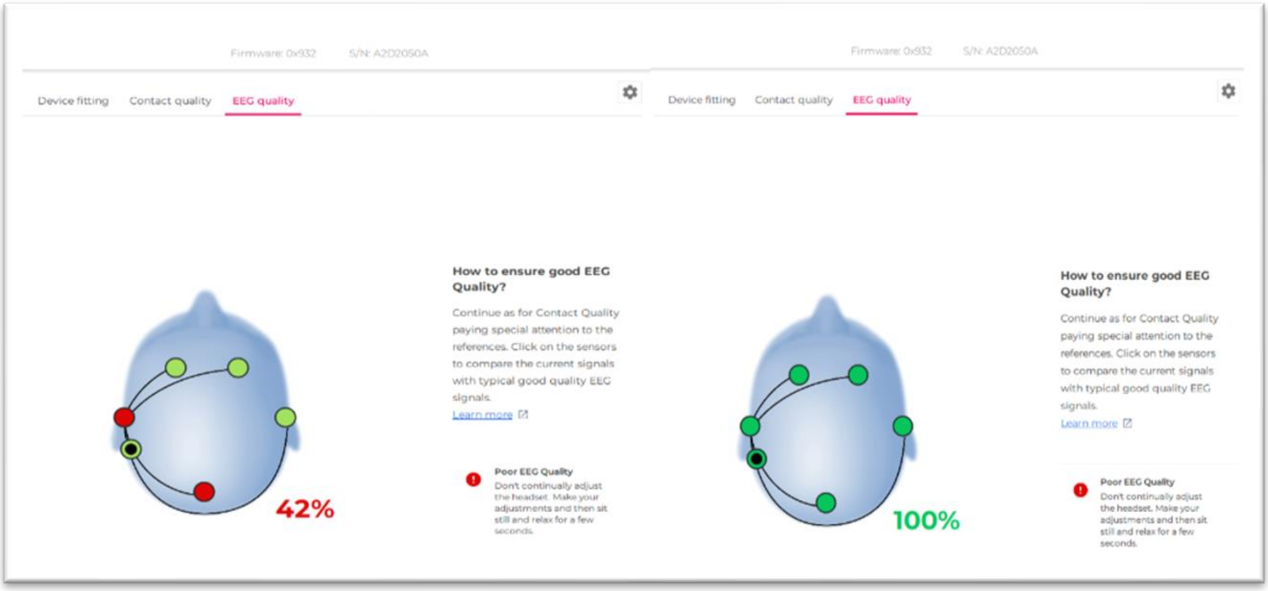


Figure 3.2 Poor and Good EEG Signal Quality

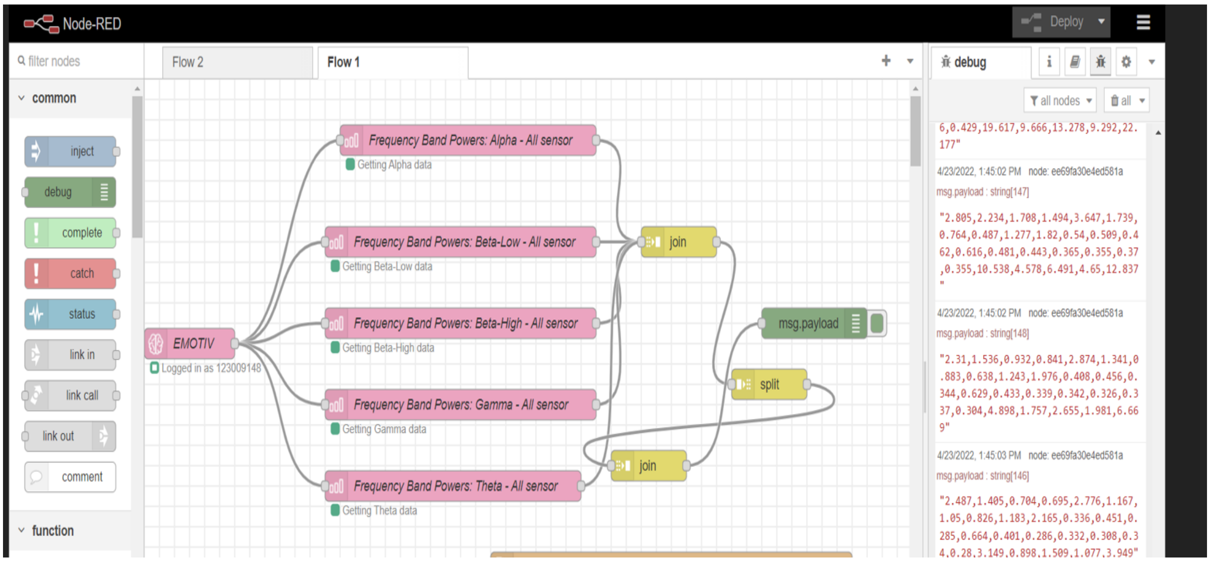


Figure 3.3 Connections in Node-RED to display frequency values

3.1. Storing EEG Frequency data

For the node red circuit designed, the EEG frequency values have to be stored which can be achieved by a node called “write file” which is meant for storage of the output data. This “write file” or “msg.payload” node is connected to the last join node in the designed circuit. The write file is configured by giving the storage location and the type of file. When all the connections between nodes are given, we can deploy the circuit which starts storing the EEG frequency data as a “csv” file into the location where we configured. As the data has been taken for a particular action performed, we add a new column in that obtained csv file called label (action) and fill the column with the name of action. Figure 3.1.1 shows all nodes used for handling the EEG frequency values.

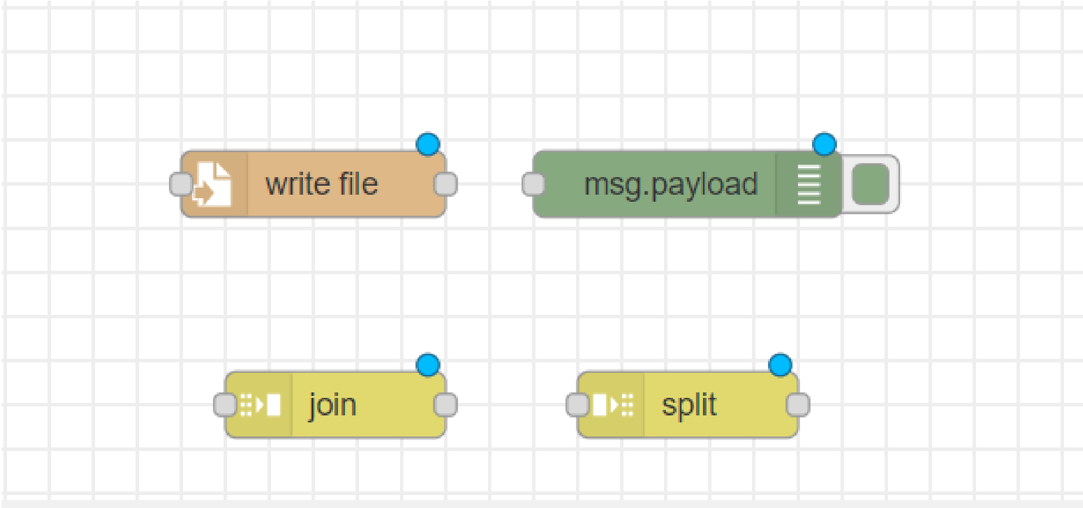
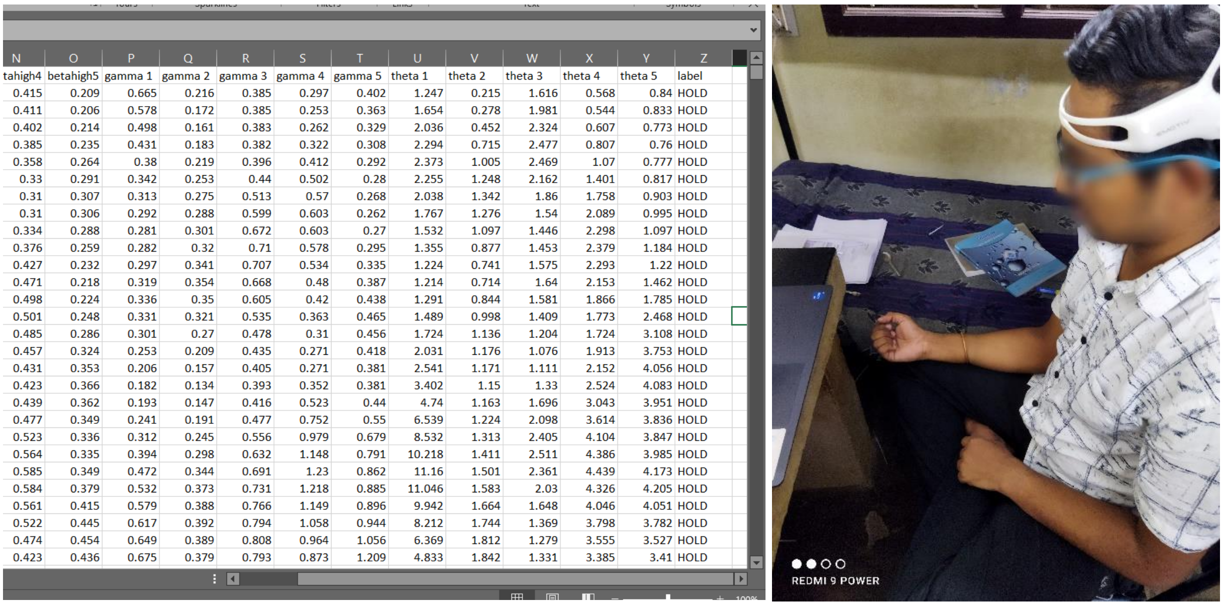


Figure 3.1.1 Different nodes used for data collection



N	O	P	Q	R	S	T	U	V	W	X	Y	Z
tahigh4	betahigh5	gamma 1	gamma 2	gamma 3	gamma 4	gamma 5	theta 1	theta 2	theta 3	theta 4	theta 5	label
0.415	0.209	0.665	0.216	0.385	0.297	0.402	1.247	0.215	1.616	0.568	0.84	HOLD
0.411	0.206	0.578	0.172	0.385	0.253	0.363	1.654	0.278	1.981	0.544	0.833	HOLD
0.402	0.214	0.498	0.161	0.383	0.262	0.329	2.036	0.452	2.324	0.607	0.773	HOLD
0.385	0.235	0.431	0.183	0.382	0.322	0.308	2.294	0.715	2.477	0.807	0.76	HOLD
0.358	0.264	0.38	0.219	0.396	0.412	0.292	2.373	1.005	2.469	1.07	0.777	HOLD
0.33	0.291	0.342	0.253	0.44	0.502	0.28	2.255	1.248	2.162	1.401	0.817	HOLD
0.31	0.307	0.313	0.275	0.513	0.57	0.268	2.038	1.342	1.86	1.758	0.903	HOLD
0.31	0.306	0.292	0.288	0.599	0.603	0.262	1.767	1.276	1.54	2.089	0.995	HOLD
0.334	0.288	0.281	0.301	0.672	0.603	0.27	1.532	1.097	1.446	2.298	1.097	HOLD
0.376	0.259	0.282	0.32	0.71	0.578	0.295	1.355	0.877	1.453	2.379	1.184	HOLD
0.427	0.232	0.297	0.341	0.707	0.534	0.335	1.224	0.741	1.575	2.293	1.22	HOLD
0.471	0.218	0.319	0.354	0.668	0.48	0.387	1.214	0.714	1.64	2.153	1.462	HOLD
0.498	0.224	0.336	0.35	0.605	0.42	0.438	1.291	0.844	1.581	1.866	1.785	HOLD
0.501	0.248	0.331	0.321	0.535	0.363	0.465	1.489	0.998	1.409	1.773	2.468	HOLD
0.485	0.286	0.301	0.27	0.478	0.31	0.456	1.724	1.136	1.204	1.724	3.108	HOLD
0.457	0.324	0.253	0.209	0.435	0.271	0.418	2.031	1.176	1.076	1.913	3.753	HOLD
0.431	0.353	0.206	0.157	0.405	0.271	0.381	2.541	1.171	1.111	2.152	4.056	HOLD
0.423	0.366	0.182	0.134	0.393	0.352	0.381	3.402	1.15	1.33	2.524	4.083	HOLD
0.439	0.362	0.193	0.147	0.416	0.523	0.44	4.74	1.163	1.696	3.043	3.951	HOLD
0.477	0.349	0.241	0.191	0.477	0.752	0.55	6.539	1.224	2.098	3.614	3.836	HOLD
0.523	0.336	0.312	0.245	0.556	0.979	0.679	8.532	1.313	2.405	4.104	3.847	HOLD
0.564	0.335	0.394	0.298	0.632	1.148	0.791	10.218	1.411	2.511	4.386	3.985	HOLD
0.585	0.349	0.472	0.344	0.691	1.23	0.862	11.16	1.501	2.361	4.439	4.173	HOLD
0.584	0.379	0.532	0.373	0.731	1.218	0.885	11.046	1.583	2.03	4.326	4.205	HOLD
0.561	0.415	0.579	0.388	0.766	1.149	0.896	9.942	1.664	1.648	4.046	4.051	HOLD
0.522	0.445	0.617	0.392	0.794	1.058	0.944	8.212	1.744	1.369	3.798	3.782	HOLD
0.474	0.454	0.649	0.389	0.808	0.964	1.056	6.369	1.812	1.279	3.555	3.527	HOLD
0.423	0.436	0.675	0.379	0.793	0.873	1.209	4.833	1.842	1.331	3.385	3.41	HOLD

Figure 3.1.2 csv file containing EEG frequency data for hold action

The Figure 3.1.2 depicts the data stored in csv file when hold action is being performed

etahigh5	gamma 1	gamma 2	gamma 3	gamma 4	gamma 5	theta 1	theta 2	theta 3	theta 4	theta 5	label
2.228	1.486	1.996	2.32	1.308	1.187	17.89	4.092	1.666	2.433	6.998	DECELRATION
2.085	1.598	2.319	2.498	1.441	1.272	21.499	3.411	1.737	3.001	11.041	DECELRATION
1.925	1.593	2.517	2.653	1.533	1.271	25.946	3.153	2.031	3.38	15.32	DECELRATION
1.833	1.472	2.518	2.71	1.551	1.168	30.053	3.249	2.527	3.473	18.736	DECELRATION
1.869	1.297	2.322	2.654	1.521	1.004	31.899	3.436	3.081	3.251	20.331	DECELRATION
2.044	1.155	2.014	2.545	1.51	0.857	30.653	3.451	3.558	2.822	19.65	DECELRATION
2.31	1.118	1.727	2.462	1.585	0.803	26.151	3.245	3.854	2.327	16.918	DECELRATION
2.573	1.193	1.563	2.44	1.752	0.869	20.201	2.846	3.955	1.89	12.894	DECELRATION
2.748	1.323	1.554	2.462	1.953	1.015	14.075	2.445	3.925	1.599	8.994	DECELRATION
2.788	1.42	1.646	2.466	2.108	1.167	9.325	2.158	3.887	1.436	5.822	DECELRATION
2.698	1.409	1.729	2.382	2.143	1.247	6.291	2.006	3.904	1.327	3.889	DECELRATION
2.533	1.262	1.702	2.155	2.021	1.212	4.795	1.933	3.941	1.241	2.994	DECELRATION
2.356	1.027	1.54	1.799	1.77	1.088	4.248	1.843	3.88	1.188	2.789	DECELRATION
2.208	0.801	1.32	1.409	1.465	0.948	4.063	1.683	3.583	1.2	2.969	DECELRATION
2.097	0.688	1.147	1.108	1.199	0.885	3.779	1.444	3.069	1.292	3.198	DECELRATION
2.009	0.729	1.093	0.974	1.023	0.944	3.367	1.19	2.46	1.415	3.365	DECELRATION
1.915	0.898	1.166	1.025	0.939	1.12	2.92	1.003	1.947	1.494	3.475	DECELRATION
1.801	1.14	1.341	1.217	0.928	1.379	2.521	0.95	1.723	1.484	3.495	DECELRATION
1.669	1.428	1.627	1.497	1.014	1.696	2.201	1.138	1.768	1.378	3.363	DECELRATION
1.537	1.77	2.041	1.821	1.231	2.053	1.951	1.632	1.949	1.203	3.057	DECELRATION
1.436	2.166	2.584	2.185	1.585	2.428	1.805	2.415	2.099	1.02	2.612	DECELRATION
1.389	2.578	3.176	2.584	2.035	2.795	1.762	3.308	2.155	0.896	2.117	DECELRATION
1.415	2.954	3.675	3.002	2.495	3.116	1.792	4.031	2.114	0.87	1.666	DECELRATION
1.52	3.226	3.919	3.369	2.846	3.336	1.864	4.383	1.998	0.924	1.34	DECELRATION
1.68	3.316	3.785	3.577	2.972	3.373	1.935	4.233	1.872	0.999	1.158	DECELRATION
1.856	3.186	3.283	3.572	2.841	3.197	1.93	3.615	1.732	1.037	1.072	DECELRATION
2.013	2.885	2.574	3.366	2.523	2.86	1.877	2.727	1.648	1.03	1.06	DECELRATION
2.139	2.504	1.875	2.989	2.143	2.458	1.766	1.81	1.519	1.013	1.065	DECELRATION

Figure 3.1.3 csv file containing EEG frequency data for deceleration action

The Figure 3.1.3 depicts the data stored in csv file when decelaration action is being performed

	alpha 5	betalow 1	betalow 2	betalow 3	betalow 4	betalow 5	betahigh1	betahigh2	betahigh3	betahigh4	betahigh5	gamma 1	gamma 2	gamma 3	gamma 4	gamma 5	theta 1	theta 2	theta 3	theta 4	theta 5	label
2	2.193	1.37	1.06	1.063	1.647	1.697	1.03	0.87	0.857	1.533	1.139	1.096	0.5	0.619	0.847	0.793	17.649	1.778	1.679	1.333	6.3	ACCELERATION
3	2.166	1.654	1.307	1.263	1.756	1.934	1.182	0.964	0.926	1.572	1.211	1.237	0.516	0.728	0.94	0.901	8.988	1.629	1.763	1.476	3.472	ACCELERATION
4	2.29	2.003	1.576	1.519	1.954	2.209	1.32	1.024	0.972	1.583	1.28	1.351	0.577	0.848	0.975	0.994	4.201	1.437	1.74	1.979	2.4	ACCELERATION
5	2.459	2.353	1.822	1.794	2.226	2.466	1.407	1.042	0.979	1.566	1.324	1.406	0.658	0.961	0.945	1.035	2.35	1.216	1.592	2.542	2.578	ACCELERATION
6	2.538	2.615	2.001	2.029	2.5	2.63	1.416	1.019	0.944	1.515	1.317	1.374	0.721	1.043	0.866	1.005	2.24	1.014	1.401	2.899	3.217	ACCELERATION
7	2.423	2.703	2.071	2.157	2.666	2.637	1.341	0.962	0.873	1.424	1.254	1.264	0.732	1.069	0.783	0.919	2.783	0.839	1.222	2.921	3.659	ACCELERATION
8	2.115	2.578	2.01	2.133	2.645	2.466	1.206	0.889	0.789	1.303	1.147	1.107	0.692	1.044	0.743	0.821	3.375	0.762	1.157	2.773	3.778	ACCELERATION
9	1.686	2.254	1.823	1.953	2.419	2.146	1.054	0.826	0.721	1.184	1.028	0.977	0.639	0.992	0.772	0.78	3.713	0.728	1.146	2.658	3.562	ACCELERATION
10	1.243	1.804	1.542	1.657	2.039	1.743	0.927	0.795	0.695	1.105	0.931	0.969	0.619	0.931	0.856	0.867	3.784	0.767	1.318	2.686	3.012	ACCELERATION
11	0.903	1.338	1.228	1.317	1.61	1.348	0.851	0.806	0.718	1.098	0.881	1.161	0.674	0.883	0.974	1.124	3.449	0.802	1.362	3.034	2.349	ACCELERATION
12	0.757	0.958	0.947	1.009	1.238	1.041	0.827	0.855	0.783	1.174	0.884	1.568	0.818	0.864	1.126	1.542	2.794	0.819	1.493	3.305	1.808	ACCELERATION
13	0.844	0.731	0.748	0.788	0.988	0.868	0.844	0.926	0.873	1.312	0.933	2.113	1.015	0.87	1.316	2.043	2.121	0.814	1.431	3.574	1.538	ACCELERATION
14	1.123	0.658	0.648	0.674	0.871	0.828	0.884	0.998	0.966	1.469	1.005	2.662	1.203	0.892	1.545	2.518	1.399	0.8	1.384	3.394	1.612	ACCELERATION
15	1.485	0.694	0.624	0.653	0.85	0.878	0.929	1.05	1.045	1.6	1.069	3.05	1.313	0.922	1.793	2.837	1.084	0.879	1.361	3.066	1.881	ACCELERATION
16	1.788	0.777	0.635	0.688	0.871	0.961	0.969	1.071	1.101	1.668	1.101	3.163	1.314	0.981	2.034	2.927	0.888	1.033	1.412	2.538	2.164	ACCELERATION
17	1.909	0.859	0.646	0.746	0.9	1.044	0.997	1.061	1.132	1.656	1.092	2.989	1.223	1.1	2.228	2.782	0.973	1.329	1.618	2.242	2.546	ACCELERATION
18	1.807	0.934	0.653	0.817	0.935	1.125	1.005	1.03	1.138	1.566	1.049	2.601	1.081	1.271	2.318	2.46	1.143	1.618	1.869	2.255	2.723	ACCELERATION
19	1.543	1.015	0.677	0.908	0.979	1.223	0.988	0.987	1.121	1.42	0.985	2.125	0.934	1.446	2.258	2.055	1.389	1.946	2.156	2.694	3.1	ACCELERATION
20	1.234	1.116	0.742	1.024	1.015	1.346	0.941	0.937	1.076	1.253	0.914	1.681	0.814	1.55	2.035	1.664	1.772	2.102	2.392	3.724	3.324	ACCELERATION
21	0.985	1.237	0.851	1.154	1.015	1.478	0.867	0.883	1.002	1.1	0.845	1.332	0.73	1.541	1.699	1.341	2.252	2.143	2.434	5.032	3.72	ACCELERATION
22	0.841	1.36	0.983	1.266	0.953	1.582	0.78	0.827	0.907	0.987	0.781	1.076	0.67	1.418	1.333	1.103	2.776	2.005	2.355	6.423	3.835	ACCELERATION
23	0.777	1.455	1.111	1.319	0.827	1.619	0.698	0.783	0.805	0.931	0.73	0.881	0.622	1.21	1.014	0.938	3.34	1.809	2.062	7.234	3.867	ACCELERATION
24	0.737	1.509	1.218	1.297	0.673	1.575	0.645	0.766	0.722	0.928	0.702	0.726	0.588	0.982	0.8	0.832	3.713	1.629	1.763	7.298	3.547	ACCELERATION
25	0.679	1.516	1.291	1.21	0.557	1.47	0.643	0.795	0.686	0.968	0.716	0.595	0.574	0.808	0.715	0.777	4.076	1.496	1.453	6.797	3.082	ACCELERATION
26	0.616	1.478	1.319	1.09	0.542	1.339	0.704	0.878	0.716	1.035	0.788	0.487	0.58	0.716	0.748	0.758	3.977	1.432	1.225	5.863	2.67	ACCELERATION
27	0.567	1.406	1.291	0.968	0.648	1.221	0.824	1.001	0.814	1.112	0.913	0.411	0.599	0.694	0.87	0.765	3.876	1.401	1.155	5.067	2.422	ACCELERATION
28	0.568	1.329	1.217	0.869	0.843	1.146	0.986	1.138	0.963	1.193	1.072	0.378	0.618	0.697	1.043	0.786	3.32	1.335	1.026	4.363	2.503	ACCELERATION
29	0.643	1.286	1.131	0.822	1.063	1.131	1.175	1.26	1.131	1.288	1.241	0.387	0.633	0.689	1.233	0.808	3.014	1.321	1.03	3.895	2.986	ACCELERATION

Figure 3.1.4 csv file containing EEG frequency data for acceleration action

The Figure 3.1.4 depicts the data stored in csv file when acceleration action is being performed

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	alpha 4	alpha 5	betalow 1	betalow 2	betalow 3	betalow 4	betalow 5	betahigh1	betahigh2	betahigh3	betahigh4	betahigh5	gamma 1	gamma 2	gamma 3	gamma 4	gamma 5	theta 1	theta 2	theta 3	theta 4	theta 5	label
2	1.233	1.826	1.21	0.857	0.746	1.135	1.131	1.095	1.078	0.727	0.981	1.101	1.944	0.922	1.006	0.642	1.027	13.186	1.536	27.47	4.093	3.105	RELEASE
3	1.07	1.665	1.312	1.018	0.749	1.057	1.074	1.086	1.127	0.76	0.98	1.081	1.721	0.794	0.959	0.618	0.851	12.994	1.903	28.103	4.957	3.878	RELEASE
4	1.082	1.614	1.459	1.201	0.801	1.038	1.069	1.04	1.122	0.772	0.94	1.035	1.522	0.762	0.989	0.693	0.802	15.449	2.557	26.717	6.09	5.156	RELEASE
5	1.244	1.618	1.629	1.373	0.884	1.099	1.111	0.972	1.081	0.77	0.892	0.974	1.401	0.833	1.088	0.864	0.885	19.915	3.25	22.585	7.389	6.595	RELEASE
6	1.492	1.642	1.769	1.484	0.955	1.213	1.161	0.898	1.028	0.761	0.866	0.905	1.348	0.963	1.208	1.079	1.034	25.937	3.751	18.29	8.369	7.752	RELEASE
7	1.769	1.731	1.813	1.494	0.963	1.322	1.174	0.831	0.977	0.749	0.877	0.834	1.317	1.091	1.277	1.268	1.162	29.047	3.831	14.177	8.86	8.107	RELEASE
8	2.028	1.949	1.722	1.393	0.887	1.373	1.123	0.779	0.929	0.734	0.92	0.768	1.265	1.171	1.244	1.38	1.21	29.403	3.457	11.283	8.666	7.491	RELEASE
9	2.23	2.323	1.504	1.209	0.741	1.352	1.014	0.749	0.882	0.717	0.972	0.714	1.164	1.184	1.096	1.383	1.157	25.929	2.782	9.143	7.787	6.393	RELEASE
10	2.329	2.802	1.214	0.99	0.57	1.277	0.883	0.736	0.831	0.697	1.007	0.681	1.019	1.132	0.865	1.288	1.032	20.507	2.094	7.507	6.757	5.18	RELEASE
11	2.297	3.258	0.926	0.788	0.422	1.185	0.764	0.731	0.775	0.67	1.008	0.668	0.858	1.046	0.618	1.147	0.907	14.056	1.564	5.8	5.42	4.102	RELEASE
12	2.134	3.515	0.703	0.637	0.329	1.11	0.68	0.722	0.717	0.635	0.972	0.666	0.72	0.958	0.416	1.021	0.837	8.465	1.337	4.141	4.415	3.567	RELEASE
13	1.887	3.444	0.58	0.547	0.3	1.073	0.634	0.698	0.66	0.588	0.906	0.664	0.63	0.884	0.284	0.96	0.85	5.26	1.359	2.923	3.418	3.044	RELEASE
14	1.612	3.028	0.553	0.508	0.327	1.085	0.62	0.664	0.608	0.537	0.827	0.655	0.605	0.842	0.237	0.977	0.91	3.762	1.473	1.912	2.833	2.975	RELEASE
15	1.367	2.361	0.601	0.508	0.4	1.147	0.632	0.629	0.563	0.495	0.753	0.638	0.652	0.831	0.272	1.064	0.996	3.379	1.633	1.389	2.411	2.486	RELEASE
16	1.185	1.628	0.695	0.547	0.511	1.252	0.671	0.61	0.535	0.477	0.698	0.622	0.752	0.841	0.413	1.179	1.074	3.256	1.611	1.395	2.082	2.087	RELEASE
17	1.069	1.03	0.802	0.622	0.648	1.375	0.73	0.615	0.531	0.49	0.671	0.612	0.878	0.848	0.633	1.278	1.135	2.829	1.638	3.148	1.834	1.642	RELEASE
18	1.002	0.687	0.898	0.723	0.792	1.476	0.795	0.646	0.55	0.529	0.67	0.616	1.002	0.85	0.902	1.329	1.179	2.92	1.497	9.216	1.673	1.457	RELEASE
19	0.945	0.59	0.961	0.824	0.916	1.515	0.847	0.697	0.592	0.582	0.696	0.639	1.098	0.838	1.142	1.299	1.217	2.86	1.454	20.415	1.712	1.602	RELEASE
20	0.869	0.651	0.978	0.892	0.991	1.464	0.864	0.769	0.651	0.645	0.764	0.693	1.139	0.809	1.287	1.178	1.244	3.114	1.303	34.316	1.91	1.978	RELEASE
21	0.778	0.739	0.952	0.897	1.002	1.323	0.838	0.859	0.72	0.716	0.879	0.781	1.13	0.758	1.298	0.992	1.246	3.21	1.212	49.903	2.228	2.474	RELEASE
22	0.695	0.785	0.897	0.843	0.954	1.13	0.783	0.957	0.781	0.793	1.03	0.894	1.064	0.687	1.194	0.774	1.198	3.413	1.082	58.437	2.558	2.895	RELEASE
23	0.643	0.775	0.831	0.755	0.867	0.945	0.732	1.047	0.821	0.869	1.187	1.006	0.96	0.608	1.023	0.57	1.099	3.457	0.954	61.397	2.782	3.171	RELEASE
24	0.623	0.722	0.765	0.673	0.765	0.814	0.705	1.111	0.831	0.934	1.313	1.087	0.843	0.549	0.845	0.42	0.985	3.599	0.868	53.741	2.843	3.206	RELEASE
25	0.63	0.649	0.706	0.628	0.674	0.756	0.71	1.129	0.813	0.976	1.374	1.11	0.763	0.539	0.711	0.344	0.915	3.55	0.718	41.306	2.738	3.057	RELEASE
26	0.68	0.592	0.656	0.63	0.61	0.753	0.736	1.093	0.773	0.987	1.352	1.067	0.754	0.59	0.644	0.339	0.939	3.428	0.639	26.355	2.57	2.713	RELEASE
27	0.813	0.58	0.62	0.668	0.576	0.772	0.759	1.009	0.726	0.962	1.255	0.971	0.827	0.68	0.635	0.382	1.054	3.069	0.502	13.884	2.423	2.305	RELEASE
28	1.058	0.63	0.604	0.713	0.568	0.784	0.763	0.895	0.684	0.91	1.113	0.854	0.96	0.769	0.654	0.438	1.213	2.51	0.444	6.999	2.403	1.892	RELEASE
29	1.403	0.728	0.606	0.738	0.575	0.774	0.739	0.775	0.658	0.84	0.973	0.75	1.094	0.816	0.665	0.473	1.334	1.934	0.404	3.92	2.418	1.537	RELEASE

Figure 3.1.5 csv file containing EEG frequency data for release action

The Figure 3.1.5 depicts the data stored in csv file when release action is being performed

4. Development of ML Algorithm

4.1. Creating & Classifying Dataset

After successfully acquiring individual actions related datasets, a single .csv file has been created which consists of all action datasets. This .csv file is given as input for machine learning algorithm to get it trained itself for real-time data when the sensor is placed on our head.

4.2. Machine Learning Model

Machine learning is used for predicting the results based on previous experiences. Out of different ML models we have selected KNN (K-Nearest Neighbor) algorithm which is best suited for numerical datasets. K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

KNN Step by Step:

I. Opening the data and split it into Train and test data sets
II. Check the similarity (Calculating the distance between the data instances)
III. Once after calculating the data finding the K value
IV. Generate a set of response
V. Accuracy function for predicting accuracy
VI. Combining in main function

We used Google collab for training and testing the model. We uploaded the csv file containing dataset. The dataset is loaded and it is split into train (80%) and test (20%) sets.

Loading

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot  as plt
     dataset = pd.read_csv('/content/ravi_data.csv')
     X = dataset.iloc[:, :-1].values
     y = dataset.iloc[:, -1].values
     print(X)
     print(y)
```

Figure 4.2.1 Loading Dataset into Collab

In the Figure 4.2.1 we have imported the data from local machine to google collab for training and testing the Machine learning model. We used inbuilt python functions for loading the dataset.

Splitting

```
[ ]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
     print(X_train)
     print(y_train)
```

Figure 4.2.2 Splitting data into train and test

We used train_test_split inbuilt functions for splitting the given dataset into training set and testing set with a separation ratio of 80% to training set and 20% to testing set as shown in Figure 4.2.2.

KNN

```
[ ]  from sklearn.neighbors import KNeighborsClassifier
     neighbours = np.arange(1,26)
     print(neighbours)
     train_accuracy =np.empty(len(neighbours))
     test_accuracy = np.empty(len(neighbours))

     for i,k in enumerate(neighbours):
         #Setup a knn classifier with k neighbors
         knn=KNeighborsClassifier(n_neighbors=k,algorithm="kd_tree",n_jobs=-1)

         #Fit the model
         knn.fit(X_train,y_train.ravel())

         #Compute accuracy on the training set
         train_accuracy[i] = knn.score(X_train, y_train.ravel())

         #Compute accuracy on the test set
         test_accuracy[i] = knn.score(X_test, y_test.ravel())
```

Figure 4.2.3 KNN Algorithm

The Figure 4.2.3 depicts the actual KNN algorithm implementation. We imported KNN predefined libraries from sklearn and used them for training and testing the model. We selected kd_tree algorithm for our project purpose using which the model will calculate the train and test accuracy for all rows in dataset.

Nearest Neighbours

```
[ ]  idx = np.where(test_accuracy == test_accuracy[5])
     x = neighbours[idx]
     knn=KNeighborsClassifier(n_neighbors=x[0],algorithm="kd_tree",n_jobs=-1)
     knn.fit(X_train,y_train.ravel())
     print(le.inverse_transform(y_train.ravel()))
     import pickle
     pickle.dump(knn, open("model.pkl","wb"))
```

Figure 4.2.4 Writing Model file

This part (Figure 4.2.4) of snippet is used for saving the model into a pickle file using which we can test the real time predictions.

Confusion Matrix

```
[ ]  cm = confusion_matrix(y_test, knn_predicted_test_labels)
     print(cm)

     [[364  11   0   2]
      [ 25 349   1  14]
      [  1   5 369   2]
      [  6  14   3 291]]
```

Figure 4.2.5 Confusion Matrix

This confusion matrix shown in the Figure 4.2.5 depicts how many predictions went correct for a given label of dataset by which we can come to a conclusion on whether to improve the model or proceed with it.

4.3. Deploying ML Model into Node-RED

The model.pkl file is obtained in which the Machine Learning model is loaded. This file is used for real-time predictions of actions. To access this file a test python file is created and the real-time data is fed into the python file from Node-red into python shell.

4.3.1. Python Shell in Node-RED

In Node-RED an inbuilt node named as python shell as shown in the Figure 4.3.1.1 is available using which the test python file is executed. The path of the test python file from local machine is copied and pasted in the python shell node description. This helps the Node-RED to access the python file and there by the model.pkl file for real-time predictions.

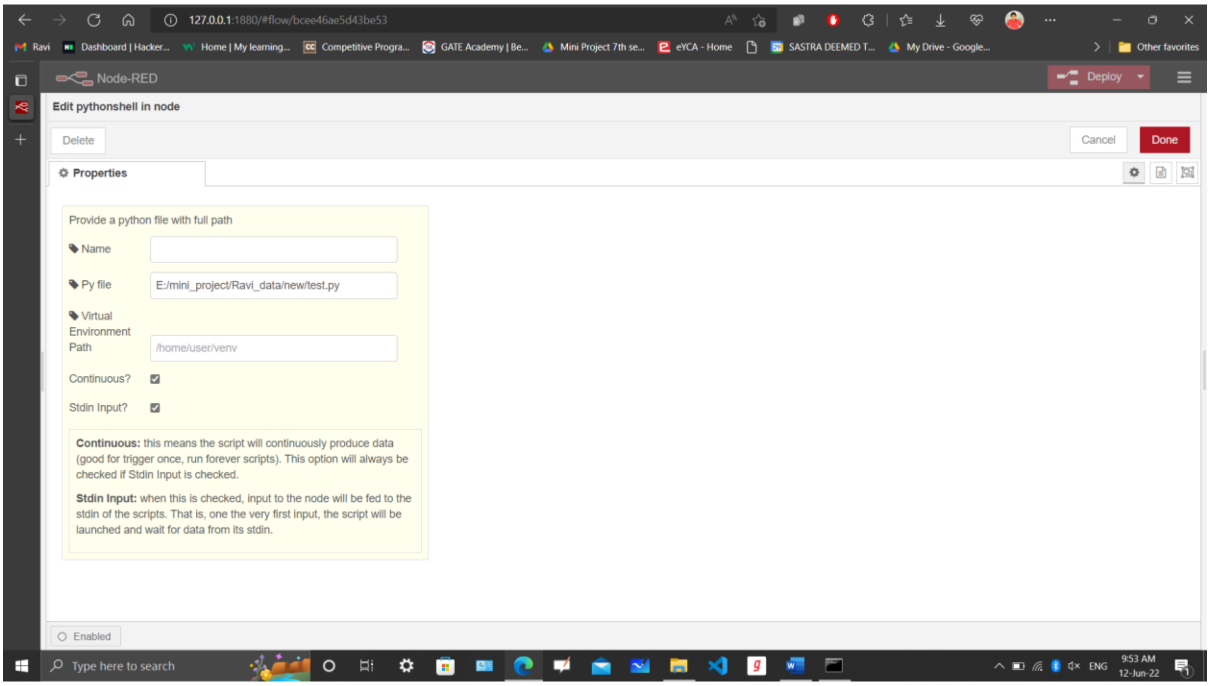


Figure 4.3.1.1 Configuration of pythonshell node in Node-RED

Nodes such as “EMOTIV”, “FREQUENCY BANDS”, “JOIN”, “SPLIT”, “PYTHONSHELL”, “SPLIT”, “SERIAL OUT” are added to the workspace of Node RED. The purpose of “EMOTIV”, “FREQUENCY BANDS” and “JOIN” nodes were discussed in the earlier section. The functions of the other nodes are as follows:

4.3.2. SPLIT Node 1

The output from the join node is a string of 25 elements each separated by comma. The input to the python file for prediction, is 25 values separated by a newline character. This is achieved by the SPLIT node. Based on the commas, this node splits those elements into 25 individual elements which later is sent to the PYTHONSHELL node.

4.3.3. PYTHONSHELL Node

The model was saved to the local disk from Google Collab. To predict the EEG signals Node RED must be given access to the saved model. A python script was written to use the classification for prediction. PYTHONSHELL Node makes this task possible. By specifying the python file’s location Node RED can access the python script thereby the incoming EEG values are predicted.

4.3.4. SPLIT Node 2

The output from the PYTHONSHELL Node is a number that corresponds to the mental state, followed by a newline character. This value is to be sent to the Microcontroller via Serial Communicator. The microcontroller actuates the servos based on the number. So, the number and the newline character should be split so that only the number is fed into the microcontroller.

4.4. Displaying output from ML Model

The sensor is placed on the head and it is connected to the Node-RED using dongle. The real-time EEG values are transferred to Node-RED and from which it is given as input to the test python file through pythonshell. The model in test python file is accessed and the EEG values are given as input to the model and the predictions are done. These predictions are displayed in the Node-RED debug window using debug node.

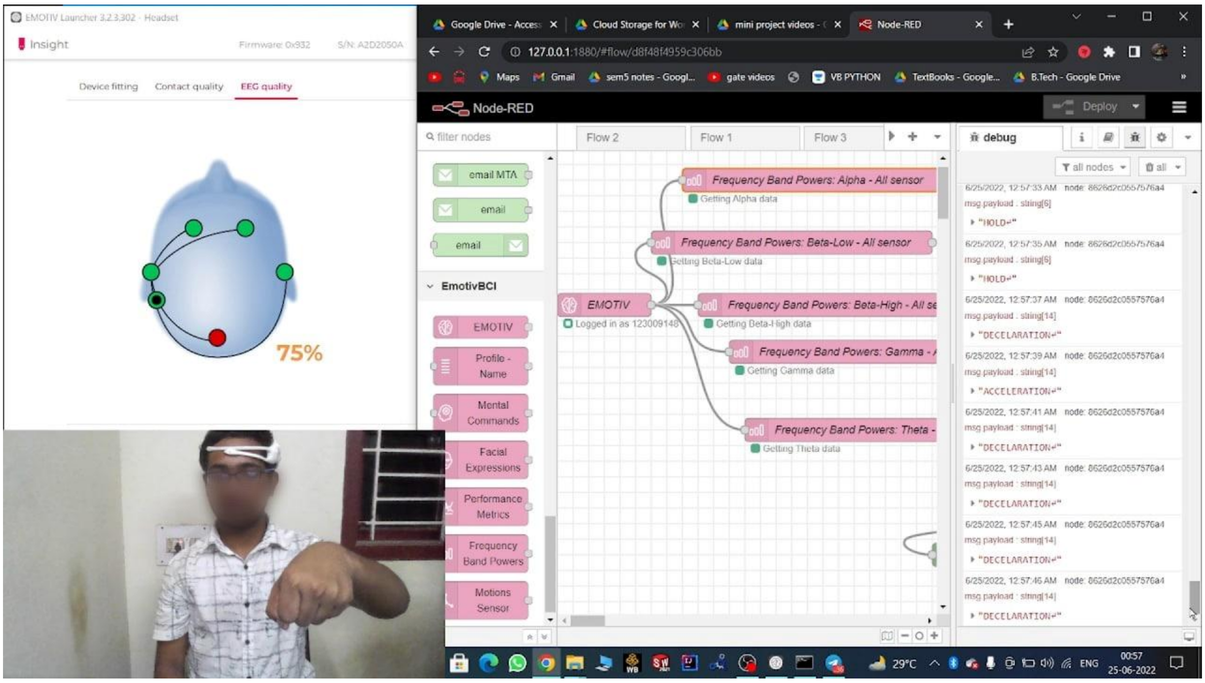


Figure 4.4.1 Displaying DECELARATION action in Node-RED debug window

In the the Figure 4.4.1 when the person is thinking about deceleration the action is displayed in Node-RED debug window.

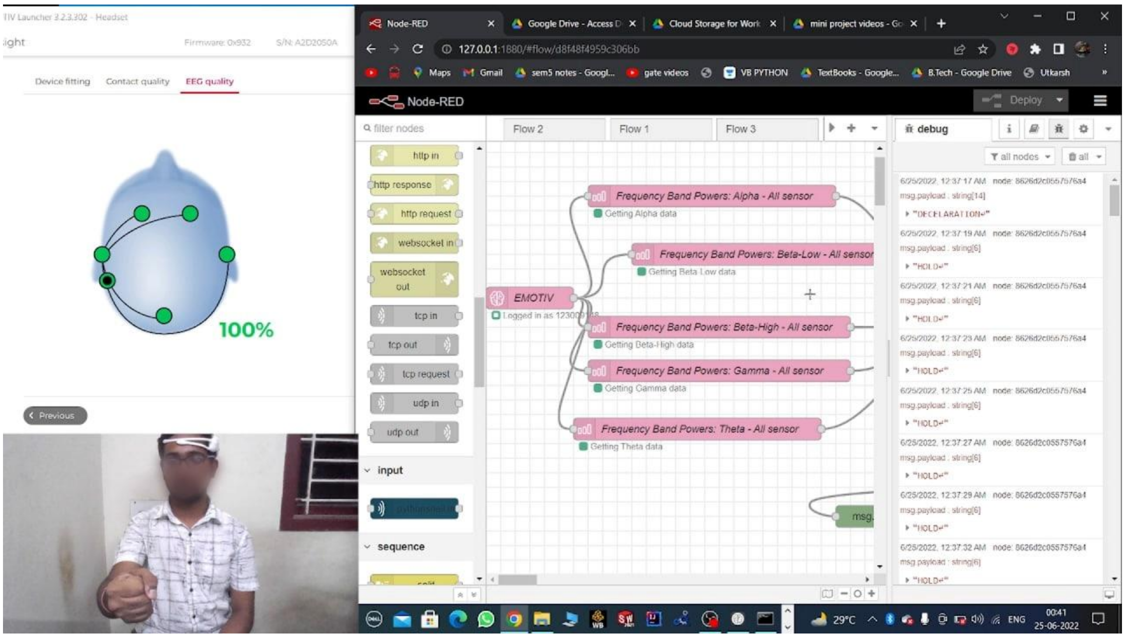


Figure 4.4.2 Displaying HOLD action in Node-RED debug window

In the Figure 4.4.2 when the person is thinking about hold the action is displayed in Node-RED debug window.

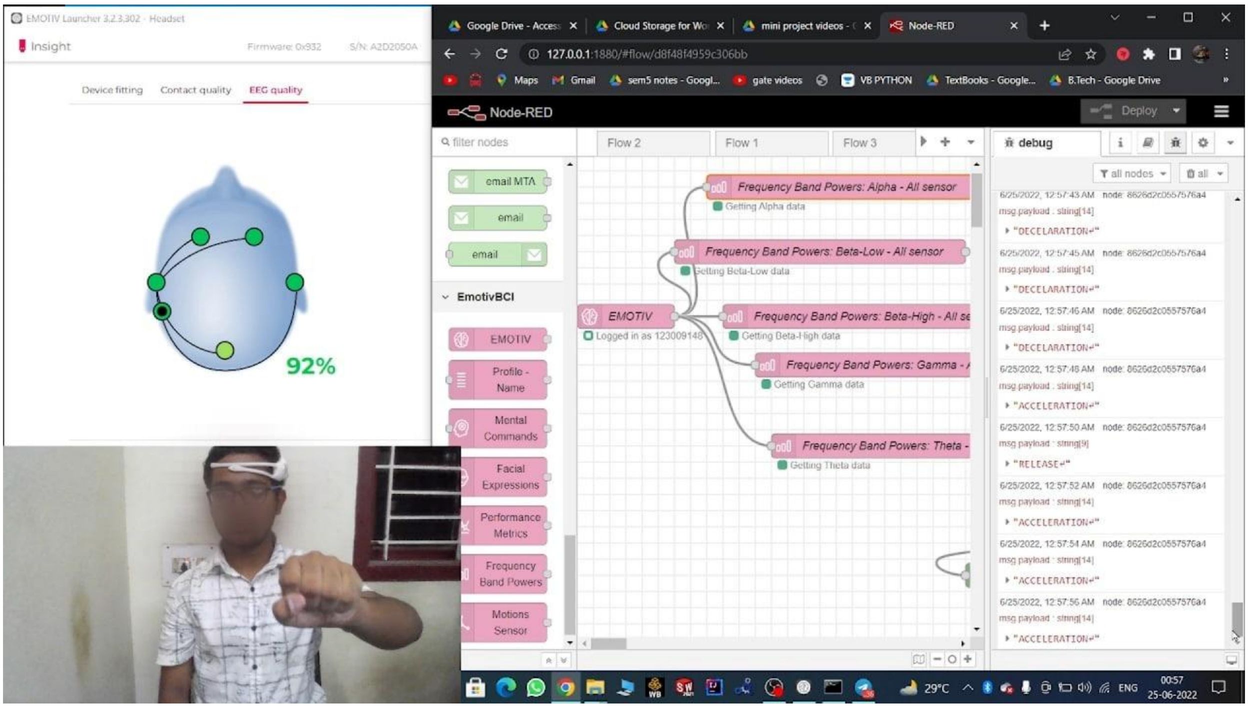


Figure 4.4.3 Displaying ACCELARATION action in Node-RED debug window

In the Figure 4.4.3 when the person is thinking about acceleration the action is displayed in Node-RED debug window.

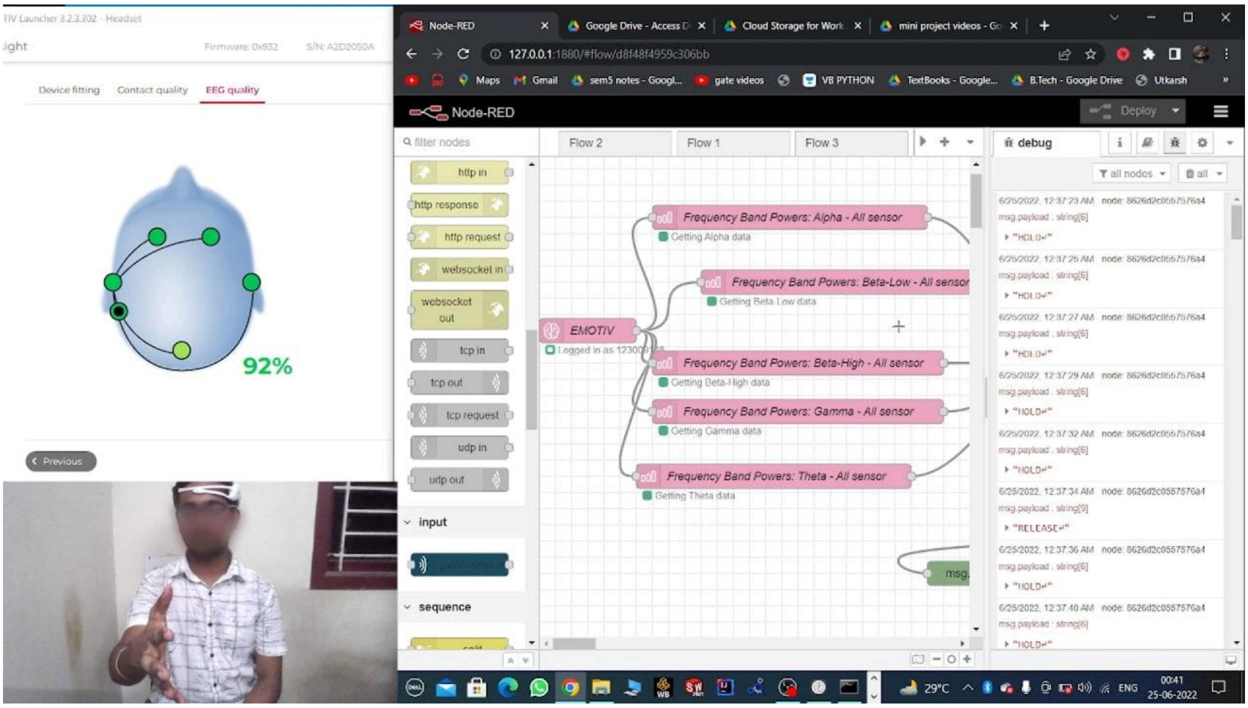


Figure 4.4.4 Displaying RELEASE action in Node-RED debug window

In the Figure 4.4.4 when the person is thinking about release the action is displayed in Node-RED debug window.

5. Establishing connection between Node-RED and Microcontroller

There are different ways to transfer data from Node-RED to a microcontroller such as Bluetooth, wi-fi etc. In this project data transfer is done using Wi-Fi. To establish connection between Node-RED and a Microcontroller we need a bridge that helps in sending data seamlessly. Blynk IoT application is used as a bridge between Node-RED and Microcontroller. The Figure 5.1 shows how the data transfer occurs.

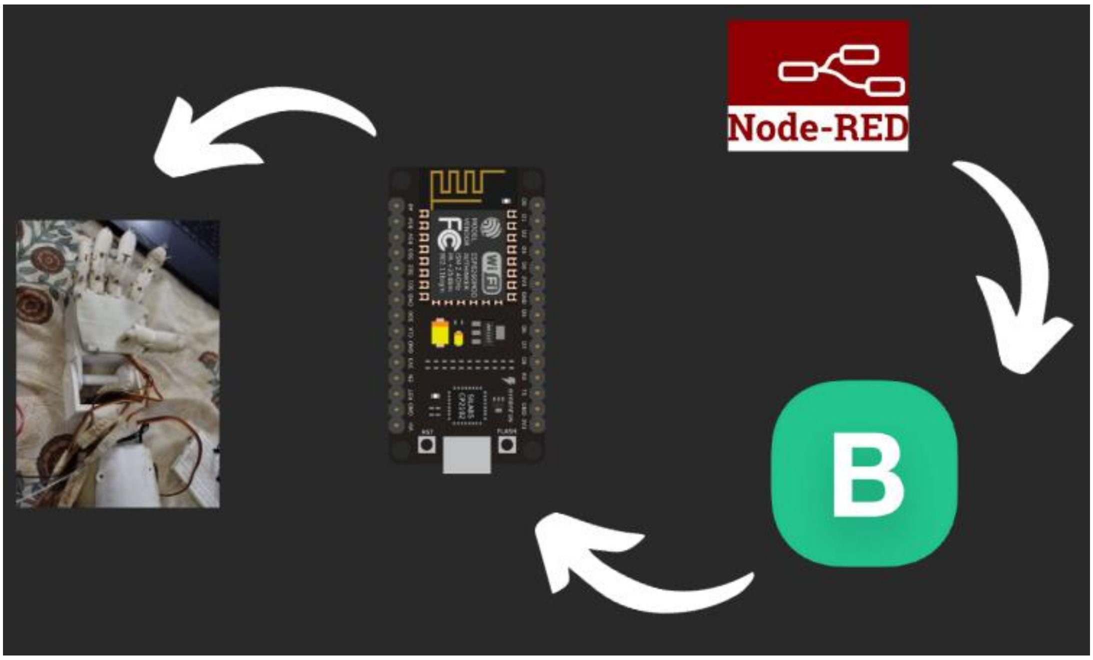


Figure 5.1 Data transfer flow diagram

5.1. Connecting Blynk with Node-RED

Node-RED sends a http request to the blynk server to update the action in the blynk so that microcontroller can access the action from blynk. The configuration of blynk is done in such a way that the string type of data i.e., action can be updated in its database and that can be retrieved whenever necessary by the microcontroller. We used NodeMCU (ESP8266) as our microcontroller which has a capability of sending and receiving the data using Wi-Fi.

5.1.1. Blynk Configuration

Blynk is configured with a virtual data pin for storing the action data that is received from Node-RED and then the data is accessed by the microcontroller from the same virtual pin.

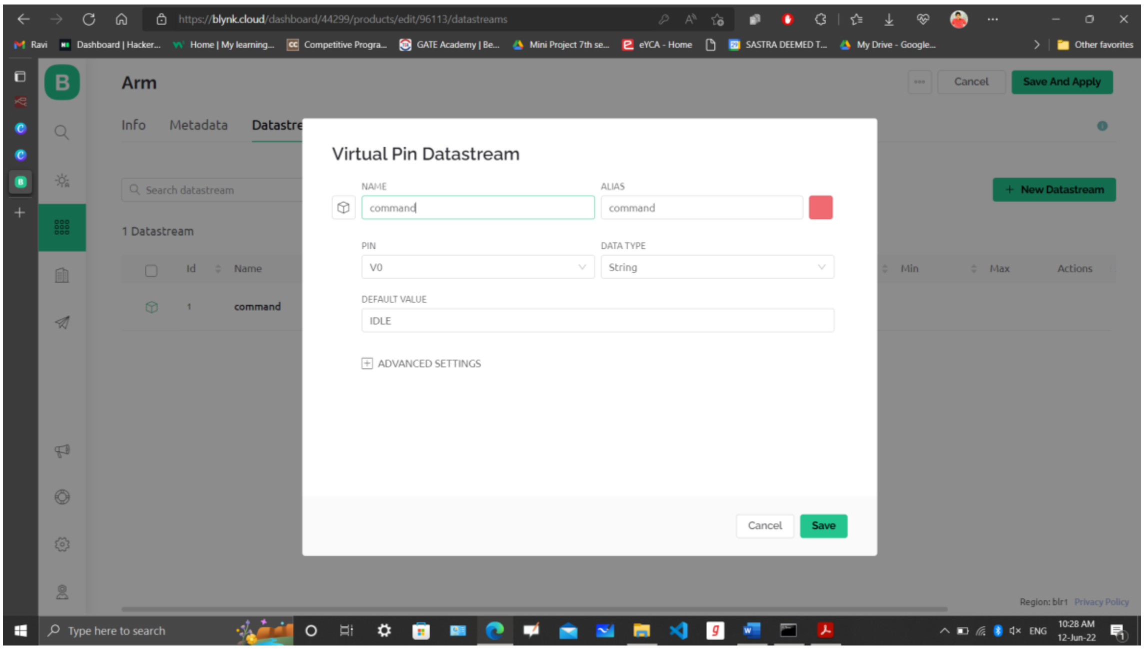


Figure 5.1.1.1 Virtual pin configuration in blynk

In the Figure 5.1.1.1 a virtual pin named as V0 is created and configured to take a datatype of string(action) from Node-RED. Initially when there is no data from Node-RED the blynk automatically assumes default action as IDLE.

5.1.2. HTTP Node Configuration

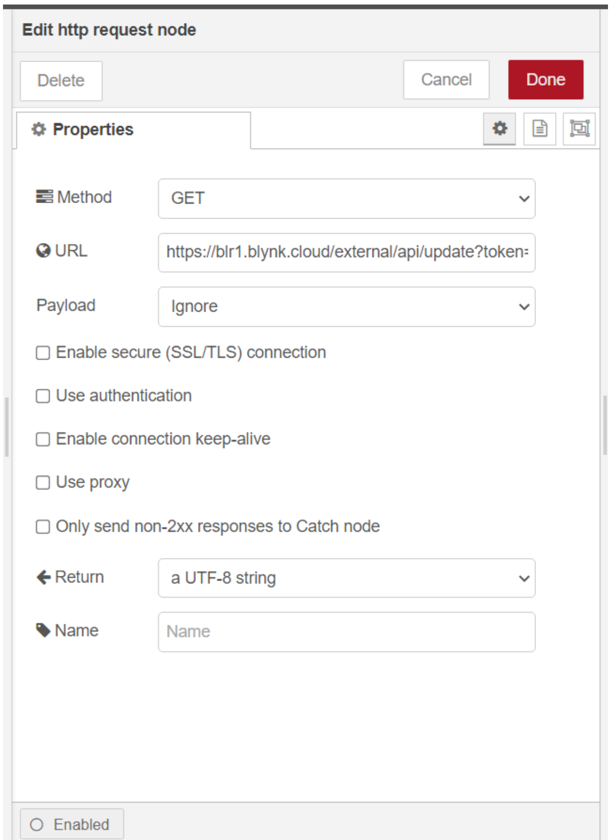


Figure 5.1.2.1 Configuration of http request node

The HTTP node is configured in such a way that the action from the python shell is transferred to blynk when a request is made to the blynk server using a URL as shown in the Figure 5.1.2.1.

5.2. Connecting Microcontroller with Blynk

Node-MCU has a Wi-fi module through which it can be connected to the internet and receive the data from the blynk server. As node-much requires internet to receive the data(actions) it is connected to mobile hotspot and then the data is retrieved using BLYNK_WRITE function in Arduino IDE.

```
BLYNK_WRITE(V0){
  command = param.asStr();
  Serial.print(command);
}
```

Figure 5.2.1 Function to get data from blynk

In this function shown in the Figure 5.2.1 param.asStr() function is used to get the data stored in V0 pin in blynk server.

6. Actuating the Bionic-Arm

In the bionic arm the microcontroller has to be programmed to actuate the servos. The microcontroller is programmed using the Arduino IDE. The logic of programming for each action in the microcontroller is explained below.

6.1. IDLE

For this action any servo in the Bionic-arm is not actuated.

6.2. ACCELERATION

For this action all the servos in the fingers are rotated to 180 degrees so that the strings connected to the fingers help in closing all the fingers and the servo located at the wrist is actuated upward.

6.3. DECELARATION

For this action all the servos in the fingers are rotated to 180 degrees so that the strings connected to the fingers help in closing all the fingers and the servo located at the wrist is actuated downward.

6.4. HOLD

For this action all the servos in the fingers are rotated to 180 degrees so that the strings connected to the fingers help in closing all the fingers.

7. Displaying classified actions on a cardboard arm using LED's

Since the bionic arm's servo motors and strings weren't working properly, a cardboard duplicate of the arm was created, and LEDs were blinking in place of the servos to show the movements**.**

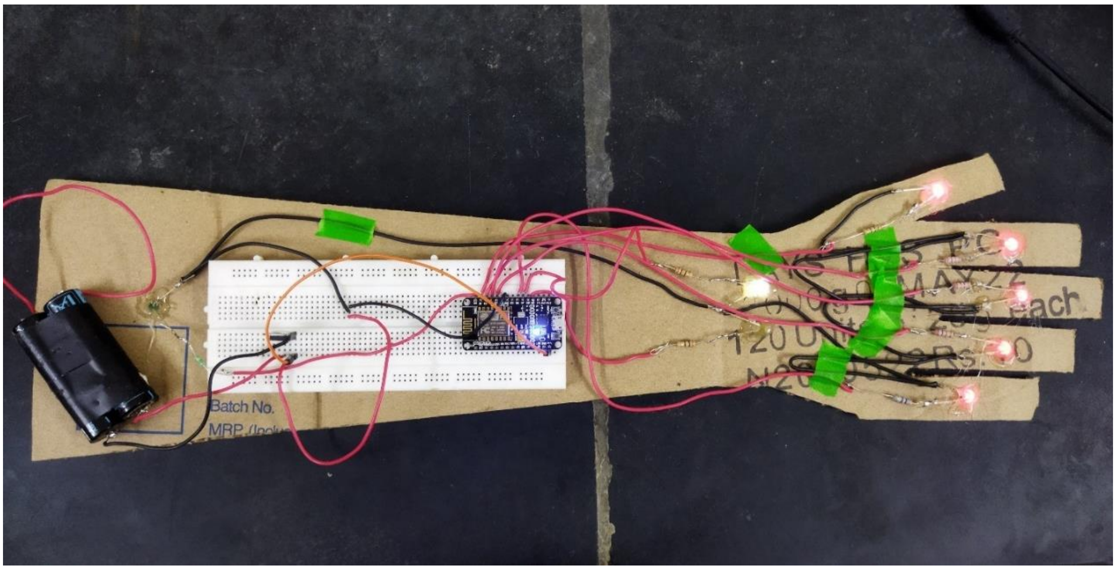


Figure 7.1 Cardboard duplicate of the arm

There are seven LEDs that are attached to the cardboard arm out of which five LEDs are attached to each finger and the other two LEDs are attached to the wrist a shown in the Figure 7.1.

The program in Arduino IDE is written in such a way that

- For the action called "hold", all the five LEDs on the fingers blink and the other two LEDs on the wrist will not blink as shown in the Figure 7.2

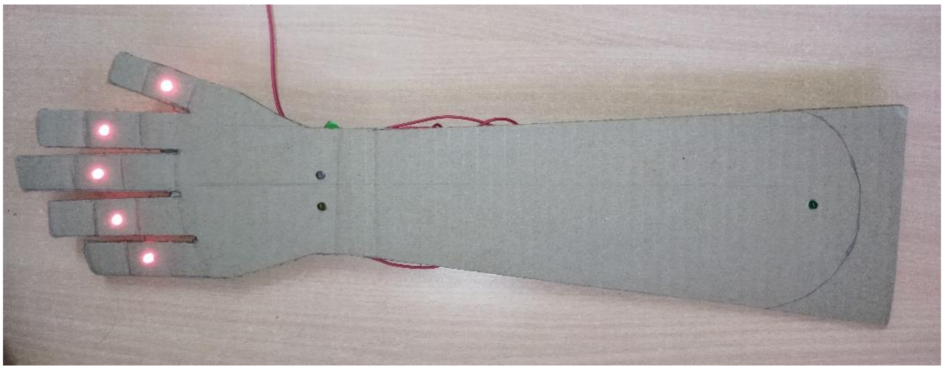


Figure 7.2 Hold

- For the action called "release", all the seven LEDs on the arm will not blink shown in the Figure 7.3

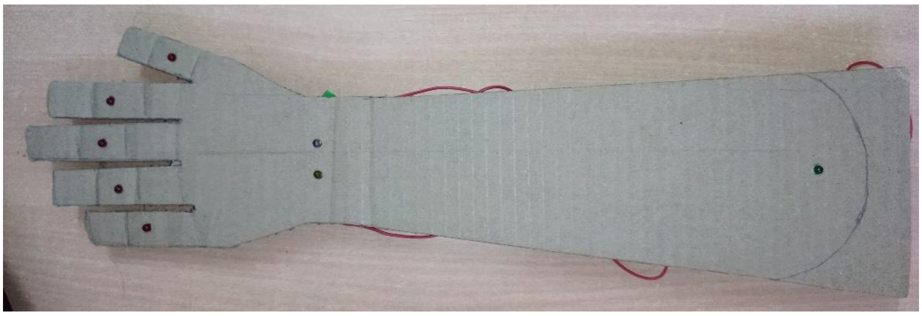


Figure 7.3 Release

- For the action called "Acceleration", all the five LEDs on the fingers blink and one LED on the left side of the wrist will blink shown in the Figure 7.4

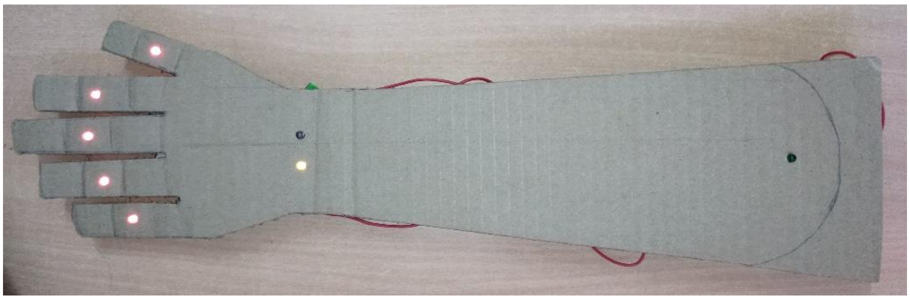


Figure 7.4 Acceleration

- For the action called "Deceleration", all the five LEDs on the fingers blink and one LED on the right side of the wrist will blink shown in the Figure 7.5

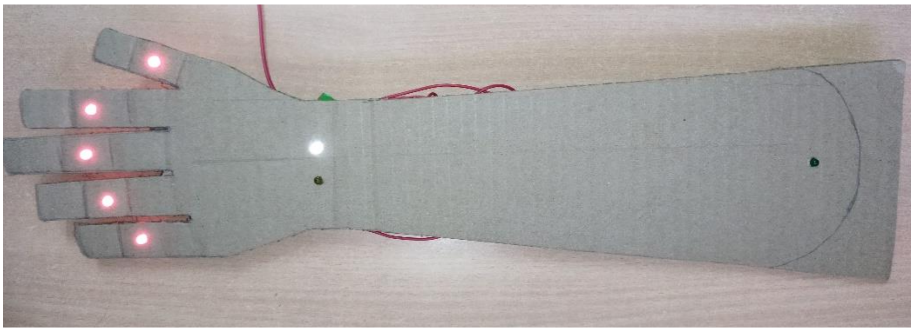


Figure 7.5 Deceleration

8. Results and Discussions

The actions have been classified successfully using data from emotiv sensor with an accuracy of 94%. The data has been successfully transferred from Node-RED to microcontroller (NodeMCU) wirelessly using internet. The output is used and the microcontroller is programmed successfully for actuating the bionic arm. As the servo motors and strings in bionic arm haven't functioned properly a carboard replica of arm is made and LEDs are blinked instead of servos actuation for depicting the actions. Final video on entire project working is embedded here.

9. Conclusions and Further Work

EMOTIV sensor is used for acquiring data for training, testing and real time predictions. The whole process of obtaining EEG frequency values of actions, real time predictions of actions are done in the software called Node-RED Toolbox. The ML model was able to classify the actions into those four actions as mentioned above. The data(actions) from Node-RED to microcontroller is successfully transferred. As part of further development getting lot of data for training the model and developing a machine learning model better than KNN will increase the quality of real time action predictions. A new bionic arm can be fabricated and configured to act accordingly for the classified actions.

10.References

[1]. ProsthBeyrouthy, T., Kork, S. A., Korbane, J. A., & Abouelela, M. (2017). EEG Mind Controlled Smart Prosthetic Arm – A Comprehensive Study. *Advances in Science, Technology and Engineering Systems Journal*, 2(3), 891–899.

[2]. Sunny T.D.a , Aparna T.b, Neethu P.c, Venkateswaran J.d*, Vishnupriya V.e, Vyas P.S.f , Robotic Arm with Brain and Computer Interfacing, *Department of Electronics & Communication Engineering, Govt. Engineering College, Thrissur*

[3]. Pattnaik, P. K., & Sarraf, J. (2018). Brain Computer Interface issues on hand movement. *Journal of King Saud University - Computer and Information Sciences,* 30(1), 18–24.

[4]. Christoff Heunis, Export and Analysis of Emotiv Insight EEG Data via EEGLAB, *University of Twente,* ResearchGate (2016).

[5]. Chean Khim Toa., Kok Swee Sim & Shing Chiang Tan, Emotiv Insight with Convolutional Neural Network: Visual Attention Test Classification, Springer Link (2021).

[6]. Mohd Nurul Al Hafiz Sha'Abani, kNN and SVM Classification for EEG: A Review, *Universiti Tun Hussein Onn Malaysia,* ResearchGate (2020).

[7]. X. Zhang, R. Li and Y. Li, "Research on brain control prosthetic hand," *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2014, pp. 554-557.

[8]. S. Ilgın and D. G. Duru, "Design and implementation of brain computer interface based on wireless EEG measurements: A pilot study," *2015 Medical Technologies National Conference (TIPTEKNO)*, 2015, pp. 1-4.

[9]. S. Wu *et al.*, "Design of the multi-channel electroencephalography-based brain-computer interface with novel dry sensors," *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 1793-1797.

[10]. A. Schuhknecht, E. Fadanelli, M. Patel, A. J. Hanson, D. Maddipatla and M. Z. Atashbar, "Development of a Flexible and Conformable EEG Sensors Using 3D Printing Process," *2021 IEEE Sensors*, 2021, pp. 1-4.

[11]. Miguel Pais-Vieira, Mikhail Lebedev, Carolina Kunicki, Jing Wang, and Miguel A. L. Nicolelis, A Brain-to-Brain Interface for Real-Time Sharing of Sensorimotor Information, PubMed Central® (PMC) (2013).

[12]. S. Bhavani; K. Lokesh Krishna; R.B. Yogananda Reddy; T. Geethika, A Low Cost Bionic Arm Based on Electromyography Sensor, IEEE Explorer (2020).

11.Peer Evaluation Form for Group Work

Name: Ravi Teja E

Evaluation Criteria	Group member: N S S Murali Prabhakar	Group member: Venkat Varun P
Attends group meetings regularly and arrives on time.	4	4
Contributes meaningfully to group discussions.	4	4
Completes group assignments on time.	4	4
Prepares work in a quality manner.	4	4
Demonstrates a cooperative and supportive attitude.	4	4
Contributes significantly to the success of the mini-project.	4	4
TOTALS	24	24

Feedback on team dynamics:

1. How effectively did your group work?

   Our group worked up to their threshold to yield the best possible results. We coordinated with each other in an effective manner by giving couple of hours daily to mini-project alone.

2. Were the behaviors of any of your team members particularly valuable or detrimental to the team? Explain.

   In our team every team members behaviour towards the project is valuable. We initially divided works among us and started working, once after accomplishing individual tasks we explained each other about our respective work so everyone is equally equipped with content.

3. What did you learn about working in a group from this mini-project that you will carry into your next group experience?

   Working in groups always helps us to improve few important skills like time management, team management, coordination, obeying each other’s opinion, commitment towards work. I have improved the above-mentioned skills which can be caried forward for future projects.

Sample Self Evaluation Form for Group Work

Name: Ravi Teja E

	Seldom	Sometimes	Often
Contributed good ideas			✓
Listened to and respected the ideas of others			✓
Compromised and cooperated		✓	
Took initiative where needed			✓
Came to meetings prepared			✓
Communicated effectively with teammates			✓
Did my share of the work			✓

My greatest strengths as a team member are:

- ➔ Motivating teammates every time
- ➔ Helping the team by sharing my knowledge in different domains
- ➔ Team and time management

The group work skills I plan to work to improve are:

- ➔ Get more inputs from members of the team for better solutions

Name: N S S Murali Prabhakar

Evaluation Criteria	Group member: Ravi Teja E	Group member: Venkat Varun P
Attends group meetings regularly and arrives on time.	4	4
Contributes meaningfully to group discussions.	4	4
Completes group assignments on time.	4	4
Prepares work in a quality manner.	4	4
Demonstrates a cooperative and supportive attitude.	4	4
Contributes significantly to the success of the mini-project.	4	4
TOTALS	24	24

Feedback on team dynamics:

1. How effectively did your group work?

   We created a work plan and worked accordingly every week. We divided tasks and worked on them individually and shared our ideas together. Frequently we came together and made progress in the project accordingly with time.

2. Were the behaviour of any of your team members particularly valuable or detrimental to the team? Explain

   Ravi Teja E found suitable way to transfer actions from Node-RED to microcontroller wirelessly and improved the machine learning algorithm
   Venkat Varun P was able to read several journals related to our project and shared few ideas and worked on EEG sensor

3. What did you learn about working in a group from this mini-project that you will carry into your next group experience?

   I came to know that a lot of knowledge can be gained and also shared when we work together as a group. The outcomes will be effective when we work as a team and works can be done on time.

Sample Self-Evaluation Form for Group Work

Name: N S S Murali Prabhakar

	Seldom	Sometimes	Often
Contributed good ideas		✓	
Listened to and respected the ideas of others			✓
Compromised and cooperated		✓	
Took initiative where needed			✓
Came to meetings prepared		✓	
Communicated effectively with teammates			✓
Did my share of the work			✓

My greatest strengths as a team member are:

- I was always willing to help my teammates whenever needed.
- I always tried to complete my portion of work without any delay.

The group work skills I plan to work to improve are:

- I should the ideas of teammates and agree with a better idea
- I should try to learn new things in a quick manner and work on them.

Name: Venkat Varun P

Evaluation Criteria	Group member: Ravi Teja E	Group member: N S S Murali Prabhakar
Attends group meetings regularly and arrives on time.	4	4
Contributes meaningfully to group discussions.	4	4
Completes group assignments on time.	4	4
Prepares work in a quality manner.	4	4
Demonstrates a cooperative and supportive attitude.	4	4
Contributes significantly to the success of the mini-project.	4	4
TOTALS	24	24

Feedback on team dynamics:

1. How effectively did your group work?

   Our group work was quite effective. Each of us had our own limitations and advantages but as a team we helped out each other in carrying the project forward and faced everything together.

2. Were the behaviors of any of your team members particularly valuable or detrimental to the team? Explain.

   Both of my teammates are responsible for the success of our project. The positive atmosphere that Ravi Teja brings to the meetings helps us to push an extra mile. His attitude and ideas towards the project always amaze us. On the other hand, Shiva who believes in the team and the project always inspired me to work harder. Both my teammates had never-give-up attitude which fueled to run forward.

3. What did you learn about working in a group from this mini-project that you will carry into your next group experience?

   This group has taught me how to coordinate and be a team player. It made me realize how coordination is important to function as a team. I also learnt how to face sudden challenges fearlessly during the project.

Sample Self Evaluation Form for Group Work

Name: Venkat Varun P

	Seldom	Sometimes	Often
Contributed good ideas		✓	
Listened to and respected the ideas of others			✓
Compromised and cooperated			✓
Took initiative where needed		✓	
Came to meetings prepared			✓
Communicated effectively with teammates			✓
Did my share of the? work		✓	

My greatest strengths as a team member are:

- ➔ Staying and facing challenges together
- ➔ Supporting my teammates in every decision we make
- ➔ Sharing my knowledge to my teammates

The group work skills I plan to work to improve are:

- ➔ To be more creative while suggesting ideas and plans.