A Project Report on

Multiclass classification of visual stimuli on EEG signals from single channel SSVEP-based brain computer interface.

By

Mr. SHANKAR BHARAMU PATIL

(220310125005)

Mr. ASHISH DALAL

(220310125016)

Mr. JANKI AHIRWAR

(220310125018)

Guide

Miss. SAUMYA KUSHWAHA

Project Engineer, CDAC Delhi

A Project Report Submitted

at

Centre for Development of Advanced Computing

In partial fulfillment of the requirements

of

PG Diploma in Big Data Analysis

[September 2022]

Centre for Development of Advanced Computing

ACT’S, Delhi

CERTIFICATE

This is to certify that the project entitled “**Multiclass classification of visual stimuli on EEG signals from single channel SSVEP-based brain computer interface.**” is a bonafide work of “**Mr. Shankar Bharamu Patil** (220310125005) **Mr. Ashish Dalal** (220310125016) **Mr. Janki Ahirwar (**220310125018)” **submitted** to C- DAC Delhi in partial fulfillment of the requirement for the award of the Post Graduate Diploma in Advanced Computing.

Miss. Saumya Kushwaha	Mr. Ankit Khurana
Guide	**Project Coordinator**

ACKNOWLEDGEMENT

Presentation inspiration and motivation have always played a key role in the success of anyventure. We express our sincere thanks to Mr. Ankit Khurana, Center for Development of Advance Computing (C-DAC), Delhi.

We pay our deep sense of gratitude to Mr. Ankit Khurana, Project coordinator, C-DAC,Delhi to encourage us to the highest peak and to provide us the opportunity to prepare the project. We are immensely obliged to our colleague for their elevating inspiration, encouraging guidance and kind supervision in the completion of our project.

We feel to acknowledge our indebtedness and deep sense of gratitude to our guide Miss. Saumya Kushwaha whose valuable guidance and kind supervision given to us.

ABSTRACT

Brain-computer interfaces (BCIs) provide humans a new communication channel by encoding and decoding brain activities. Steady-state visual evoked potential (SSVEP)-based BCI stands out among many BCI paradigms because of its non-invasiveness, little user training, and high information transfer rate (ITR). However, the use of conductive gel and bulky hardware in the traditional Electroencephalogram (EEG) method hinder the application of SSVEP-based BCIs. Besides, continuous visual stimulation in long time use will lead to visual fatigue and pose a new challenge to the practical application. This study provides an open dataset, which is collected based on a wearable SSVEP-based BCI system, and comprehensively compares the SSVEP data obtained by wet and dry electrodes. The paper presents a collection of electroencephalography (EEG) data from a portable Steady State Visual Evoked Potentials (SSVEP)-based Brain Computer Interface (BCI). The collection of data was acquired by means of experiments based on repetitive visual stimuli with four different flickering frequencies. The main novelty of the proposed data set is related to the usage of a single-channel dry-sensor acquisition device. Different from conventional BCI helmets, this kind of device strongly improves the users' comfort and, therefore, there is a strong interest in using it to pave the way towards the future generation of Internet of Things (IoT) applications. Consequently, the dataset proposed in this paper aims to act as a key tool to support the research activities in this emerging topic of human-computer interaction.

Keywords: Brain computer interface Steady state visual evoked potentials EEG signals Single-channel brain computer interface Internet of Things, Machine learning.

INTRODUCTION

Steady State Visual Evoked Potentials (SSVEP) are brain signals generated in the visual cortex area when focusing on an intermittent source of light, which is emitted at a specific frequency. Brain Computer Interfaces (BCIs) based on this paradigm are of growing interest in the scientific community due to the high information transfer rate and few training requirements. In current applications of BCI, SSVEP signals are usually acquired by means of helmets equipped with a collection of wet electrodes, where a saline solution is used to improve the capability of each sensor in capturing brain signals. Both the large number of sensors used in BCI helmets and the annoyance caused by the saline solution make these devices uncomfortable to be used in real world applications. For this reason, recently several research activities are focusing on innovative BCI devices based on few dry electrodes. As a consequence, there is a strong need to develop and publish new BCI datasets where aforementioned devices are used to capture brain signals. This paper bridges this gap. Indeed, the proposed dataset contains EEG raw data related to SSVEP signals acquired from eleven volunteers by using an acquisition equipment based on a single-channel dry-sensor recording device. The recorded EEG data from a single volunteer contains the response to an intermittent source of light, which is emitted at four different frequencies, namely 8.57 Hz (F1), 10 Hz (F2), 12 Hz (F3) and 15 Hz (F4). The considered frequencies that lie in the alpha and low-beta brain frequency bands. Each frequency stimulus has a duration of 16sec and was digitized at a fixed sampling frequency of 256 Hz. The EEG data of each volunteer was stored in a .csv file, represented in the format of "subject-N", where N represents the serial number of volunteers. Each .csv file contains four columns, named F1, F2, F3 and F4, corresponding to the four stimuli frequencies, respectively. Each column is composed of 4096 samples (corresponding to 16 s of signal length sampled at 256 Hz) whose values range from 0 to 1023.Each one of them displays data contained in one of the .csv files. In detail, each plot contains four subplots. Each subplot displays the values for one of the considered frequencies. Hence, each subplot is characterized by the number of samples on the x-axis and the frequency values on y-axis.

1.1. Objective: Brain-computer interfaces (BCIs) provide humans a new communication channel by encoding and decoding brain activities. EEG data Steady-state visual evoked potential (SSVEP)-based BCI. The recorded EEG data from a single volunteer contains the response to an intermittent source of light, which is emitted at four different frequencies. Each .csv file contains four columns, named F1, F2, F3 and F4, corresponding to the four stimuli frequencies, respectively.

Each column is composed of 4096 samples (corresponding to 16s of signal length sampled at 256Hz) whose values range from 0 to 1023. Implementation of different ML algorithm, predicted the possibly of getting the visuals stimuli displayed on the monitor consists of 4 alternating black and white squares with freq. 8.50Hz, 10Hz, 15Hz, 12Hz, respectively. We choose the best algorithm in terms of accuracy.

1.2. Scope

The scope of this project is to showed that BCI assists people living with disability to acquire relevant skills and knowledge, diagnose and manage depression, communicate, move and interact socially, as a useful tool to develop applications in the IoT area.

1.3 Problem Domain

BCI assists people living with disability to acquire relevant skills and knowledge, diagnose and manage depression, communicate, move and interact socially, as a useful tool to develop applications in the IoT area.

1.4 Solution Domain

To achieve possible solution for problem we first analyze the SSVEP BCI based dataset with the help of Machine learning algorithms for best accuracy to build a good Multiclass classification model and to develop applications in the IoT area of visual stimuli on EEG signals from single channel SSVEP-based brain computer interface.

1.5 System Requirements

1.5.1 Hardware Requirements

- Processor: Intel Core i5 or above
- RAM: 8 GB or more
- Video Card: Integrated Graphics (4 GB)

1.5.2 Software Requirements

- Python, Transfer Learning, ML
- Tools: Jupyter Notebook

Experimental Design, Materials and Methods

The hardware architecture of the experiment is shown in Fig. 2 and it is composed of three main parts: the acquisition unit, the processing unit, and the stimulating platform. The acquisition unit is based on the open source Olimen. EEG-SMT device shown in Fig. 3b, a two-channel differential input 10-bit analogue-digital converter (ADC) with 256 Hz sampling rate. In our experimentation, in order to consider a single-channel device, only one of these channels was used. In detail, one channel consists of two active electrodes (CH- and CH+) whose voltage difference is given in input to the ADC. These electrodes (see Fig. 3d-e) were arranged on a self-made 3D printed headset, shown in Fig. 3a, in order to better place them in the Fpz (Frontal Parietal area) and Oz (Occipital area) points according to the international 10–20 system. Moreover, a passive reference electrode was positioned on the earlobe (DRL). It is worth noting that Oz electrode has been modified by adding gold plated pins, shown in Fig. 3e, for a better contact with the scalp through the hair. The ADC provides the digitized data as an integer number ranging from 0 to 1023 with a dynamic range of ±0.39 mV due to the default internal gain of the device (G = 6427). Data is automatically transferred to the processing unit, namely a Raspberry Pi 3 minicomputer (see Fig. 3c), which runs software modules to store data to create the proposed dataset.

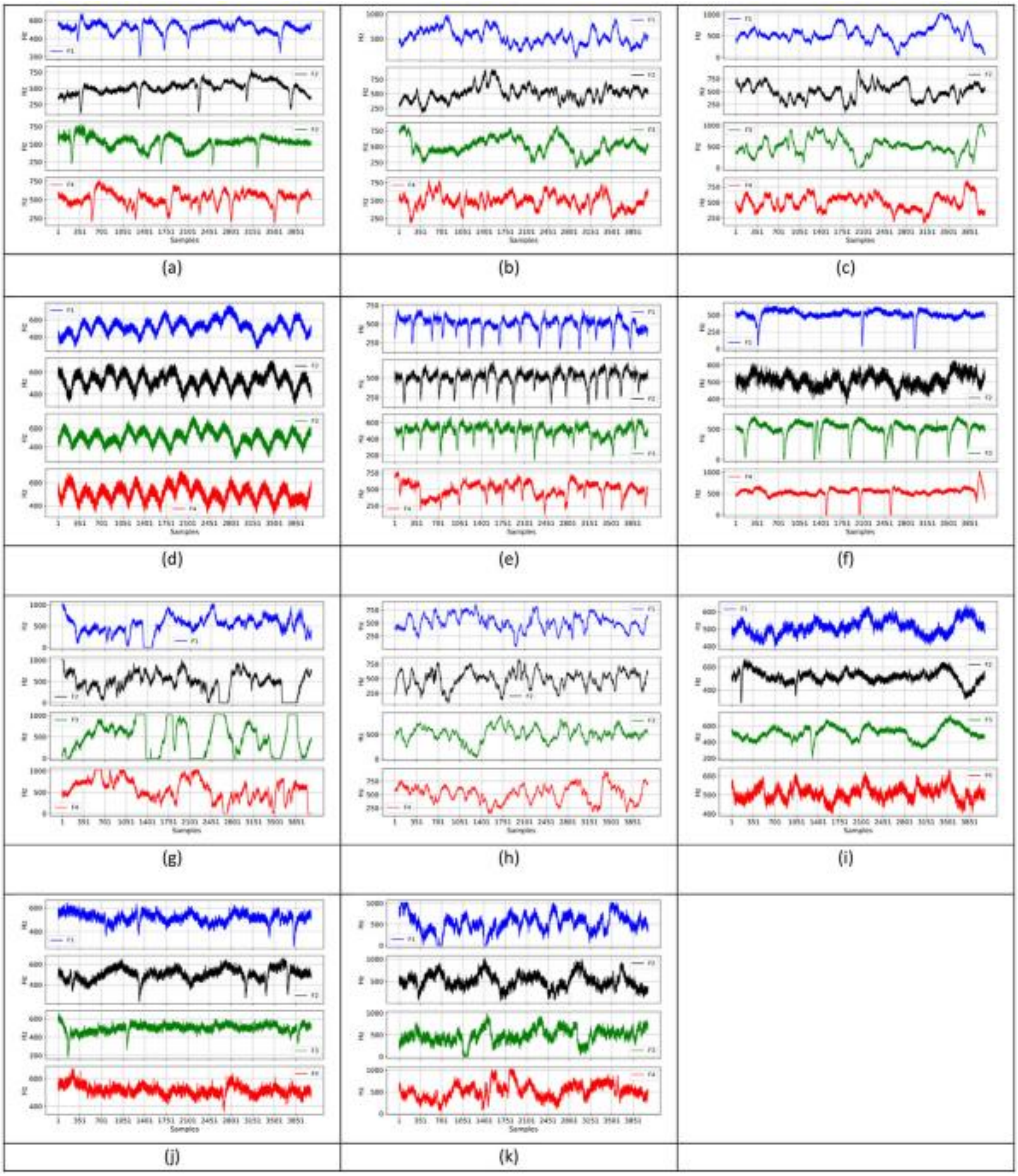


Fig. 1. (a) Data in the file subject1.csv; (b) Data in the file subject2.csv; (c) Data in the file subject3.csv; (d) Data in the file subject4.csv; (e) Data in the file subject5.csv; (f) Data in the file subject6.csv; (g) Data in the file subject7.csv; (h) Data in the file subject8.csv; (i) Data in the file subject9.csv; (j) Data in the file subject10.csv; (k) Data in the file subject11.csv.

The Olimex device and the Raspberry Pi are packed together with a power bank in a single portable unit characterised by 3 h of battery life. The stimulating platform was implemented on a 15.6-inch laptop (see Fig. 4). Visual stimuli consist of four alternating black white 80 × 80-pixel squares on a black background with oscillation frequencies of 8.57 Hz (F1), 10 Hz (F2), 12 Hz (F3) and 15 Hz (F4) respectively, compatible with 60 Hz monitor refresh rate [8]. The stimulating platform runs a software module that guides the user through the experiment

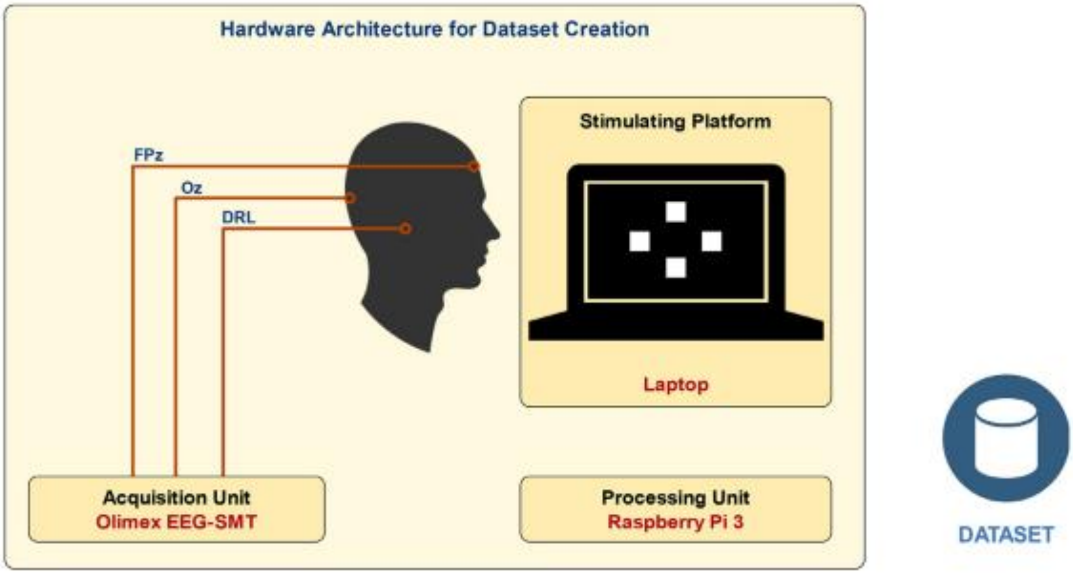


Fig. 2. Architecture of the system [1].

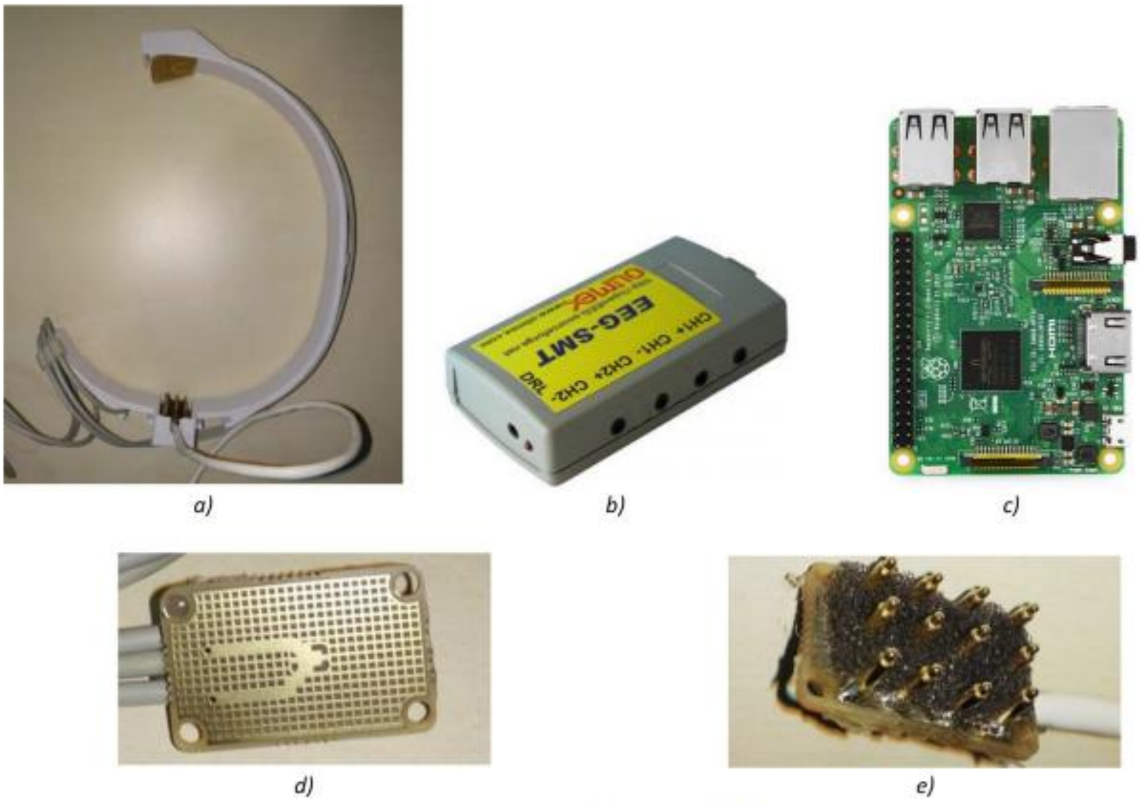


Fig. 3. (a) The 3D-printed headset; (b) the Olimex EEG-SMT device; (c) the Raspberry Pi 3 single-board computer; (d) the Fpz electrode; (e) the Oz electrode improved with gold plated pins.

The experiment was carried out on eleven healthy volunteers, aged from 25 to 50 years. Each volunteer was equipped with the acquisition headset, sat on a chair positioned at 70 cm away from the laptop monitor, and required to follow the steps below:

1. Focus on the stimulus related to the frequency F1 for 16 s.

2. Wait four seconds and relax before focusing on the next stimulus (during the 4 s break no stimuli were projected on the monitor);

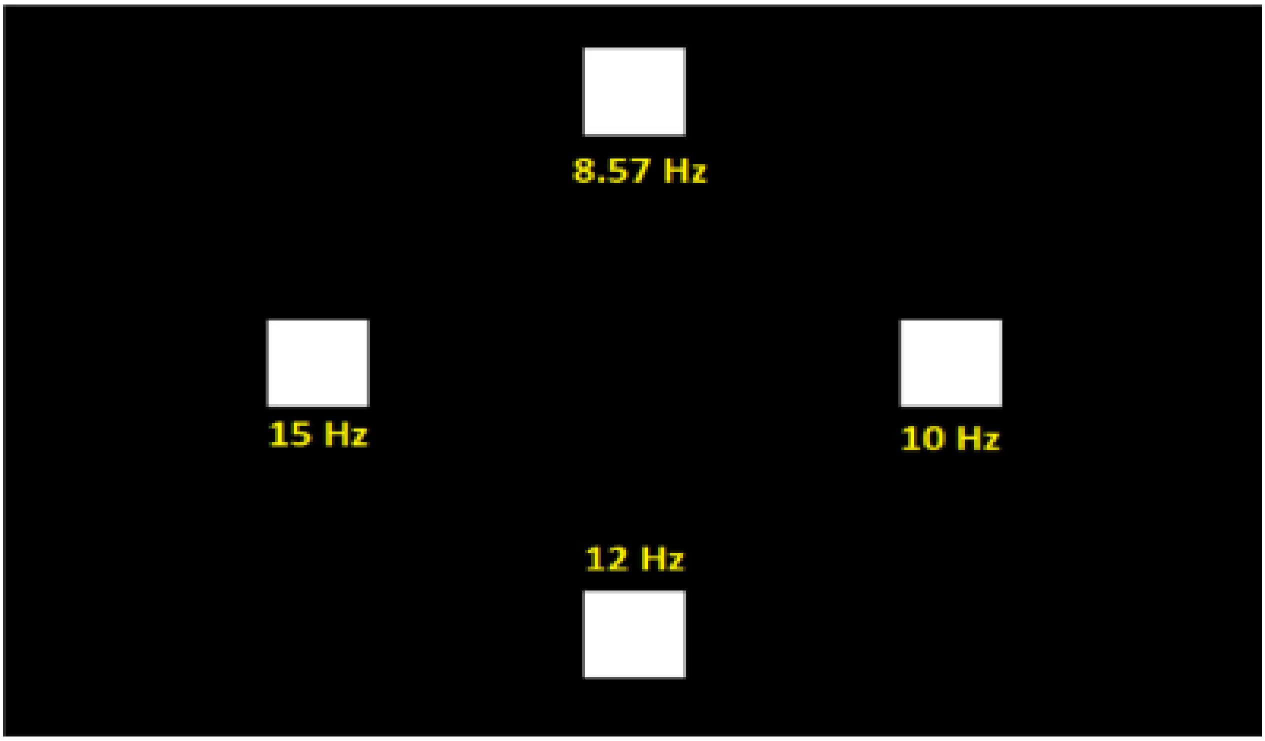
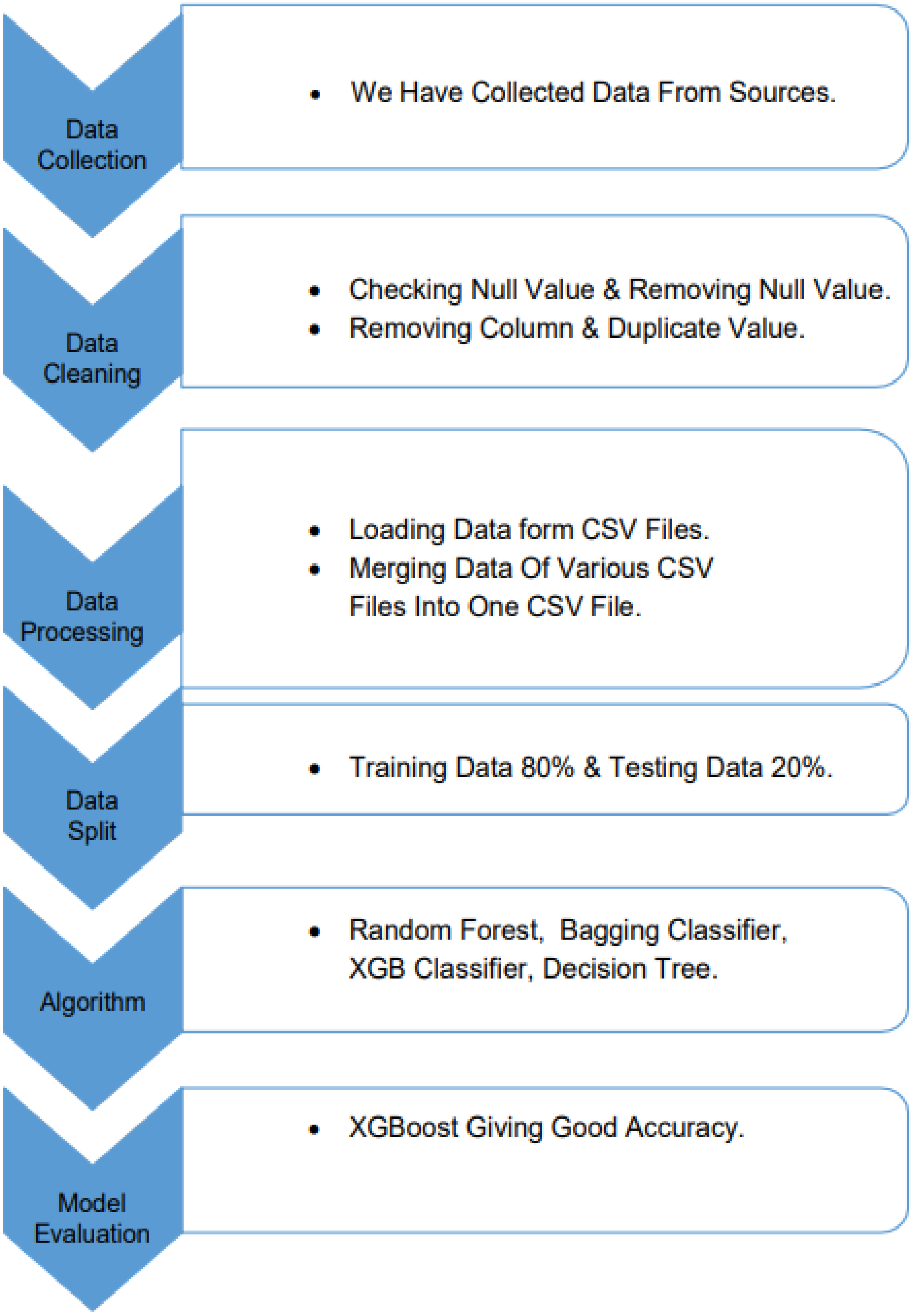


Fig. 4. Visual stimuli window showing the 4 flickering squares [1].

3. The steps 1 and 2 were repeated until data related to the three remaining stimuli (F2, F3, F4) was acquired.

4. At the end of the experiment, a total of 44 recordings of 16 s each one sampled at 256 Hz, for a total of 180,224 samples was obtained. In the future, the proposed dataset will be extended with more samples from different people in order to make it suitable to be used in both research and ready to market applications.

Project Work-Flow Diagram



Data Collection

- We Have Collected Data From Sources.

Data Cleaning

- Checking Null Value & Removing Null Value.
- Removing Column & Duplicate Value.

Data Processing

- Loading Data form CSV Files.
- Merging Data Of Various CSV Files Into One CSV File.

Data Split

- Training Data 80% & Testing Data 20%.

Algorithm

- Random Forest, Bagging Classifier, XGB Classifier, Decision Tree.

Model Evaluation

- XGBoost Giving Good Accuracy.

Python Libraries Used

1) **NumPy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

2) **Pandas:** Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks

3) **Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4) **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Machine Learning Libraries Used

1) **Scikit-**learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

2) **robust scaler-** Scale features using statistics that are robust to outliers. This Scaler removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

3) **Recursive Feature Elimination(RFE)**: is a popular feature selection algorithm. RFE is popular because it is easy to configure and use and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable.

4) **RandomizedSearchCV**: randomly passes the set of hyperparameters and calculate the score and gives the best set of hyperparameters which gives the best score as an output.

5) **GridSearchCV**: is a library function that is a member of sklearn's model_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set.

Model Used:

1. Logistic Regression
2. Random Forest
3. Extra trees classifier
4. Bagging Classifier
5. Decision Tree classifier
6. XGboost Classifier.

TOOLS USED

❖ Statistical Tools:

1) Exploratory Data Analysis:

- Descriptive statistics
- Bar charts

2) Machine Learning Algorithms (Data Mining Classifiers):

- Logistic Regression
- Random Forest
- Extra trees classifier
- Bagging Classifier
- Decision Tree classifier.

3) Validation Techniques: - K-cross validation

❖ Software:

1) PYTHON

2) MS-EXCEL

3) JUPYTER NOTEBOOK

Data Mining Classifiers

Model Building: -

After data preprocessing, we use Logistic Regression, Random Forest, Extra trees classifier, SVM, Bagging Classifier, Decision Tree classifier, KNN classifier algorithms by using Python Software. This model has been selected for this study because of their popularity in therecent literature. Different classifiers are used below and their performances measured.

Logistic Regression:

Logistic regression is the generalization of linear regression. It is used primarily for predicting binary or multi class dependent variables. Because the response variable is discrete, it cannot be modeled directly by linear regression. While logistic regression is a powerful modeling tool, it assumes that the response variable is linear in the coefficients of the predictor variable.

Decision Tree:

Decision tree are powerful classification algorithm that are becoming increasing morepopular with the growth of data mining in the field of information systems. In the literature there are so many popular decision tree algorithms like ID3, C4.5, and C5 etc. As the name implies, this technique recursively separates observations in branches to construct a tree for the purpose of improving the prediction accuracy. In doing so, they use mathematical algorithms like information gain, Gini index to identify a variable and corresponding threshold for the variable that splits the input observation into two or more subgroups. This step is repeated at each leaf node until the complete tree is conducted. The objective of the splitting pair that maximizes the homogeneity of the resulting two or more subgroup samples.

Random Forest Classifier:

Random Forest or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees‟ habit of overfitting to their training set.

Extra Tree Classifier:

Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees.

XGBOOST:

XGBoost is an implementation of Gradient Boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

Confusion Matrix:

A confusion matrix contains information of actual and predicted classification done bya classification model. The performance of such system is commonly evaluated usingthe data in the matrix. Predicted Actual Negative Positive Negative TN FP Positive FNTP

Measures of performance evaluation

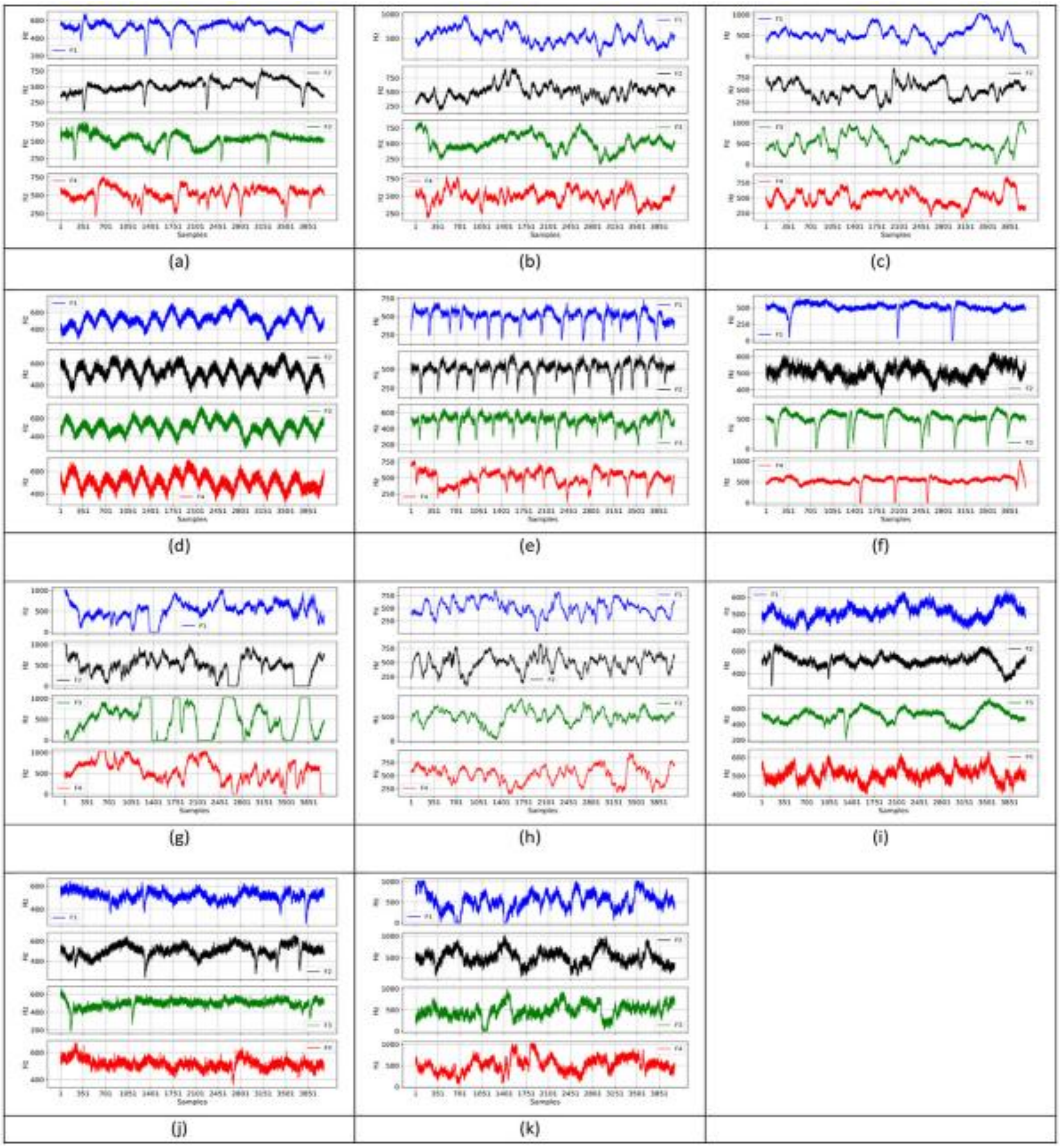
1) Accuracy = (TP+TN)/(TN+TP+FN+FP) i.e. It is the % of correct classification by the classifier.

2) Recall = TP/(TP+FN) i.e. Proportion of correct positive classification (True positives) from cases that are actually positive.

3) Precision = TP/(TP+FP) i.e. Proportion of correct positive classification (True positive) from cases that are actually positive.

4) F1-Score = 2TP/(2TP+FP+FN) i.e. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

“Exploratory Data Analysis”



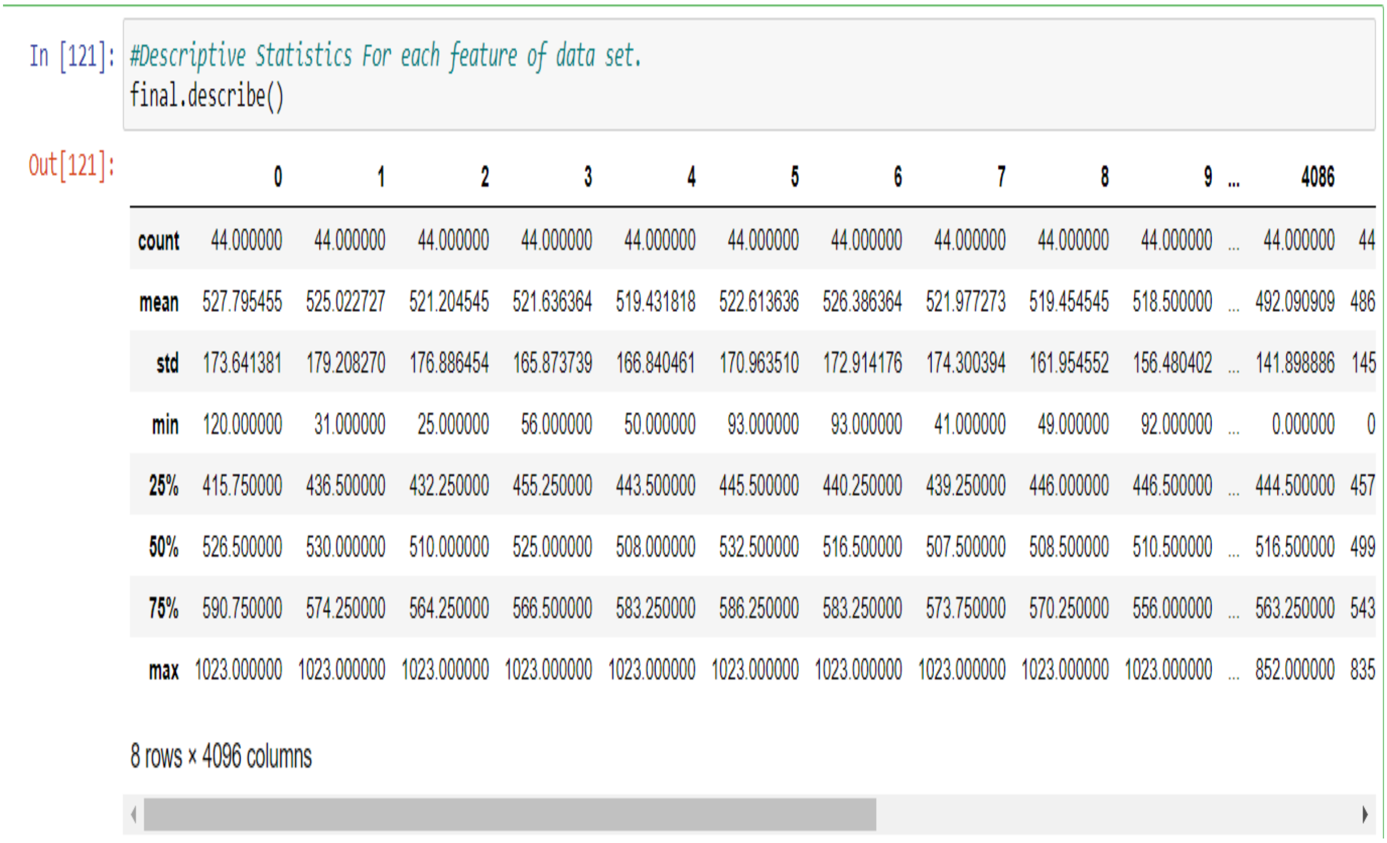
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Fig. 1. (a) Data in the file subject1.csv; (b) Data in the file subject2.csv; (c) Data in the file subject3.csv; (d) Data in the file subject4.csv; (e) Data in the file subject5.csv; (f) Data in the file subject6.csv; (g) Data in the file subject7.csv; (h) Data in the file subject8.csv; (i) Data in the file subject9.csv; (j) Data in the file subject10.csv; (k) Data in the file subject11.csv.

Observations:

Each one of them displays data contained in one of the .csv files. In detail, each plot contains four subplots. Each subplot displays the values for one of the considered frequencies. Hence, each subplot is characterized by the number of samples on the x-axis and the frequency values on y-axis.

Descriptive Statistics Of dataset:



```
In [121]: #Descriptive Statistics For each feature of data set.
          final.describe()
```

Out[121]:

	0	1	2	3	4	5	6	7	8	9	...	4086	
count	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	44.000000	...	44.000000	44
mean	527.795455	525.022727	521.204545	521.636364	519.431818	522.613636	526.386364	521.977273	519.454545	518.500000	...	492.090909	486
std	173.641381	179.208270	176.886454	165.873739	166.840461	170.963510	172.914176	174.300394	161.954552	156.480402	...	141.898886	145
min	120.000000	31.000000	25.000000	56.000000	50.000000	93.000000	93.000000	41.000000	49.000000	92.000000	...	0.000000	0
25%	415.750000	436.500000	432.250000	455.250000	443.500000	445.500000	440.250000	439.250000	446.000000	446.500000	...	444.500000	457
50%	526.500000	530.000000	510.000000	525.000000	508.000000	532.500000	516.500000	507.500000	508.500000	510.500000	...	516.500000	499
75%	590.750000	574.250000	564.250000	566.500000	583.250000	586.250000	583.250000	573.750000	570.250000	556.000000	...	563.250000	543
max	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	...	852.000000	835

8 rows × 4096 columns

Machine Learning Algorithm

Performance of the above classifier is as follows:

Accuracy chart of models:

Model Name	**Testing Accuracy**
XGBoost Classifier	**0.55**
Logistic Regression	**0.44**
Random Forest	**0.22**
Extra trees classifier	**0.11**
Bagging Classifier	**0.44**
Decision Tree classifier	**0.15**

Observations:

Out of the above only xgboost classifiers gives us more accuracy. Now we use only xgboost model for further also try the Recursive Feature Elimination method to achieve good accuracy.

Recursive Feature Elimination:

No. OF Features	Testing Accuracy with XGBOOST Classifiers
4000	**0.56**
3000	**0.44**
2000	**0.33**
1000	**0.22**

Observations:

The above tables show the accuracy corresponding with No. of features. Here we can see that as we take more features for model building it gives better accuracy. For 4000 features Xgboost gave us 56% of accuracy.

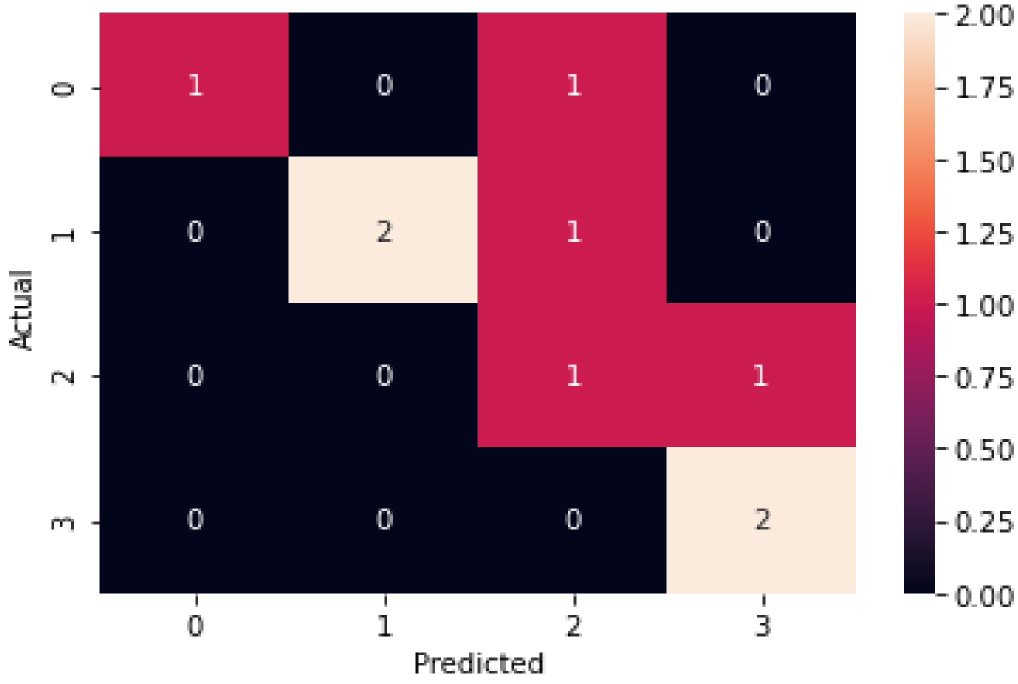
Best Parameters with RandomizedSearchCV:

❖ Xgboost Classifier:

Classification Report:

Labels	precision	recall	f1-score	support
0	1.00	0.50	0.67	2
1	1.00	0.67	0.80	3
2	0.33	0.50	0.40	2
3	0.67	1.00	0.80	2
Accuracy	0.50	0.67	0.67	9
macro avg	0.50	0.67	0.67	9
weighted avg	0.59	0.67	0.68	9

Confusion matrix:



K-cross validation:

Mean	0.200000
Standard deviation	0.145686
Min	0.000000
Q1	0.142857
Q2	0.142857
Q3	0.285714
Max	0.428571

Interpretation

68% of predicted values are correctly classified with 32% misclassification rate by the Decision Tree classifier. The average accuracy is 58%.

Observations:

From the above output of cross-validation we can see that the standard deviation between scores is high. Because of that every time it gives us different accuracy.so we can say still xgboost giving good accuracy but variance of accuracy is high which is not good for us.

Major Finding:

1. We have applied machine learning algorithms such as logistic Regression, Random Forest, Extra-tree classifier, Bagging classifier, Decision tree classifiers, XGboost classifier without any model improvement.

2. Then we used RFE with XGboost model, in that we have seen XGboost model gives better result with number of features= 4000.

3. Again, for finding best parameter we used Randomized search cv method for XGboost, then we get an improved accuracy = 67%.

4. Finally for model validation we have used K-fold cross validation, after that we have seen standard deviation of the accuracy score is very high (i.e., variability of the score is very high) and because of that whenever we will fit the model by using best parameter it will give us deviated accuracy scores.

Limitations:

With result of Cross- validation, as in this project we have less amount of data (row = 44, features = 4096) because of that model gave us train data accuracy = 100%, test data accuracy = 66%. this scenario is clearly showing problem of overfitting.

To overcome this overfitting problem, we should collect more data subjects.

References

1) G. Acampora, P. Trinchese, A. Vitiello, Applying logistic regression for classification in single-channel SSVEP-based BCIs, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2019.

2) A.M. Norcia, L. Gregory Appelbaum, J.M. Ales, B.R. Cottereau, B. Rossion, The steady-state visual evoked potential in vision research: a review, J. Vis (2015)

3) D. Anwar, P. Garg, V. Naik, A. Gupta, A. Kumar, Use of portable EEG sensors to detect meditation, in: Proceedings of the International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, 2018.

4) H. Hinrichs, M. Scholz, A.K. Baum, et al., Comparison between a wireless dry electrode EEG system with a conventional wired wet electrode EEG system for clinical applications, Sci. Rep. 10 (2020).

5) M. Ogino, S. Kanoga, M. Muto, Y. Mitsukura, Analysis of prefrontal single-channel EEG data for portable auditory ERP-based brain–computer interfaces, Front Hum. Neurosci. 13 (2019).

6) S.J. Johnstone, R. Blackman, R., J.M. Bruggemann, EEG from a single-channel dry-sensor recording device, Clin. EEG Neurosci (2012) 112–120.

7) V. Odom, M. Bach, M. Brigell, G.E. Holder, D.L. McCulloch, A.P. Tormene, Vaegan, ISCEV standard for clinical visual evoked potentials (2009 update), Documenta Ophthalmologica (2010).

8) H. Cecotti, I. Volosyak, A. Graser, Reliable visual stimuli on LCD screens for SSVEP based BCI, in: Proceedings of the European Signal Processing Conference, 2010