

Hand and Leg Movement Prediction using EEG Signal by Stacked Deep Auto Encoder

Kulsheet Kaur Viridi
P.G. Student

Department of Computer Science & Engineering
Samrat Ashok Technological Institute
Vidisha, India
kulsheetkaur@gmail.com

Satish Pawar
Professor

Department of Computer Science & Engineering
Samrat Ashok Technological Institute
Vidisha, India

Abstract- Brain Computer Interface (BCI) is device that enables the use of the brain's neural activity to communicate with others or to control machines, artificial limbs, or robots without direct physical movements. Brain-computer interfacing is an uprising field of research wherever signals extracted from the human brain are used for deciding and generation of control signals. Selection of the most appropriate classifier to find the mental states from electroencephalography (EEG) signal is an open research area due to the signal's non-stationary and ergodic nature.

In this research work the proposed algorithm is designed to solve an important application in BCI where left hand forward-backward movements and right hand forward-backward movements as well as left leg movement and right leg movement are needed to be classified. Features are extracted from these datasets to classify the type of movements. A stacked Deepauto encoder is used for classification of hand and leg movements and compared with other classifiers. The accuracy of stacked deepauto encoder is better with respect to other classifiers in terms of classification of hand and leg movement of EEG signals.

Keywords: BCI, Non-Invasive, EEG, Feature Extraction, Stacked, Encoder, Deep learning, Accuracy.

I. INTRODUCTION

A BCI (Brain-Computer Disability Interface) technology is a means of communication that allows people with simple immunity to communicate with peripheral devices that use EEG or other brain signals. The brain is made up of over 100 billion neurons. The BCI scheme should be able to classify the different EEG signals of the brain activity as precisely as possible, and the BCI user should learn how to generate separate brain signals to perform the different task. BCI has developed a synergistic combination of computational neuroscience, physiology, engineering, signal processing, computer science and different types of interdisciplinary research [1]. Research groups have focused on many areas of light and television control, yes / no questions, word processing, wheelchair control, robotic prostheses, autonomous vehicles, etc [2],[3].

The promising future foreseen for BCI has inspired analysis community to review the involvement of BCI

within the lifetime of non-paralyzed humans through medical applications. However, the scope of analysis has been any widened to incorporate non-medical applications. newer studies have targeted traditional people by exploring the employment of BCIs as a completely unique data input device and investigation the generation of hands-free applications [4].

Motor neural activity of the brain are often translated into a real movement activity which might be wont to control a prosthetic device. By victimization graphical record recording machine, brain signal are often recorded from the surface of the human scalp. Necessary options are often extracted victimization mathematical formula from the recorded signal and translated into real activity movement command. The strategy of changing the brain signal into a command signal is understood as Brain computer Interfacing (BCI). BCI is outlined as a system that produces an impression output through a conversion method of electrical brain signal as associate signal originating from the brain psychological feature processes. The management signal are often applied to regulate associate application like computer control and mechanical prosthetic device [5].

Study on the brain machine interface (BMI) using motor mental imagery still provides an open discussion among the researchers. Numerous feature extraction and classification strategies were applied by the early researchers to investigate the electroencephalogram signal [6],[7]. The objective of this paper is not only to study neurophysiological understanding of the human brain but also to investigate electroencephalography as a means of identifying mental activity. This paper reviews on EEG based BCI systems that are implemented and also compared the performance of different feature classification techniques and puts forth best classification methods for hand and leg movement.

II. RELATED WORK

Hema et al. [8] proposed Elman neural network and Functional Link neural network are employed using principle component analysis (PCA) features.

Marquez L. et al. [9] classified four arm movements such as left arm forward-backward movements and right arm forward backward movements using discrete wavelet transform (DWT) coefficients of corresponding EEG signals and multilayer perceptron neural network (MPNN). Researchers identified four movements of the arms of a healthy person: right hand back and forth, left hand back and forth. The accuracy obtained was 88.72%. There is lack of precision of the system in the final tests.

Zhiwei et al. [10] used wavelet packet entropy of EEG signal as feature where SVM is used as classifier.

Ting et al. [11] used wavelet packet transform and Probabilistic Neural Network (PNN) for classification. Since wavelet packet transform decomposes a signal into different frequency bands, the information about the frequencies of the EEG signal is distributed in several levels. However, to obtain more frequency information of EEG signal, Fourier analysis is carried out on sub band components. Statistical features from pre-processed EEG signal are obtained using wavelet packet decomposition and Fourier transform. KNN classifier is used to classify hand movements and compared to artificial neural network (ANN) which has many parameters to control.

Syed Khairul Bashar and Mohammed Imamul Hassan Bhuiyan [12], proposed a method to classify arm movements using statistical features of electroencephalogram (EEG) signals calculated from wavelet packet and Fourier transforms and classified using K-nearest neighbor (KNN)-based classifiers to identify the arm movements, right hand forward and backward; left hand forward and backward. A mean accuracy of 92.84% is achieved.

Saugat Bhattacharyya et al.[12], introduced interval type-2 fuzzy system within the fray to enhance its uncertainty handling. Also, real time situations need a classifier to identify 2 mental states. Thus, a multi-class discriminating formula supported the fusion of interval type-2 mathematical logic and ANFIS is used in this research work. Two variants of this algorithm are developed on the idea of One-Vs-All and One-Vs-One strategies [14]. Each of the variants are tested on an experiment involving the real time robotic arm control, wherever each the variants of the proposed classifier, produces an average success rate of reaching a target to 65% and 70% respectively.

Prasant KumarPattnaik et al. [15] discussed how Brain Computer Interface has helped severely disabled patients. In this research, a small set of EEG signals were generated for various movements of the fingers and hands, and artifacts were removed using a low-pass filter technique and compared to the original data sets, which yielded results satisfactory. The frequencies of the alpha and beta bands were extracted from the EEG

signal using the discrete wavelet transform and were therefore considered advanced for signal classification. This work intended for left hand movement and right-hand movement classification and obtained approx. 50 RMSE. In this paper there is no focused on feature extraction as well as classification Accuracy is not mentioned.

Muhammed Al-Suify et al. [16] with an aim to provide an effective system for classifying left and right movements of hand. The scheme is based on the combination of linear and non-linear characteristics to improve the classification rate. The linear characteristics are extracted by amplitude frequency (AFA) of the EEG signal. While non-linear features are extracted from a density matrix generated by the signal phase space. Four classification approaches in this study are used which are linear discriminant analysis (LDA), support vector machine (SVM), Bayes and KNN classifiers. The 2003 Graz data sets were used in this study. The maximum classification rate reached is 89.3%. The results confirmed the robustness of the new technique and demonstrated its value as a classification approach in the field of the BCI computer interface.

It is observed through literature is that hand and leg movement classification is possible using EEG signals. Extracting important features and use of various classifiers with more statistical features and use of classifier for better result can be obtained.

III. PROPOSED METHODOLOGY

The aim of this research is to classify different tasks of hand as well as leg movements as illustrated in figure 1. The proposed methodology is discussed in four steps. The first step is involves acquired EEG data. The second step was signal Preprocessing, to remove the noise and unwanted data. After noise removal the third step includes feature extraction from the EEG signals. The fourth step is to classify EEG signals to the corresponding movements i.e. hand and leg movements. Flow chart in figure 1 provides the proposed methodology which is discussed in details in section 3.1.

3.1 Signal Acquisition

In this paper, for simulation of proposed methodology, dataset is taken from the BCI project for EEG signal of 21-year-old man which include left hand right hand and leg movement signals. The data taken for simulation are following:

- Left hand forward movement
- Left hand backward movement
- Right hand forward movement
- Right hand backward movement
- Left leg movement
- Right leg movement

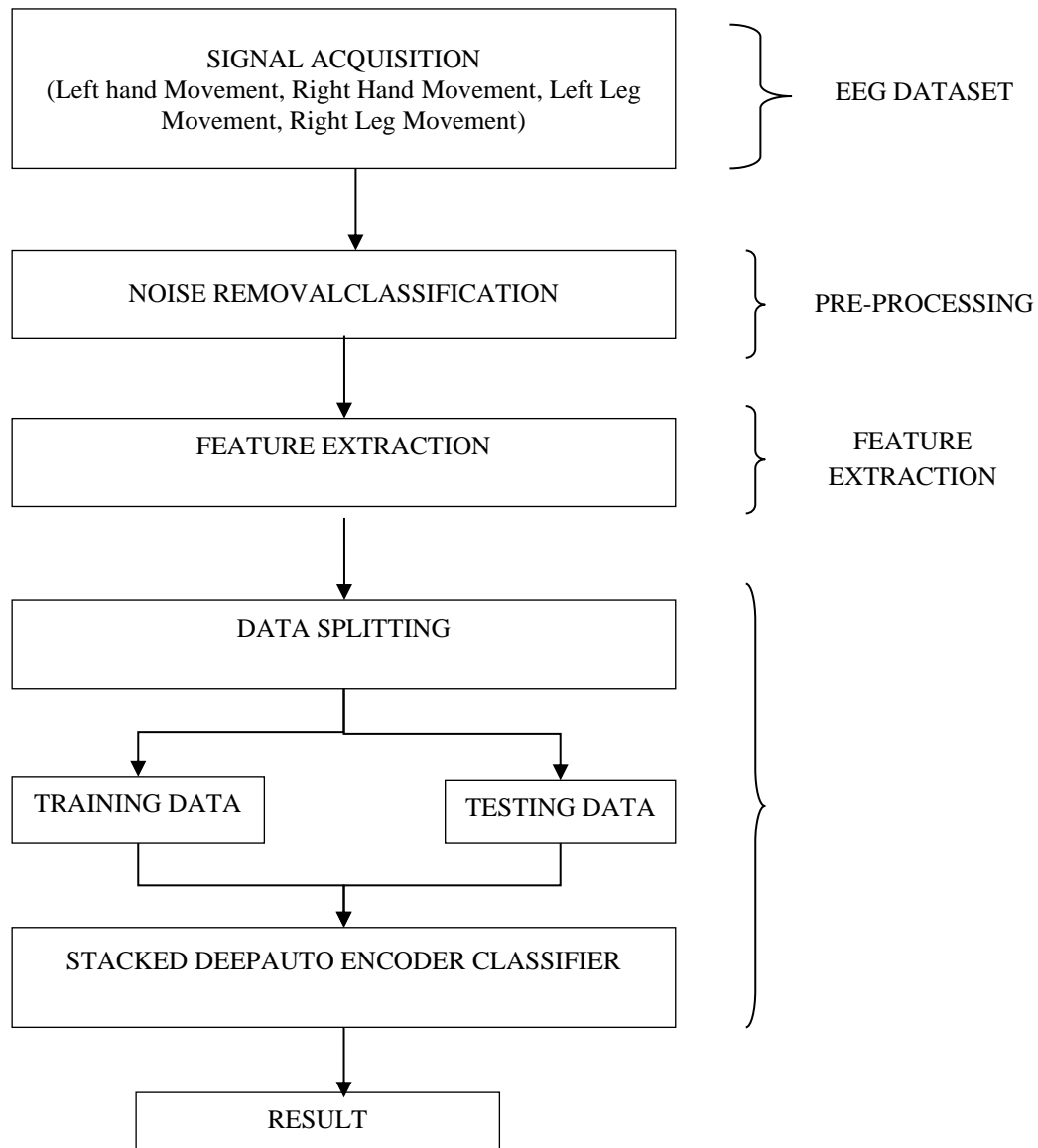


Figure 1: Proposed methodology analysis of EEG signals

3.2 Pre-processing

EEG data acquired is usually noisy and therefore call for noise removal. The EEG signals were filtered with band pass filter between 8 and 25Hz using a Butterworth filter to remove the unwanted artifacts. A band pass filter allows a predetermined band of frequencies to pass through. This is achieved by connecting a high pass filter with a low pass filter.

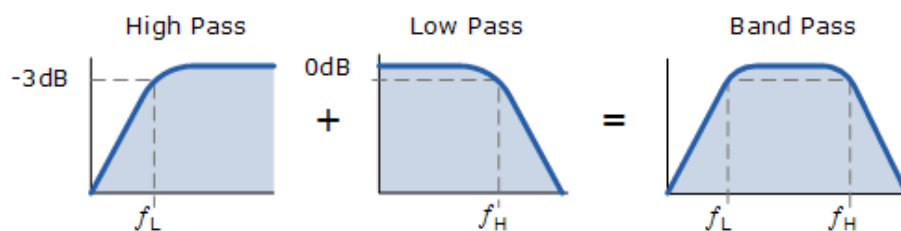


Figure 2: Bandpass Filter

The higher corner point (f_H) as well as the lower corner frequency cut-off point (f_L) are calculated the same as before in the standard first-order low and high pass filter circuits. Obviously, a reasonable separation is required between the two cut-off points to prevent any interaction between the low pass and high pass stages. The amplifier also provides isolation between the two stages and defines the overall voltage gain of the circuit.

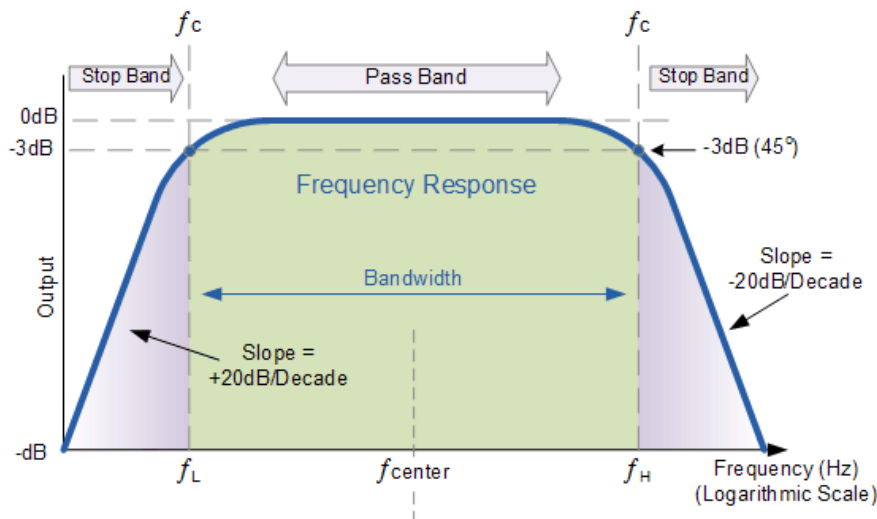


Figure 3: Band Pass Frequency Response

An ideal electrical filter should not only completely reject the unwanted frequencies but should also have uniform sensitivity for the wanted frequencies. Using a Band pass filter to reduce the frequency band used, reducing the numbers of channels (channel selection) and reducing the number of features, have a direct impact on reducing the execution time, and increasing the utilization of memory which enhance the system performance which is shown in figure 4 and 5 for hand and leg dataset.

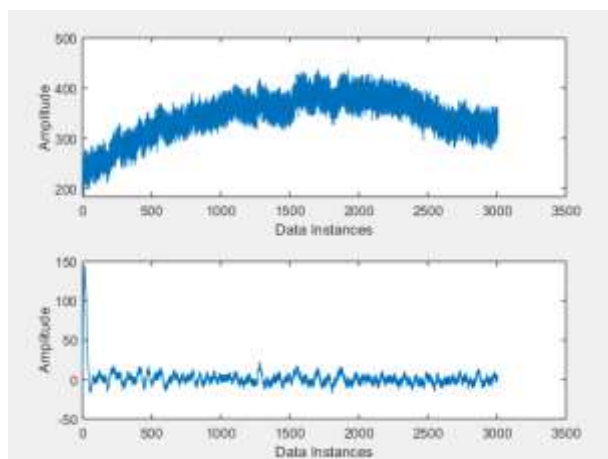


Figure 4: Original & Filtered Signal of Left Hand Movement

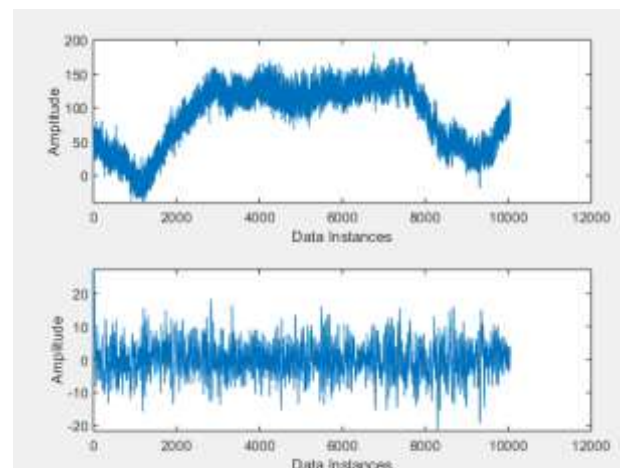


Figure 5: Original & Filtered Signal of Left Leg Movement

3.3 Feature Extraction

Once noise has been removed DWT features are extracted from the frequency band of the EEG dataset. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called features extraction. Thus the extraction of discriminatory features in the signal enhances the reduction of the length of the data vector by eliminating redundancy in the signal and compressing the relevant information into a feature vector of significantly lower dimension.

Discrete wavelet transform is used to extract characteristics from a signal on various scales proceeding by successive high pass and low pass filtering. The wavelet coefficients are the successive continuation of the approximation and detail coefficients The basic feature extraction procedure consists of:

1. Decomposing the signal using DWT into N levels using filtering and decimation to obtain the approximation and detailed coefficients.
2. Extracting the features from the DWT coefficients

Out of these DWT some useful information are extraction that reduces the dimensionality of the DWT features. Ten different useful information or features like mean, median, variance, standard deviation, skewness, kurtosis, complexity and mobility are generated and explained below:

i. Mean

Mean relates to the focus/center of a set of value. The Mean is considered for each and every sub-band signals[14]. It is computed as in equation (3.1).

$$Mean = 1/N \sum_{i=1}^n X_i \quad (3.1)$$

Here x_i is the i th data value and N is total number of instances.

ii. Median

The median [14] is the value splitting the upper half of a data sample/a population, or a probability distribution, from the lower half. In modest relations, it may be considered as the "intermediate" value of a data set.

iii. Variance

The average of the squared differences from the Mean is called variance. It is computed as in equation (3.2).

$$\sigma = \sum_{i=1}^n (X - \mu)^2 / N \quad (3.2)$$

Here x is the data value and N is total number of instances and μ is mean.

iv. Standard deviation

Standard deviation [15] is a simple measure of the changeability of a data set. The Standard deviation is the root-mean-square (RMS) deviation of its values from the mean computed as in equation (3.3).

$$std = \sqrt{\left(\sum_{i=1}^n (X_i - X)^2\right) / N - 1} \quad (3.3)$$

Here x is the data value and N is total number of instances.

vi. skewness

Skewness is a degree of the asymmetry of the probability distribution of a real-valued random variable about its mean[15,16]. The skewness value can be either positive or negative, or even undefined. This is given by equation (3.4).

$$skewness = E[(X - \mu\sigma)^3] \quad (3.4)$$

Here x is the data value, μ is mean and σ is standard deviation.

vii. Kurtosis

Kurtosis characterizes the comparative peakedness or flatness of a distribution associated with the normal distribution [15]. This is given by equation (3.5).

$$kurtosis = \mu_4 / \sigma^4 \quad (3.5)$$

where, μ_4 is the fourth moment about the mean and σ is the standard deviation.

ix. Complexity

The Complexity parameter signifies the variation in frequency[15]. This is given by equation (3.6).

$$Complexity = Mobility\left(\frac{dy(t)}{dt}\right) / Mobility(y(t)) \quad (3.6)$$

Here y is the frequency.

x. Mobility

The mobility parameter signifies the mean frequency or the proportion of standard deviation of the power spectrum [15,16]. This is given by equation (3.7).

$$Mobility = \sqrt{\text{var}\left(\frac{dy(t)}{dt}\right) / \text{var}(y(t))} \quad (3.7)$$

Here y is the frequency.

Sample of all extracted features are shown in table 1.

Table 1: Extracted Features Sample from EEG Signal for Hand and Leg Movements

Mean	Median	Variance	Standard Deviation	Skewness	Kurtosis	Mobility	Complexity	Class
-5035.96	0.017568	3.98E+08	19938.34	-2.01694	7.814556	1.41277	1.060004	Left Hand
4734.076	1.192113	3.9E+08	19756.77	1.947809	7.70422	1.40496	1.068737	Left Hand
-4320.43	0.891534	3.82E+08	19555.12	-1.9664	7.73024	1.43323	1.027758	Right Hand
4643.96	2.296905	3.77E+08	19404.16	2.016666	8.003585	1.42010	1.046357	Left Leg
-3952.4	-1.20394	3.74E+08	19334.16	-1.94751	7.810168	1.44956	1.00742	Left Hand
4220.06	2.34258	3.76E+08	19382.89	1.96256	7.945793	1.44202	1.02657	Left Hand
-4358.65	-3.87483	4.02E+08	20045.35	-1.89134	7.693731	1.41701	1.055424	Right Leg
3886.815	1.948793	3.75E+08	19357.06	1.896447	7.601353	1.43697	1.03031	Right Leg

3.4 Classification

Once the features are obtained these features are combined into vector to use. The available data set is first splitted into training and testing dataset respectively. Once the data is splitted classifier is first applied on training data for each ratio to generate rules of classifier. Then the test dataset was used to classify the data. For classification stacked deep auto encoder is used.

Stacked Deep Autoencoder

A stacked deep autoencoder is constructed by combining a stacked autoencoder, which comprises a desired number of cascaded autoencoder layers with a softmax classifier. The basic architecture of an autoencoder is a move forward with an input layer, often one hidden layers and an output layer. An autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs.

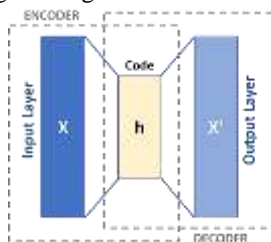


Figure 6: Autoencoder Architecture

Internally, it has a hidden layer 'h' that describes a code used to represent the input. The network may be viewed as consisting of two parts: an encoder function, $h=f(x)$ and a decoder that produces a reconstruction, $r=g(h)$.

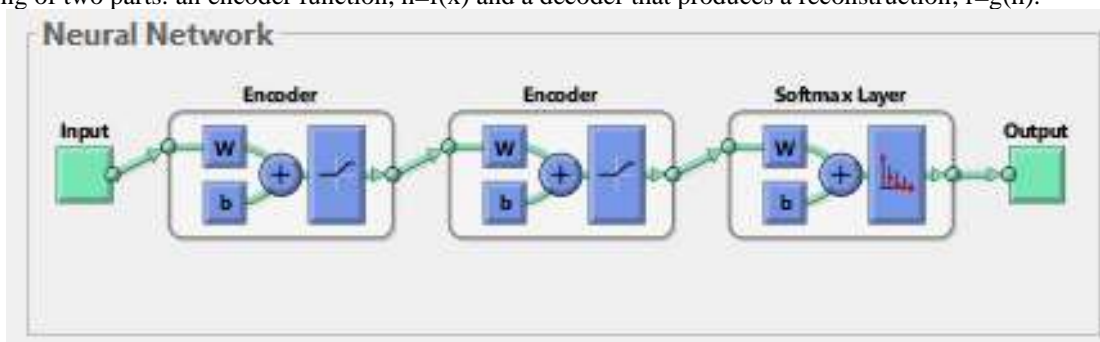


Figure 7: Stacked Deep Auto Encoder Architecture

The layer of deep autoencoder contains input layer, hidden layer and output layer.

The input layer is for passing the input values

The value at hidden layer is calculated as:

$$\hat{X} = \sum W * X_i + B_i \tag{i}$$

Where, W= weight matrix

X= input data values coming from input layers

B=Bias Matrix

And transfer function is calculated as :

$$f(x) = \frac{1}{1 + e^{-y}} \tag{ii}$$

Where, e=error value

The stacked deep autoencoder neural network involves multiple layer of autoencoders neural network and the loss function that is to be minimized as :

$$loss_{min} = |X - (W_1\theta(W_2\theta \dots \dots (W_l(f(x)))))| \quad (iii)$$

Where, W_1, W_2, \dots, W_l = weight function of all autoencoders

θ = Decoding function of autoencoders

$f(x)$ = function to calculate data values at each layer

Softmax classifier

A frequent use of the softmax function appears in the domain of machine learning, it associate with each output possibility a score, which is converted into probability with the softmax function. When a classification task has more than two classes, it is standard to use a softmax output layer. It is the latest layer and it offers a way to predict a discrete probability distribution over the classes.

IV. RESULT ANALYSIS

A. Performance Parameters

i. Accuracy

This measure signifies the recognition accuracy in percentage for each known test input to the total trained data and is given by [10]:

$$Accuracy = (TP + TN) / TP + TN + FP + FN$$

Where, TP = True positive

TN = True negative

FP = False positive

FN = False negative

The result analysis is performed by analyzing the performance of different classifiers, i.e. SVM, KNN, RF, Neural Network, Naïve Bayes and Stacked Deepauto Encoder. For evaluating the performance of these classifiers, the EEG hand and leg movement dataset is first of all cleaned. For cleaning purpose Butterworth filter is used to remove unnecessary noise from the dataset. Further features are extracted from the cleaned dataset and finally dataset is divided into two groups i.e. training set as well as testing set. For more efficient result analysis training and testing ratio are divided into 60:40 ratio, 70:30 ratio and 80:20 ratio respectively.

Table 2: Performance Evaluation for 60:40 Training and Testing Ratio

TESTING SAMPLE I			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.84	74.96	52.93
k-Nearest Neighbour	82.96	77.093	75.13
Random Forest	81.90	63.80	63.80
Neural Network	76.42	39.88	28.57
Naïve Bayes	63.78	57.67	93.45
Stacked DeepAuto Encoder	84.67	80.710	99.980
TESTING SAMPLE II			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.71	74.52	53.10
k-Nearest Neighbour	82.63	77.19	75.36
Random Forest	81.71	63.43	63.43
Neural Network	76.47	42.52	24.96
Naïve Bayes	63.80	57.75	93.10
Stacked DeepAuto Encoder	86.47	82.81	100

TESTING SAMPLE III			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.02	73.52	51.54
k-Nearest Neighbour	83.05	77.21	75.90
Random Forest	81.91	63.84	63.84
Neural Network	76.44	40.82	25.09
Naïve Bayes	63.53	57.59	92.91
Stacked DeepAuto Encoder	87.38	83.87	99.96
TESTING SAMPLE IV			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.66	74.14	53.15
k-Nearest Neighbour	83.02	77.33	75.72
Random Forest	81.35	62.71	62.71
Neural Network	77.62	49.62	48.47
Naïve Bayes	63.77	57.52	93.29
Stacked DeepAuto Encoder	86.34	82.73	100
TESTING SAMPLE V			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.48	74.55	51.95
k-Nearest Neighbour	82.86	77.00	75.27
Random Forest	81.61	63.24	63.24
Neural Network	76.94	44.45	26.15
Naïve Bayes	63.49	57.79	93.18
Stacked DeepAuto Encoder	86.11	82.43	99.89

Table 2 illustrates the accuracy analysis of different classifiers for 60:40 ratio. From the result, it has been seen that stacked deepauto encoder gives more accurate result for all 4 classes of dataset.

Table 5: Average Performance Evaluation for 60:40 Training and Testing Ratio

Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.542	74.338	52.534
k-Nearest Neighbour	82.904	77.1646	75.476
Random Forest	81.696	63.404	63.404
Neural Network	76.778	43.458	30.648
Naïve Bayes	63.674	57.664	93.186
Stacked DeepAuto Encoder	86.194	82.51	99.966

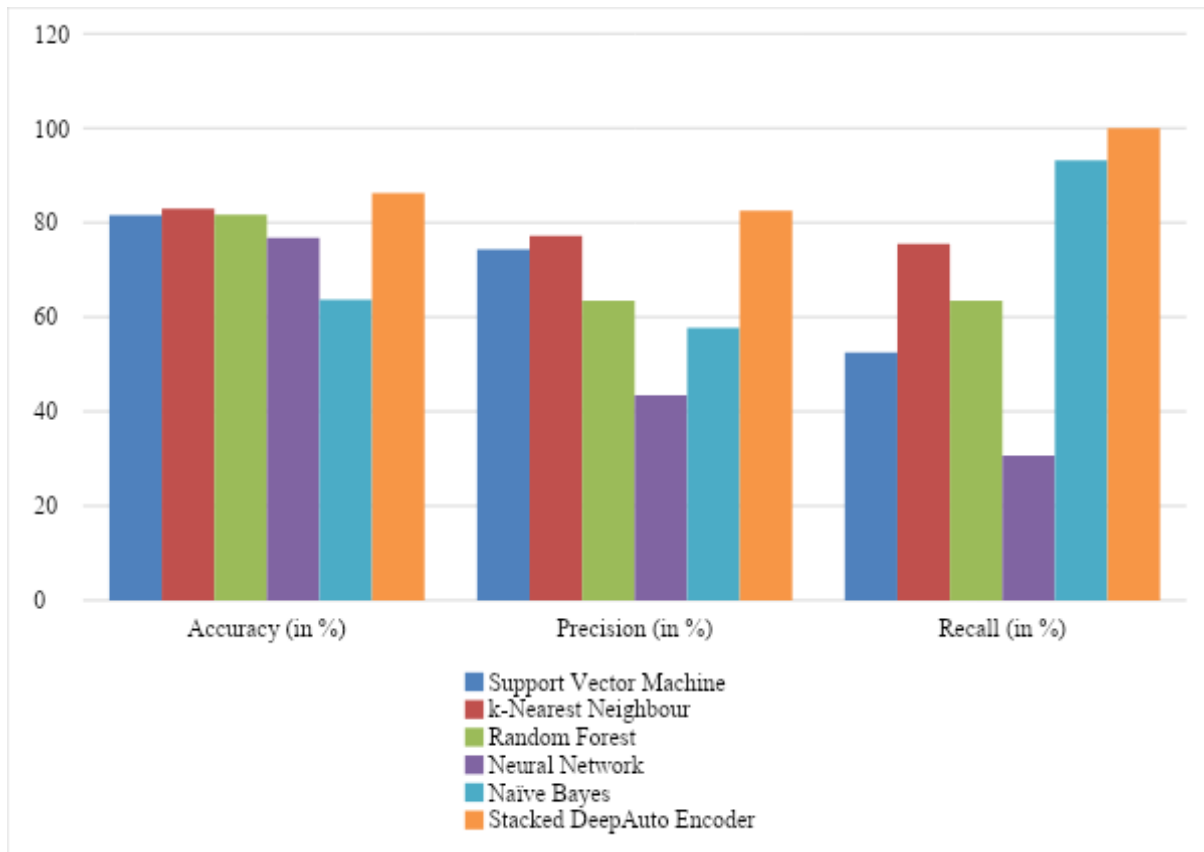


Figure 8: Average Performance Evaluation for 60:40 Training and Testing Ratio

Figure 8 illustrates the graph of performance evaluation of proposed classifier against some existing classifiers and it has been seen that stacked deep auto encoder gives better performance. The result analysis was performed on five different testing data samples and their average value is considered as an output for the 60:40 ratio dataset. From the result it is seen that stacked deep auto encoder have achieved approx. 87% accuracy and approx. 83% precision which is highest among all other classifiers.

Similarly, table 4 illustrates the accuracy analysis of different classifiers for 70:30 ratio. From the result, it has been seen that stacked deepauto encoder gives more accurate result for all 4 classes of dataset and performance also increases with increase in training ratio. So, it is also concluded that with increase in training ratio accuracy of the classifier also increases.

Table 4: Performance Evaluation for 70:30 Training and Testing Ratio

TESTING SAMPLE I			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.2961	75.83	54.45
k-Nearest Neighbour	83.90	79.08	72.13
Random Forest	81.41	62.83	62.83
Neural Network	77	41.36	23.36
Naïve Bayes	64.37	59.16	93.89
Stacked DeepAuto Encoder	90.36	87.35	99.97
TESTING SAMPLE II			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.86	75.47	53.91
k-Nearest Neighbour	84.03	79.27	72.35

Random Forest	82.15	64.31	64.31
Neural Network	76.97	42.79	21.71
Naïve Bayes	65.09	59.61	94.33
Stacked DeepAuto Encoder	90.42	87.43	99.98
TESTING SAMPLE III			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.81	75.43	52.82
k-Nearest Neighbour	83.80	79.39	72.30
Random Forest	81.45	62.91	62.91
Neural Network	78.74	59.46	49.71
Naïve Bayes	64.59	59.19	94.20
Stacked DeepAuto Encoder	90.66	87.74	99.92
TESTING SAMPLE IV			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.26	76.61	53.62
k-Nearest Neighbour	84.26	79.96	72.57
Random Forest	81.93	63.86	63.86
Neural Network	77.33	41.82	22.38
Naïve Bayes	64.63	59.19	94.01
Stacked DeepAuto Encoder	90.27	87.32	99.88
TESTING SAMPLE V			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.80	75.16	52.77
k-Nearest Neighbour	83.86	79.55	71.93
Random Forest	81.80	63.61	63.61
Neural Network	77.27	44.57	22.21
Naïve Bayes	64.97	59.10	94.35
Stacked DeepAuto Encoder	90.22	87.14	100

Table 5: Average Performance Evaluation for 70:30 Training and Testing Ratio

Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.00522	75.7	53.514
k-Nearest Neighbour	83.97	79.45	72.256
Random Forest	81.748	63.504	63.504
Neural Network	77.462	46	27.874
Naïve Bayes	64.73	59.25	94.156
Stacked DeepAuto Encoder	90.386	87.396	99.95

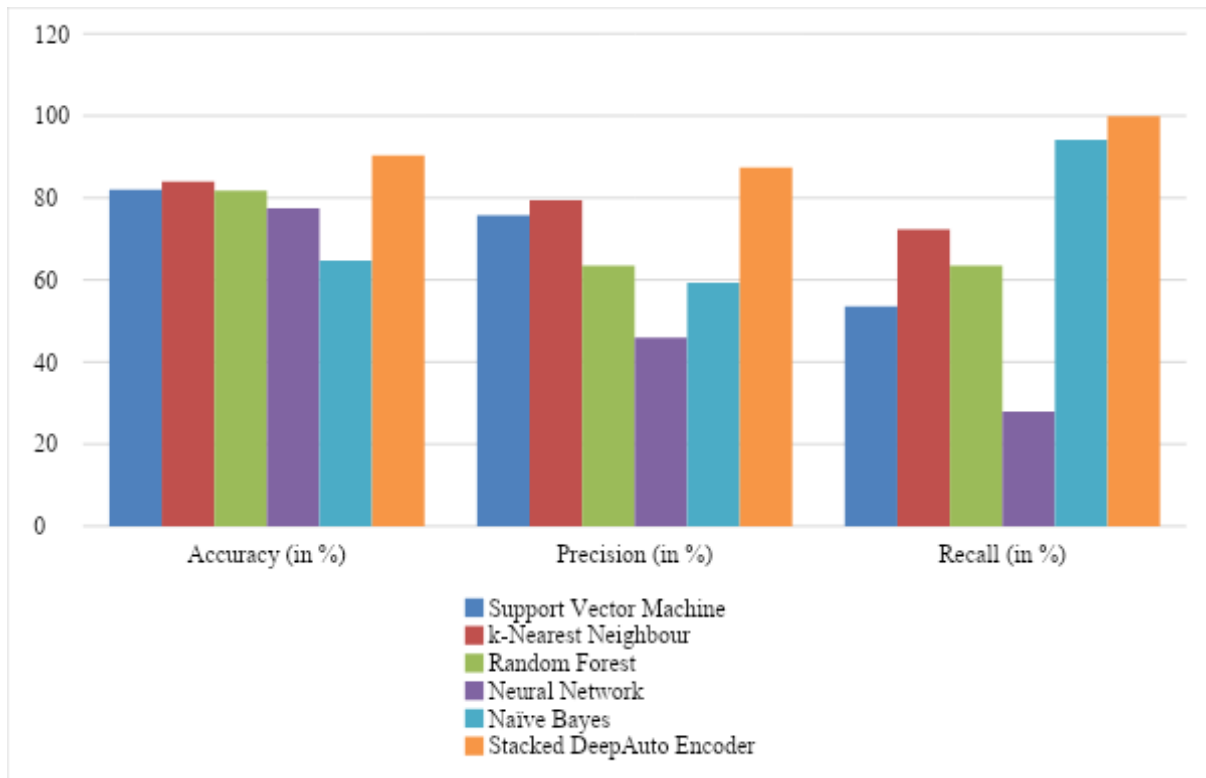


Figure 9: Average Performance Evaluation for 70:30 Training and Testing Ratio

Figure 9 illustrates the graph of performance evaluation of proposed classifier against some existing classifiers and it has been seen that stacked deep auto encoder gives better performance. The result analysis was performed on five different testing data samples and their average value is considered as an output for the 70:30 ratio dataset. From the result it is seen that stacked deep auto encoder have achieved approx. 90% accuracy and approx. 87% precision which is highest among all other classifiers.

Table 6: Performance Evaluation for 80:20 Training and Testing Ratio

TESTING SAMPLE I			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.39	76.23	54.28
k-Nearest Neighbour	83.89	79.08	72.13
Random Forest	81.40	62.81	62.81
Neural Network	77.58	41.63	29.02
Naïve Bayes	64.24	59	93.71
Stacked DeepAuto Encoder	90.29	87.34	99.88
TESTING SAMPLE II			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.69	75.01	52.70
k-Nearest Neighbour	83.97	79.55	72.34
Random Forest	81.27	62.55	62.55
Neural Network	78.26	58.44	43.70
Naïve Bayes	64.23	58.84	93.63
Stacked DeepAuto Encoder	90.34	87.29	100

TESTING SAMPLE III			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.03	75.53	53.45
k-Nearest Neighbour	83.83	79.42	72.45
Random Forest	81.89	63.80	63.80
Neural Network	77.25	42.78	25.33
Naïve Bayes	64.66	59.17	94.40
Stacked DeepAuto Encoder	90.81	87.88	99.98
TESTING SAMPLE IV			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.25	75.08	53.64
k-Nearest Neighbour	84.37	79.80	72.85
Random Forest	82.02	64.04	64.04
Neural Network	79.05	60.89	45.60
Naïve Bayes	64.56	58.84	94.46
Stacked DeepAuto Encoder	90.11	87.02	100
TESTING SAMPLE V			
Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	81.92	74.98	53.69
k-Nearest Neighbour	84.21	79.86	72.73
Random Forest	81.81	63.63	63.63
Neural Network	76.91	39.04	22.50
Naïve Bayes	64.48	59.13	94.46
Stacked DeepAuto Encoder	90.68	87.73	100

Table 6 illustrates the accuracy analysis of different classifiers for 80:20 ratio. From the result, it has been seen that stacked deepauto encoder gives more accurate result for all 4 classes of dataset and performance also increases with increase in training ratio. So, it is also concluded that with increase in training ratio accuracy of the classifier also increases.

Table 7: Average Performance Evaluation for 80:20 Training and Testing Ratio

Techniques	Accuracy (in %)	Precision (in %)	Recall (in %)
Support Vector Machine	82.056	75.366	53.552
k-Nearest Neighbour	84.054	79.542	72.5
Random Forest	81.678	63.366	63.366
Neural Network	77.81	48.556	33.23
Naïve Bayes	64.434	58.996	94.132
Stacked DeepAuto Encoder	90.446	87.452	99.972

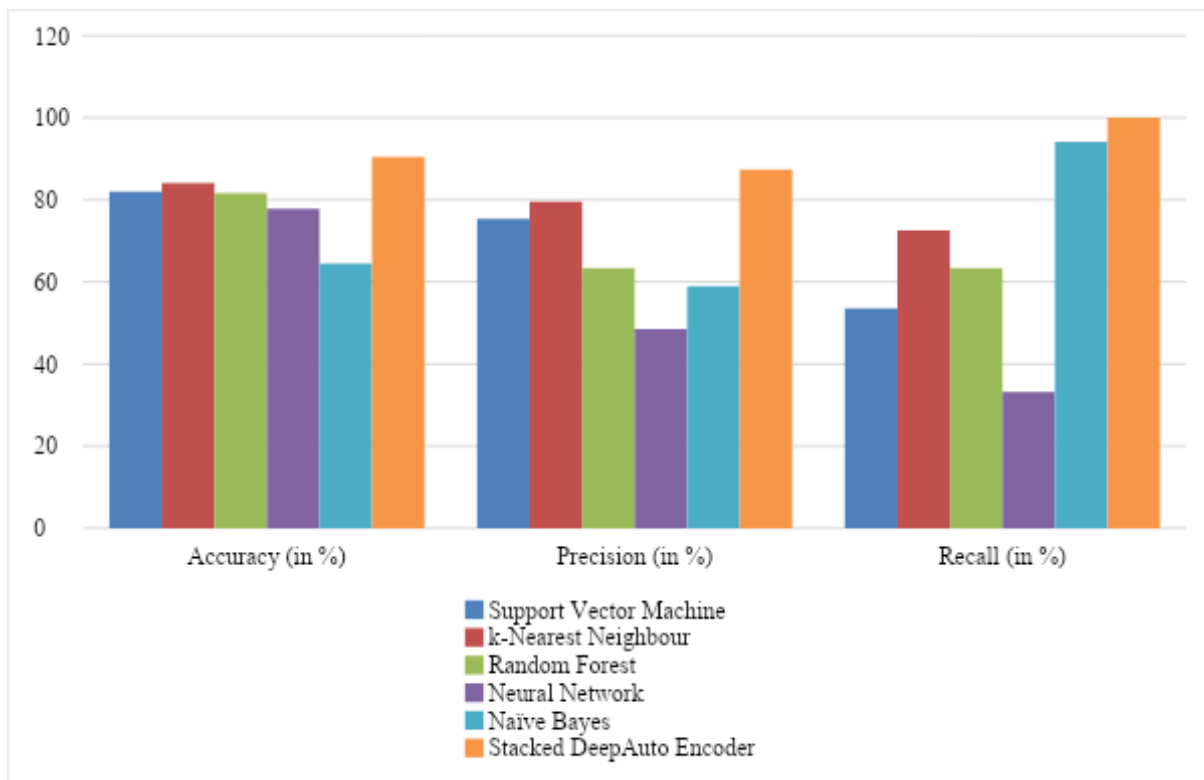


Figure 10: Average Performance Evaluation for 80:20 Training and Testing Ratio

Figure 10 illustrates the graph of performance evaluation of proposed classifier against some existing classifiers and it has been seen that stacked deep auto encoder gives better performance. The result analysis was performed on five different testing data samples and their average value is considered as an output for the 80:20 ratio dataset. From the result it is seen that stacked deep auto encoder have achieved approx. 91% accuracy and approx. 87% precision which is highest among all other classifiers.

V. CONCLUSION

BCI is the gift for people with disabilities, especially for those who cannot use the normal way out and muscle movements of the brain. BCI techniques vary depending on the application and require different methods to recognize the characteristics of pre-processed EEG signals and monitoring devices. This research presented the current assessment and trends in BCI. Non-invasive methods such as EMG, fMRI and NIRS are more popular and easier to use because no surgical implant is required.

Non-invasive BCI records the brain signals with integrally embedded noise which may include electromyography, signal evoked by muscular activity, eye blinking etc. These unwanted components can be filtered out with suitable filters. In this research work the proposed algorithm is designed for BCI applications where left hand forward-backward movements and right hand forward-backward movements as well as left leg movement and right leg movement are classified. Features are extracted from these datasets for classification of type of movements SVM, KNN, RF, Neural Network, Naïve Bayes and Stacked Deepauto Encoder techniques are used to classify and evaluate the performance measures. After result analysis it is concluded that stacked Deepauto encoder outperforms better as compared to all other classifiers in terms of accuracy.

BCI is a most innovative application that are being developed day-by day with an application of artificial intelligence. Additional research is also required for BCI development. These include explorations of: useful brain signals; signal recording techniques; feature extraction and translation strategies; methods for engaging short- and long-term variations between user and system therefore as to optimize performance; acceptable BCI applications; and clinical validation, dissemination, and support. So in future work, in all above areas it is needed to be used more effectively in real-life environments.

REFERENCES

- [1] E. Ianez, M. C. Furio, J.M. Azorn, J.A. Huizzi, and E. Fernandez, "Bmental tasks based brain robot interface, robotics and autonomous systems," Elsevier, 2010, pp. 1238-1245.
- [2] U.Gl, Y. Gltrk, C. Loo, "Evaluation of fractal dimension estimation methods for feature extraction in motor imagery based brain computer interface," Elsevier, 2011, vol. 3, pp. 589-594.
- [3] H. Sun, Y. Xiang, H. Zhu and J. Zeng, "On-line EEG classification for Brain Computer Interface based on CSP and SVM", *IEEE*, 2010.
- [4] Howida A. Shedeed, Mohamed F. Issa, Salah M. El-sayed, "Brain EEG Signal Processing For Controlling a Robotic Arm", *IEEE*, 2013, pp. 152-157.
- [5] Mohd Shuhanaz Zonar Azalan, Paulraj M P, Sazali bin Yaacob, "Classification of Hand Movement Imagery Tasks for Brain Machine Interface Using Feed-Forward Network", *IEEE*, 2014, pp. 431-436.

- [6] Muhammed. S. P. P. • Waseem Raza. and David Martin Wart Powers. "Brain computer Interface Classification of eeg for left and right wrist movements using AR modeling & Bhattacharyya distance", *Sensor Networks and Information Processing*, 2011, pp.7-10.
- [7] Bhattacharyya, Saugat, Anwesha Khasnobish, Amit Konar, D. N. Tibarewala, and Atulya K. Nagar. "Performance analysis of left right hand movement classification from EEG signal by intelligent algorithms." *Cognitive Algorithms, Mind, and Brain (CCMB), IEEE*, 2011 pp. 1-8.
- [8] Hema, Chengalvarayan Radhakrishnamurthy, M. P. Paulraj, S. Yaacob, A. H. Adorn & R.Nagarajan. "Recognition of motor imagery of hand movements for BMI using PCA features." *Electronic Design (ICED), IEEE*, 2008, pp. 1 -4.
- [9] Marquez L, Alejandro, & Munoz. "Analysis & classification of electroencephalographic signals (EEG) to identify arm movements.", *Electrical Engineering, Computing Science and Automatic Control (CCE), IEEE*, 2013, pp. 138- 143.
- [10] Zhiwei, Li, and Shen Minfen. "Classification of mental task EEG signals using wavelet packet entropy and SYM.", *IEEE*, 2007, pp.906-909.
- [11] Ting, Wu, Yan Guo-zheng, Yang Banghua, & Sun Hong. "EEG feature extraction based on wavelet packet decomposition for brain computer interface.", *Elsevier*, 2008, vol. 41, issue 6, pp. 618-625.
- [12] Syed Khairul Bashar and Mohammed Imamul Hassan Bhuiyan, "Identification of Arm Movements Using Statistical Features from EEG Signals in Wavelet Packet Domain ", *IEEE*, 2015.
- [13] Saugat Bhattacharyya, Debabrota Basu, Amit Konara, D.N. Tibarewala, "Interval type-2 fuzzy logic based multiclass ANFIS algorithm for real-time EEG based movement control of a robot arm", *Elsevier*, 2015.
- [14] Faisal Farooq and Preben Kidmose, "Random Forest classification for p300 based brain computer interface applications," *IEEE*, 2013, pp 1-5.
- [15] Prasant Kumar Pattnaik , Jay Sarraf, " Brain Computer Interface issues on hand movement", *Journal of King Saud University – Computer and Information Sciences* (2018) 30, 18–24.
- [16] Muhammed Al-Suify, Walid Al-Atabany, Mohamed.A.A.Eldosoky, "Classification of right and left hand movement using nonlinear analysis" *IEEE*, 2017.
- [17] <https://sites.google.com/site/projectbci/>