International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 8 (2017) pp. 1736-1741

Implementing and analyzing different Machine Learning Algorithms using EEG based BCI

Anupama.H.S[1], Raj V Jain[2], Revannur Venkatesh[3], N.K.Cauvery[4] and Lingaraju.G.M[5]

[1, 2, 3] Department of Computer Sc. and Engineering, R. V. College of Engg., Bangalore, India.
[4]Department of Information Sc. and Engineering, R. V. College of Engg., Bangalore, India.
[5] Department of Information Sc., M. S. Ramaiah Institute of Tech., Bangalore, India.

Abstract

Communication that exists between Human Brain and Computer is termed as Brain Computer Interface (BCI). This uses neuronal activity of the brain. Information is passed from one part of the body to another through the neurons present in the human brain. This paper focuses on different types of classification algorithm to explore a new kind of BCI paradigm. This also focuses on the Information Transfer Rate of the existing paradigms and also compares our paradigm with the existing paradigms.

Keywords: Brain Computer Interface, like K nearest Neighbors (KNN), Random Forest (RF), Hidden Markov Models (HMM), HMM with Gaussian Mixture Model (HMM with GMM), Discriminant Analysis(DA), Naïve Bayes (NB), Support Vector Machine (SVM) and Decision Tree (DT).

INTRODUCTION

Machine learning (ML) is the field which adds intelligence to devices providing them with capabilities to process and identify patterns in data just like human beings do. Programming devices in this manner can help in identifying those patterns which human beings often cannot. Machine learning is based on modeling data mathematically. ML has been gaining a lot of attention in the last few decades, especially in fields of interdisciplinary research. Brain Computer Interface (BCI) is one such area where applying machine learning has become a necessity. One of the ways in which the performance of a BCI system is measured is using Information Transfer Rate. All the current research is aimed at proposing systems with higher ITR. The main aim of the proposed system is to achieve higher ITR by combining two different approaches, namely Steady State Visually Evoked Potential and Steady State Auditory Evoked Potential. The system developed aims at checking if such a system can exist and if so, if it is providing accuracy that is high enough to be put to use in real world applications [1].

The system proposed aims to enhance the decision making capability of the person by providing two stimuli, which evoke two responses in different regions of the brain. This allows the user to make 2 decisions simultaneously. Since such a system has not been developed before, it should first be validated if the system provides results with sufficient accuracy to be able to be put to use for end users or patients.

Hence, a validation system has been developed which employs several feature extraction and classification algorithms. A comparison of different algorithms for this system has also been provided. The result of the experiment whether such a system can provide accurate results or not, has also been discussed.

Many of the existing BCI paradigms have very good accuracy measures but problematic ITRs, which makes it difficult to use them in real life applications like communication mechanism for the physically challenged, gaming etc. The motivation behind this project is to explore a new kind of BCI paradigm and validate whether it can give a better ITR as compared to the existing paradigms. Also, see the feasibility of making such a BCI paradigm real time. The inherent challenge that exists is that the accuracy should not be compromised. Any EEG based BCI system suffers from a lot of noise. Thus, machine learning is used so that the effects of this noise can be offset and higher accuracy can be gained.

The main aim of this work is to use this model in various hospitals and nursing centres for the paralysed where currently only SSVEP system is used or any other system is used to understand the needs of the physically challenged. Faster decision making means faster services for the physically challenged. With minor changes, it can be used in gaming applications and in building sentences and so on. Further, this model can be used as a basic model and extensive optimization approaches can be applied to it for better accuracy in near future [5, 6].

METHODOLOGY

The methods Steady State Visually Evoked Potential (SSVEP) and Steady State Auditory Evoked Potential (SSAEP) are two paradigms used widely till now to elicit decision from the user by using his/her attentions visually or auditory respectively. In SSVEP paradigm two images flickering at different frequencies which are 7.5 Hz and 10Hz respectively. The frequencies are selected such that they are not harmonics of each other and well within the range of 6-24 Hz. SSAEP signals are generated using auditory input having two pure tones of different frequencies (amplitude modulated). The frequencies here are selected such that they are around 40 Hz so as to maximize the SNR. The characteristics of the frequencies define an SSAEP system. For example, in this system it has been decided to give 37 Hz and 43 Hz frequency audio input. Both the visual and auditory inputs are given concurrently to the subject [7].

Each of choices in the visual and auditory stimuli stand for a particular value of a particular decision. For example, the visual stimuli can represent food selection and the auditory stimuli can represent drink selection. After making the subject understand the meaning of each of the choices, the training trials are started. Each training trial will begin by a rest period of approximately 2 sec. Here the subject is asked to concentrate on one of the visual and auditory inputs as determined by the training program. This gives the subject to form his decision and also helps to get the baseline EEG data. Then, the visual and auditory inputs are given and EEG data is taken and used to get features.

APPROACHES TO MACHINE LEARNING ALGORITHMS

Some machine learning algorithms require non-time series data, meaning the data should be represented by a single value over the whole time period for which the data was collected. Machine leaning algorithms which use non-time series data are Support Vector Machines (SVM), K-Nearest Neighbour (KNN), etc. Some machine learning algorithms require time series data, meaning the data should represent some quantity varying with respect to time. Machine leaning algorithms which use time series data are Hidden Markov Models (HMM), Recurrent Neural Networks (RNN), etc.

➢ *Decision Tree*

A decision tree is a graph like model in which every node represents a logical decision to be made, which finally results in nodes that contain only those observations which belong to a particular class [20]. Every decision tree has zero or more internal nodes and 1 or more leaf nodes. The tree is first constructed from the training data set. This process is called decision tree induction. Most of the decision tree algorithms build a tree in the top down manner. The basic steps used to construct a decision tree are given below:

- If all the training feature vectors have the same label l at the node n, then a leaf node of class n is formed.
- Find all possible splits at the next node.
- A goodness measure is used to evaluate each of these splits.
- The best split is determined based on the score using the goodness measure.
- The node has as many child nodes as the number of outcomes of the split.
- Repeat this process for each node in the tree until no more splits are possible, i.e., all the nodes have training observations that belong only one particular class.
- Classification using decision trees start at the root node. The observation passes through each node. The next node that it goes to is decided based on the split at that node. The observation finally reaches one of the pure or leaf nodes and hence its class is determined.

➢ *Naïve Bayes Classifier*

This is a classification technique based on the Bayes theorem [21]. The main assumption of this classifier is that the features in a particular class are all independent of each other (although this is untrue in most of the cases).
Naïve Bayes classification works as follows: Assign a feature vector x to class c if the posterior probability $p(y = c|x)$ is greater than the posterior probability $p(y = c'|x)$ for any other class $c' \neq c$.

The calculation of posterior probability $p(y|x)$ is difficult. Hence, Bayes theorem is used instead as it is easier to calculate prior probabilities and the likelihood. Bayes theorem is as given below

$$p(c|x) = \frac{p(x,c)}{p(x)} = \frac{p(x|c)*p(c)}{\Sigma p(x|c')*p(c')}$$

Where, $p(c|x)$ is called the posterior probability. $p(x|c)$ is called the likelihood, $p(c)$ is called the class prior probability.

➢ *Random Forest*

As the name indicates, random forest uses a set of trees for classification [22]. These random forests are created and grown as follows:

- Let the number of training feature vectors be n. Let s be the number of trees in the forest. Then each tree will be grown using a dataset of size$n$. But the dataset contains replacements or training features which are also considered in the dataset of other trees. The number of replacements in the tree should not be more that 33% of the total observations.
- Each training feature vector contains m attributes or features. Each tree considers a subset of these m features, say $\sqrt{m}$ features to create the tree (or the splits). The number of features used remains fixed.

Trees so created are grown without pruning. The s trees then form the random forest. To classify new data, the data is passed through every tree, just like a decision tree. Each tree then classifies the input data to a particular class. The class with the highest number of votes from the trees is taken as the class to which the new data value belongs to.

➢ *Hidden Markov Model (HMM)*

Hidden Markov Model is stochastic model which incorporates the time series data by following the Markovian property [17]. It is observed to be a finite state machine which has the value of the current state depending on certain fixed number of previous states.
For this model parameters are $\lambda = \{\pi_i, a_{ij}, b_{ij}\}$, where, π_i is the probability that state i is the initial state, a_{ij} is the probability that transition from hidden state i to hidden state j would occur and b_{ij} is the probability that from a hidden state i there outputs the visible symbol j. The initial state probabilities help the model to choose the optimal path to predict the most probable sequence. a is called transition matrix and b is called the emission matrix.
Equation for HMM is as follows:

$$P(x_{1,\ldots,}x_{T,}y_{1,\ldots,}y_T) = P(x_1)P(y_1|x_1)\prod_{t=2}^{T} P(x_t|x_{t-1})P(y_t|x_t)$$

The discrete HMM models require discrete symbols of hidden states [18]. For any classification procedure two phases, one training phase and a testing phase are seen. The training and testing data are cross validated before classification. In training phase HMM is provided with the training decision sequences. In testing phase cross validated part of decision sequence is given to predict the transition and emission probabilities. Upon these results log-likelihood is obtained to classify it to any of the classes to which it belongs.

➢ ***Hidden Markov Model with Gaussian Mixture Model (HMM with GMM)***

The spectral clustering is a process which calculates the optimal number of clusters required to improve the accuracy of the model applied with Gaussian mixtures. This provides the 'n' states required to be made for further calculation of log likelihood probabilities of such training or testing sequences. The mathematical view of the spectral clustering is as follows [19]: Consider n trials in R^P which have been obtained by cross validating the set of all trails. These n trial records are grouped into k clusters by calculating validity score, which is represented by α, where

$$\alpha = \sum_{c=1}^{k} \frac{1}{N_c} \sum_{i,j \in Z_c} Y_{ij}$$

And Z_c denotes cluster c, N_c stands for number of items in cluster c and Y_{ij} is the normalized eigenvector matrix.

DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) [55] is a graphical representation of the "flow" of data through an information system. Data flow models are used to show how data flows through a sequence of processing steps. The data is transformed at each step before moving on to the next stage. These processing steps or transformations are program functions when DFD are used to document a software design. The DFD for the Software can be decomposed into three levels such as level 0, level 1 and level 2.

The level 0 is the initial level Data Flow Diagram and it's generally called as the Context Level Diagram. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities as shown in figure 1.1. This context-level DFD is then "exploded" to show more detail of the system being modeled.

DFD0 describes the process of eliciting EEG responses from the subject under experimentation. The process accesses the data bases for storing and retrieving the EEG files, features and models for providing a decision as an output. The validation parameters are the keys through which the output is formed.

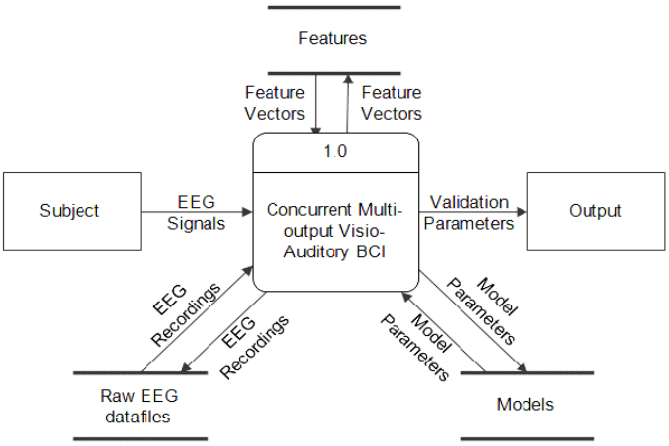

Figure 1.1: DFD Level 0

The Level 1 DFD in figure 1.2 clearly shows the bifurcation of CMOVA BCI Software into its constituent modules which performs set of tasks from the acquisition of data to delivering the output. The sequence showed in the figure elucidates how the elicited data from the subject is passed on pre-processing and to feature extraction methods and then to classification providing a result or a decision. After the data acquisition process, the generated files are stored into database, upon which pre-processing techniques are applied. The generated feature vectors are stored back into feature database. These features are made use of in classifying the data into one of the classes as a decision. The classification algorithms later on depend on the earlier generated models to test its efficiency as one's classifier strength. The number of classes involved is four, which are dependent on the combination of number of objects to be observed, which is two and number of audio to be listened, which is two.

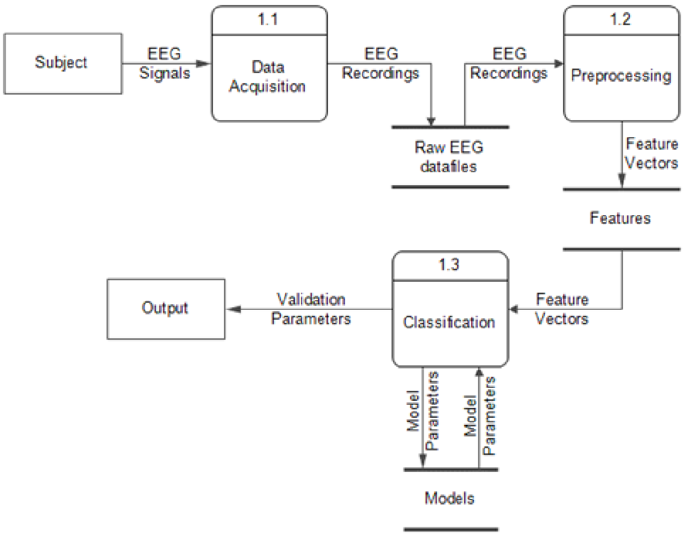

Figure 1.2: DFD Level 1

The Level 2 Data Flow Diagram gives the complete picture of all the sub modules in this project as shown in the Figure 1.3, 1.4, 1.5. Following figure shows the Level 2 DFD of the Data Acquisition module.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 8 (2017) pp. 1736-1741

The DFD with processes 1.1.1 and 1.1.2 show that in the acquisition of signals performed by BESS software has two phases. The first phase is to record the signals received from the electrodes placed on the scalp of the subject, the second phase is to convert files where the files received in .EEG format are converted into .EDF format. These .EDF files are stored in a database of raw EEG data files.

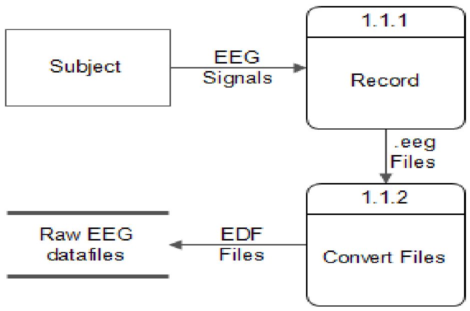

Figure 1.3: DFD Level 2a – Data Acquisition

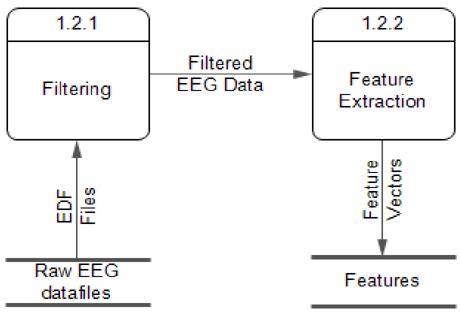

Figure 1.4: DFD Level 2b – Pre-processing

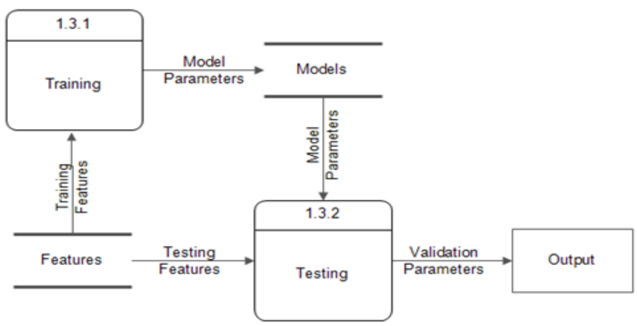

Figure 1.5: DFD Level 2c – Classification process

In the figure 1.5 the classification of the features into different categories by forming a decision is illustrated. In every classification process there are two methods to make a model. Firstly, training model which trains the EEG data given from the subject. In this case, data for training were the signals obtained from the different parts of the brain i.e. temporal and occipital regions signifying that responses were prompted for visual and auditory stimuli from this area. K-Fold cross validation was applied to this data from different subjects due to which for training the model, training features were taken from the features database and for testing the model, testing features were selected.

The Classification module informs the decision based on the model's prediction and accuracy after receiving feature's set. This module also informs the functionality and component attributes of the feature set. This module helps in finding best features extraction method with classification algorithm on SSVEP and SSAEP paradigms.

Purpose: To classify the EEG signals into a decision depending on whether the subject's attention towards left or right video/audio.

Functionality: The features set is passed through the following algorithms: SVM, DA, KNN, Naïve Bayes, HMM, HMM Gaussian mixture model.

Input: Features obtained from feature extraction technique.

Output: K fold cross validated testing results and accuracy of the model.

Flowchart: The models developed in the classification module are trained and trusted using K fold cross validation, employing all the trail data in the training dataset. Each example is labelled for Video and Audio are represented by four labels depending on the combination of respective attentions. This labeled data set is now used to define the Classification Model. The test set is then run over the Classification Model and accuracy score is obtained as shown in figure 1.6 and figure 1.7

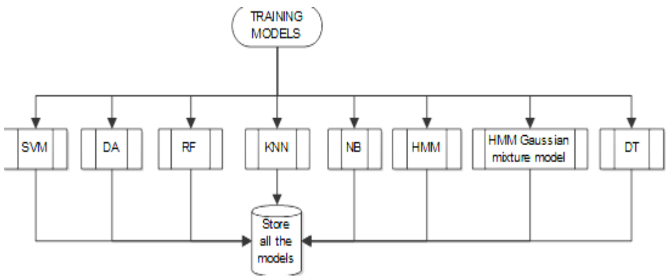

Figure 1.6: Training algorithms of Classification Module

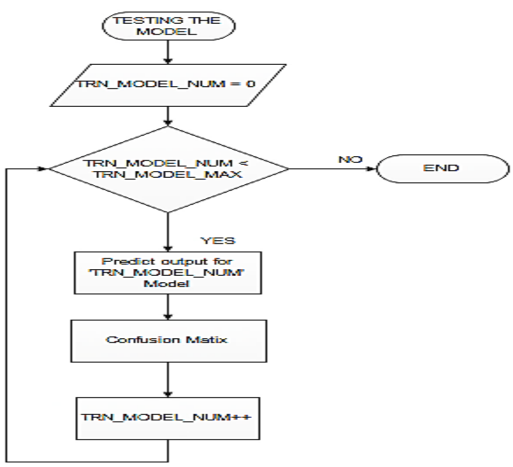

Figure 1.7: Testing Module

EXPERIMENTAL RESULTS AND ANALYSIS

➢ ***Experimental Dataset***

Any machine learning application is data driven. Hence the dataset considered for the application is of utmost importance. The details of dataset are as given below.

- ***Participants***

Data was collected from a total of 10 healthy subjects consisting of seven males and three females. All of them were aged between 18 and 23. Nobody had had any neurological disorder.

➢ ***Data acquisition***

The experiment was conducted in an isolated room with no electronic devices running except for the data acquisition system. The participants were also instructed not to wear any metallic items and to keep away all electronic gadgets, to minimize any interference with the acquisition. The acquisition system consisted of an EEG cap that could be adjusted according to the participant's head dimensions, 16 Ag-AgCl electrodes, an amplifier, a Bluetooth device to record signals, a display on which the visual stimulus was presented, earphones through which the audio stimulus was presented and a system with BESS software running on it, to collect the EEG data. The electrodes were placed according to the 10 – 20 system. The EEG data was sampled at the rate of 256Hz. Also a Notch filter of 45Hz is applied to the data. Time stamped data is then recorded using the BESS software. The BESS software saves data as a .EEG file. However the MATLAB readable format is .EDF file. Hence, this conversion has to be done manually.

EDFbrowser is software which can be used to view edf files. A part of the EEG data recorded for one of the subjects viewed using EDFbrowser is shown in figure 1.8. The annotations made during the recordings are displayed in the right side of the window. The figure shows the readings from all the sixteen channels that have been used for recording. However, only 6 of these channels are required for our experiment, namely, O1, O2 and Oz for the response to visual stimulus and T3, T4, Cz and Oz for response to auditory stimulus.

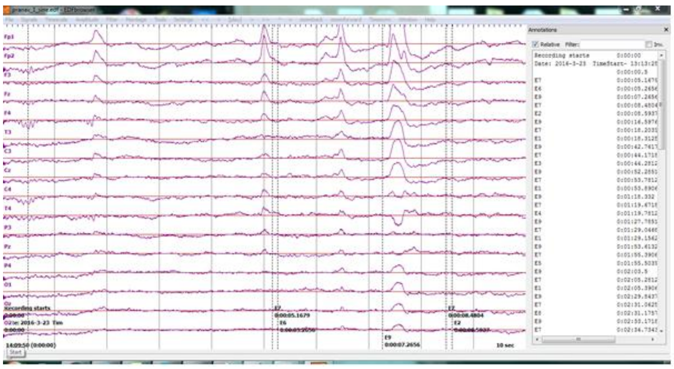

Figure 1.8: EEG recordings of one of the subjects

➢ ***Performance Analysis***

The output obtained from a system is evaluated against the different metrics to measure the performance of the system. Several machine learning algorithms have been used to validate the paradigm. Each graph represents a comparison of the performance of each feature extraction algorithm for a given machine learning algorithm. The accuracies plotted are based on 10-fold cross validation. The x axis values represent the combination of the parameters for each machine learning algorithm.

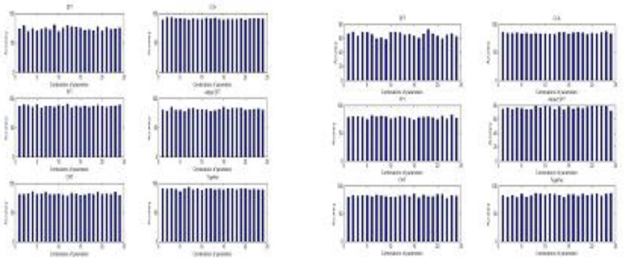

Figure 1.9: Graph of parameters Vs. accuracy for decision tree for SSVEP with sine and words stimulus

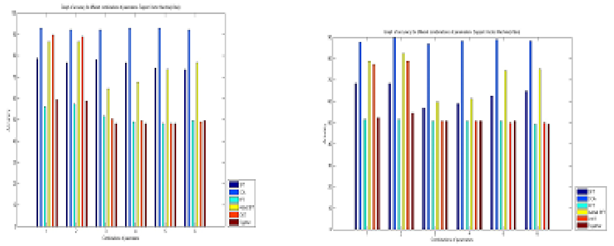

Figure 1.10: Graph of parameters Vs. accuracy for SVM for SSVEP with sine and words stimulus

CONCLUSION AND INFERENCE

This system was proposed to check if two BCI paradigms can be combined. The classification accuracies obtained above show that such combinations can be made. The best algorithm for this is GHMM which gives an average classification of 95%. Also with respect to the non-time series data, CCA is the best feature extraction algorithm for SSVEP. However, such a conclusion cannot be made for SSAEP. Another important aspect of the project was to improve the ITR.

ACKNOWLEDGEMENT

We would like to thank Axxonet solutions pvt limited for their immense support in getting the data for our work. They helped in providing the BESS software and the supporting Hardware for data extraction from the Human Brain.

REFERENCE

[1] A Vallabhaneni, et. al., "Brain-Computer Interface," in Neural Engineering, 2005, pp. 85-121

[2] Brent J Lance, et. al., "Brain-Computer Interface Technologies in the Coming Decades," Proc. IEEE, vol. 100, 2012, pp. 1585-1599

[3] Danhua Zhu, et. al., "A Survey of Stimulation Methods used in SSVEP-based BCIs," in Computational

Intelligence and Neuroscience, 2010, doi: 10.1155/2010/702357

[4] Yu Zhang, et. al., "LASSO based stimulus frequency recognition model for SSVEP BCIs," in Journal of Biomedical Signal Processing and Control, 2012, pp. 104-111, DOI: 10.1016/j.bspc.2011.02.002

[5] Sandra Mara Torres Muller, et. al., "Incremental SSVEP Analysis for BCI Implementation," in Ann. Int. IEEE EMBS Conf., Buenos Aires, Argentina, no. 32, 2010, pp. 3333-3336

[6] Kian B Ng, et. al., "Effect of competing stimuli on SSVEP-based BCI," in Proc. Ann. Int. IEEE Engineering in Medicine and Biology Society Conf., 2011, pp. 6307-6310, doi: 10.1109/IEMBS.2011.6091556

[7] R M Tello, et. al., "Evaluation of different stimuli color for an SSVEP-based BCI," in Congresso Brasileiro de Engenharia Biomedica, 2014, pp. 25-28

[8] M A Lopez, et. al., "Evidences of cognitive effects over auditory steady-state responses by means of artificial neural networks and its use in brain-computer interfaces," Neurocomputing, pp. 3617-3623, 2009, doi: 10.1016/j.neucom.2009.04.021

[9] Do-Won Kim, et. al., "Classification of selective attention to auditory stimuli: Toward vision-free brain-computer interfacing," in Journal of Neuroscience Methods, pp. 180-185, 2011, doi: 10.1016/j.jneumeth.2011.02.007

[10] T Nakamura, et. al., "Classification of Auditory Steady-State responses to Speech Data," in Ann. Int. IEEE EMBS Neural Engineering Conf., San Diego, California, 2013, pp. 1025-1028

[11] Alan J Power, et. al., "Extracting Separate Responses to Simultaneously Presented Continuous Auditory Stimuli: An Auditory Attention Study," in Int. IEEE EMBS Neural Engineering Conf., Antalya, Turkey, 2009, pp. 502-505

[12] Hiroshi Higashi, et. al., "EEG Auditory Steady State Responses Classification for the Novel BCI," in Ann. Int. IEEE EMBS Conf., Boston, Massachussets, 2011, pp. 4576-4579

[13] Kian B Ng, et. al., "Effect of posterized naturalistic stimuli on SSVEP-based BCI," in Proc. Ann. Int. IEEE Engineering in Medicine and Biology Society Conf., 2013, pp. 3105-3108, doi: 10.1109/EMBC.2013.6610198

[14] W. S. Stiles and B. H. Crawford, "Luminous Efficiency of Rays entering the Eye Pupil at Different Points," Nature, vol. 139, no. 3510, pp. 246-246, 1937

[15] Alexander McFarlane Mood, Franklin A Graybill, Duane C Boes, Introduction to the Theory of Statistics, McGraw-Hill, 1974

[16] Peggy Korczak, et. al., "Auditory Steady-State Responses," in Journal of the American Academy of Audiology, vol. 23, no. 3, 2012, pp. 146-170, doi: 10.3766/jaaa.23.3.3

[17] Hyekyung Lee and Seunhjin Choi, "PCA+HMM+SVM for EEG pattern classification," in Proc. Signal Processsing and its Applications, vol. 1, 2003, doi: 10.1109/ISSPA.2003.1224760

[18] Ali Ozgur Argunsahy and Mujdat Cetin, "AR-PCA-HMM Approach for Sensorimotor Task Classification in EEG-based Brain-Computer Interfaces," in Int. IEEE Pattern Recognition Conf., 2010, doi: 10.1109/ICPR.2010.36

[19] Faisal Bashir, et. al., "HMM-based Motion Recognition System using Segmented PCA," in Int. IEEE Image Processing Conf., vol. 3, 2005, pp. III-1288

[20] Roger J Lewis, "An Introduction to Classification and Regression Tree (CART) Analysis," in Ann. Meeting of the Society for Academic Emergency Medicine, San Francisco, California, 2000

[21] Kevin P. Murphy, "Naive Bayes classifiers," in University of British Columbia, 2006

[22] Liaw, Andy, and Matthew Wiener, "Classification and regression by randomForest," in R news, 2002, pp. 18-22