BCI Text Entry Using Hierarchical Keyboard With Probabilistically Dynamic Clustering

by

Ishrak Hayet (134425)
Tanveer Fahad Haq (134439)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh
November 2017

Approved By

Dr. Md. Kamrul Hasan

Associate Professor
Department of CSE, IUT

Hasan Mahmud

Assistant Professor
Department of CSE, IUT

Acknowledgements

We have our sincerest appreciation for the help, assistance and advice that many people have given us on countless occasions during the course of our undergraduate work. First, we would like to express our profound gratitude to our supervisor, Dr. Md. Kamrul Hasan sir for his guidance, support and encouragement. He has been a wonderful supervisor, who has continuously inspired and motivated us to complete many challenging research assignments. We also want to express our utmost regards to our advisor Hasan Mahmud sir. He has given his thoughtful suggestions, and we would like to thank him for all that. Acknowledgement is due to CSE department of Islamic University of Technology for supporting our B.Sc. thesis/project work.

We are very thankful to family members for supporting us throughout our B.Sc. academics. Above all, we would like to glorify Almighty Allah Subhanahu Wa Ta’la for giving us knowledge and patience to carry out this work.

Declaration of Authorship

We Ishrak Hayet (134425) and Tanveer Fahad Haq (134439) declare that this thesis titled "BCI Text Entry Using Hierarchical Keyboard With Probabilistically Dynamic Clustering" and the works presented in it are our own. We confirm that:

- This work has been done on partial fulfillment of the Bachelor of Science in Computer Science and Engineering degree at Islamic University of Technology.
- Any part of thesis has not been submitted anywhere else for obtaining any degree.
- Where we have consulted published work of others, we have always clearly attributed the sources.

Submitted by:

Ishrak Hayet (134425)

Tanveer Fahad Haq (134439)

Contents

List of Figures

List of Tables

Abstract

The ability to feel, adapt, reason, remember and communicate makes human a social being. Disabilities limit opportunities and capabilities to socialize. With the recent advancement in brain-computer interface (BCI) technology, researchers are exploring if BCI can be augmented with human computer interaction (HCI) to give a new hope of restoring independence to disabled individuals. This motivates us to lay down our research objective, which is as follows. In this study, we propose to work with a hands-free text entry application based on the brain signals, for the task of communication, where the user can select a letter or word based on the intentions of left or right hand movement, and left, right, up or down nodding movement. The three major challenges that have been addressed are (i) interacting with only four imagery signals (ii) how a low-quality, noisy EEG signal can be competently processed and classified using novel combination of feature set to make the interface work efficiently, and (iii) using a language prediction model to increase characters per minute.

Keywords

Brain Computer Interface, Human Computer Interface, EEG, CSP, Hierarchical Keyboard, Probabilistically Dynamic Clustering

1. Introduction

About 15% of the world's population, some 785 million people in the globe are suffering from mental and physical disabilities, including about 5% of children, according to a new report prepared jointly by the World Health Organization and the World Bank [3]. Of these, over 5% of the world's population, that is, 360 million people have motor disabilities (328 million adults and 32 million children), where most of the people are from developing and under developed countries [3]. People with motor disabilities experience difficulties coping with the demands that are placed upon them from the environment. They always depend on some individual to communicate and restrict themselves from entertainments and joyful society.

Over years the way humans interact with computers have made remarkable progress, from punch cards to swipe cards to touchless system. Nevertheless, the HCI system designed for disabled lags behind. In fact, for those people with severe disabilities, who can't use their hands and legs properly, only HCI-based systems are not compatible to meet their special needs. If we take a look at the various assistive technologies [6] for disabled users, be it screen readers, eye tracker or something else, we find that all of these use the same data flow path from human brain to hands or some other body part to computer peripherals like camera, keyboard to the computer memory or CPU. What if the humans only think actively and computers somehow understands the users' intention? This forms the underlying capability of BCI where it distinguishes different patterns of brain activity, each being associated to a particular intention or mental task, to directly control the HCI application. Hence, augmenting BCI with HCI and creating hands-free, touch-free applications can greatly enhance the life of people with disabilities.

At present, quite a few functional imaging modalities like EEG, MEG, fMRI, fNIRS, etc. are available for research [16]. Among them the electroencephalography (EEG) is unique and most often used, since it promises to provide high temporal resolution of the measured brain signals, relatively convenient, affordable, safe and easy to use BCI for both healthy users and the disabled. The term EEG is the process of measuring the brain's neural activity as voltage fluctuations along the scalp due to the current flow between neurons. BCI technology has traditionally been unattractive for serious scientific investigation. However, this context undergone radical changes over the last two decades. Now it is a flourishing field with a huge number of active research groups all over the world [16]. The assistive technologies available nowadays lack either speed or accuracy or both. These technologies require interactions that are explicit and exhaustive [6]. Moreover, it is not easy to apply BCI systems to operate an application like virtual keyboard. There are two main challenges associated with this. First, multi-class interaction, that is, more than two gesture usages will reduce the accuracy of the system and make the protocol design complicated [10]. Hence, the system should be designed to work with minimal number of interaction. Second, how a poor quality, noisy EEG signals from an affordable device can be expertly processed to work the interface effectively.

To overcome the above problems, we proposed a BCI augmented text-entry application for people with limited mobility, so that the user can interact with the system using their left and right hand motor imagery brain signals. The proposed approach would be with comparable performance for disabled and able-bodied users, and would not be exhaustive to work with. More specifically our goal is to explore and evaluate various pre-processing methods and feature set, to combine them with classification methods and to finally find a good set of techniques for our project. Besides, users do not have to use any additional equipment like eye tracker, screen readers to improve the system performance.

1.1. Problem Statement

Our goal is to build a BCI based application that will help motor disabled and/or speech-disabled patients or both to communicate with their environment and other people. Our focus to develop a hierarchical soft keyboard layout with dynamic clustering of letters and an integrated language model that can potentially augment the characters per minute of an average user considerably.

Existing assistive technologies do not address the issue of immobility of users. This poses a problem because the aforementioned class of users cannot make use of their limbs to interact with machines. This warrants the construction of a system that can provide a hands-free mode of interaction to these specific set of users.

1.2. Related Work

1.2.1. Steady State Visually Evoked Potentials (SSVEP)/P300

Farewell et al. [7] describe the development and testing of a P300 component based system for one to communicate through a computer. The system is designed for users with motor disabilities. The alphabets are displayed on a computer screen which serves as the keyboard or prosthetic device. The subject focuses on characters he wishes to communicate. The computer detects the chosen character on-line and in real time, by repeatedly flashing rows and columns of the matrix. The data rate of 0.20 bits/sec were achieved in this experiment. BCI system such as SSVEP or P300 [8] involves visual stimulations to control the interface. It uses flickering LEDs and continuous use of this may cause eye fatigue, epileptic seizures and visual impairment.

1.2.2. Code Modulation of Visually Evoked Potentials (c-VEP)

Scherer et al. [14] use a specially designed GUI and employed three motor imagery signals to control the interface. One MI signal for scrolling the letters and the other two to decide a target letter. Here, the typing speed attained was about 4 cpm. Bin et al. [5] aim to improve the low communication speed of the EEG based BCI systems based on code modulation of visual evoked potentials (c-VEP). The target stimuli were modulated by a time-shifted binary pseudo-random sequence. The on-line system achieved an average information transfer rate (ITR) of 108±12 bits/min on five subjects and with a maximum ITR of 123 bits/min for a single subject.

Changes in μ (8–12 Hz) and β (18–25 Hz) rhythms are associated with motor imagery signals [4]. Prasad et al. [11] aim at goal-directed rehabilitation tasks leads to enhanced functional recovery of paralyzed limbs among stroke sufferers. It is based on motor imagery (MI) and is also an EEG-based BCI. The MI activity is used to devise neuro-feedback for the BCI user to help him/her focus better on the task. The positive gains in outcome measures demonstrated the potential and feasibility of using BCI for post-stroke rehabilitation.

1.2.3. Predictive Spelling Program

Ryan et al. [13] compared a conventional P300 speller brain-computer interface with a predictive spelling program. Time to complete the task in the predictive speller (PS) condition was 12 min 43 s as compared to 20 min 20 s in the non-predictive speller (NS) condition. Even though there is marked improvement in overall output, accuracy was significantly higher in the NS speller. These results demonstrate the potential efficacy of predictive spelling in the context of BCI. Multi-degree BCI system [10] can be used to control various selecting tasks of the user interface like left, right, up, down, backwards, close, delete, open etc. But the accuracy of the classification progressively reduces when the number of classes' increases and more mental task adds complexity in the protocol design.

Figure 1 P300 Speller

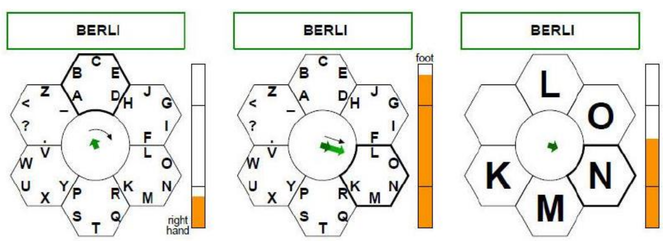

Figure 2 Berlin BCI

1.2.4. Context Information

Results have been published on the design of the RSVP Keyboard - a brain computer interface (BCI) for expressive language generation for functionally locked-in individuals using rapid serial visual presentation of letters or other symbols such as icons [6]. The proposed BCI design tightly incorporates probabilistic contextual information obtained from a language model into the single or multi-trial event related potential (ERP) decision mechanism. The tight fusion of contextual information with instantaneous and independent brain activity is demonstrated to potentially improve accuracy in a dramatic manner. Specifically, a simple regularized discriminant single-trial ERP classifier’s performance was increased from a naive baseline of 75% to 98% in a 28-symbol alphabet operating at 5% false ERP detection rate. Results show that trained healthy subjects can achieve real-time typing accuracies over 90% mostly relying on single-trial ERP evidence when supplemented with a rudimentary n-gram language model. Further discussion and preliminary results included initial efforts involving a locked-in individual and efforts to train him to improve his skill in performing the task.

1.2.5. Letter Prediction

In the paper, "INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER THROUGH HMMs", a brain-operated typewriter which is accelerated by a language prediction model was proposed. The proposed system uses three kinds of strategies to improve the entry speed: word completion, next-syllable prediction, and next word prediction. It was found that the entry speed of BCI-based typewriter improved about twice as much through the demonstration which utilized the language prediction model.

2. Proposed System

Our proposed system will comprise of two subsystems, namely, a BCI subsystem and an HCI subsystem. Both of these systems will follow a standard pattern recognition paradigm.

2.1. BCI Subsystem

The BCI sub system involves all the steps of a typical components of BCI system as shown in Fig. 1. The BCI sub system is developed in two steps, first creating a model for the application, that is, off-line analysis, second real-time processing of captured signals, that is, on-line analysis. The BCI subsystem consists of the following steps which represent a pattern recognition paradigm:

- Data Acquisition
- Preprocessing of raw data
- Feature Extraction
- Classification
- Evaluation

2.2. HCI Subsystem

In our study, we preferred dynamic keyboard for text-entry purpose. This dynamic keyboard ranks good among people with disabilities. This virtual keyboard is designed to have big selection boxes, each box containing letters or words that can be selected for typing as shown in Fig. 2. Each box consists of five letters. Five such boxes are arranged at the top with optimum design aspects in mind. Three control buttons (Space, Backspace and Options) are arranged at the bottom layer.

The keyboard coupled with a language prediction model produces letter and word prediction to aid a user with spelling. Whenever a user locks into a select letter, above that letter box, a vertical column consisting of a list of 5 word predictions will appear. The user can then execute Up/Down nod movement thoughts to traverse the column.

The cluster of letters is also dynamic so that each letter has neighboring letters which are closest to it in terms of likelihood when a user's currently typed text is the prior condition.

These conditions are prerecorded in a user's personalized corpus. This greatly reduces the search time required to select a word.

The HCI subsystem will consist of two components:

- A hierarchical soft keyboard with dynamic clustering.
- Language prediction model

2.3. Language Prediction Model

Languages have their sequences. Language modeling is done to solve these phenomena. It can predict the next word when given the previous words. This task is used in speech or optical character recognition, statistical machine translation, and these tasks are often referred to as a Shannon game [18]. Here, we deal with a theoretical language model using the n-gram which is described in [19] and how it can be applied to Korean prediction modeling. The n-gram model is a simple and strong probability model which adopts the modeling of actual language. n is the size of the previous word or syllable. When the words w_1 ,...,w_m are given, the sequence for the probability in an n-gram model is as follows:

$$\mathrm{P}(w_1, \dots w_m) = \prod_{i=1}^{m} P(w_i | w_1, \dots , w_{i-1})$$
$$\approx \prod_{i=1}^{m} P(w_i \middle| w_{i-(n-1)}, \dots , w_{i-1})$$

Here, it is assumed that the probability of observing the nth word w_i in the context history of the preceding i-1 words can be approximated by the probability of observing it in the shortened context history of the preceding n-1 words that follows the nth order Markov property. For example, in a bigram(n=2) language model, the probability of the sentence “I saw the red house” is approximated as P(I, saw, the, red, house) = P(I| < s >)P(saw|I)P(the|saw)P(red|the)P(house|red). Each probability of n-grams can be calculated from the frequency counts in the corpus as shown below:

$$P(w_i \middle| w_{i-(n-1)}, \dots , w_{i-1}) = \frac{Count(w_{i-(n-1)}, \dots, w_{i-1}, w_i)}{Count(w_{i-(n-1)}, \dots, w_{i-1},)}$$

When predicting the next word or next syllable, we most likely need wi. This can be calculated as follows:

$$w_i = argmax_{w_i} P(w_1, \dots, w_i) \approx argmax_{w_i} P\big(w_{i-(n-1)}, \dots, w_i\big)$$

This expression occurs earlier in the corpus by calculating the frequency of syllables and words receiving the highest probability of the current syllable or word as recommended by the system. Accordingly, the number n refers to uni-gram, bi-gram and tri-gram. We use the word uni-gram, the syllable bi-gram, and the word bi-gram in order to predict the syllable.

2.3.1. Incorporation of a language model through HMMs

The P300 speller is one of the most common types of BCI that enables a subject to write text on a computer screen. The P300 speller in the brain as a response to a visual or auditory stimulus. Columns and rows of a matrix of characters (See Fig. 1) flash randomly as the subject attends to one character. This paradigm was introduced by Farwell and Donchin in [4]. P300 is an event related potential (ERP) that occurs in the brain as a response to a visual or auditory stimulus. Columns and rows of a matrix of characters (See Fig. 1) flash randomly as the subject attends to one character. The brain is expected to generate a P300 response for the flashes containing the attended character. Due to the low SNR and variability of EEG signals, P300-based BCI typing systems need several number of stimulus repetitions to increase classification accuracy, which causes low symbol rates [7, 8]. Various aspects of the P300 speller were examined to improve the performance such as electrode selection, stimulus shape and dimension, different flashing paradigms [9] and several signal processing and classification methods [10, 11]. However, the idea of integration of a language model into the decision making algorithm to predict the current letter using the previous letters is not common. Speier et al. [12] proposed a natural language processing (NLP) approach which exploits the classification results on the previous letters to predict the current letter based on learned conditional probabilities. Orhan et al. [7] created a system using a non-conventional flashing paradigm, the RSVP keyboard, and merged the context-based letter probabilities and EEG classification scores by using a recursive Bayesian approach. Both of these ideas showed that integrating information about the linguistic domain can improve the speed and accuracy of a BCI communication system.

In the paper, "INCORPORATION OF A LANGUAGE MODEL INTO A BRAIN COMPUTER INTERFACE BASED SPELLER THROUGH HMMs", researchers propose a new approach for the integration of a language model and the EEG scores based on a second order Hidden Markov Model (HMM). We use Forward-Backward and Viterbi algorithms to make decisions on the letters typed by the subjects. There have been several works on HMM applications of

BCI-based contexts, such as in [11]. However, using HMM based on a language model as we propose has not been practiced before. Our work is significantly different from previous work in [7, 12]. The approach in [12] is greedy in the sense that the prediction for the current letter is performed conditioned only on the letters declared by the system for the previous time instants. On the other hand, our approach is fully probabilistic. It acknowledges that previous decisions contain uncertainties as well, and performs prediction by considering the computed probabilities of all letters in the previous instant(s), rather than just the declared ones. Both [7] and [12] exploit information in the previous letters for the current letter. In contrast, our approach takes advantage of both the past and the future. In this way, previously declared letters can be updated as new information arrives. We present experimental results based on EEG data collected in our laboratory through P300-based spelling sessions. We consider both the original measurements, as well as their noisy versions to test the robustness of the approach to reductions in SNR. Our results show that the speed and the classification accuracy of the BCI system can be improved by using the proposed approach in both noiseless and the noisy case.

2.3.2. Simulation of Proposed System:

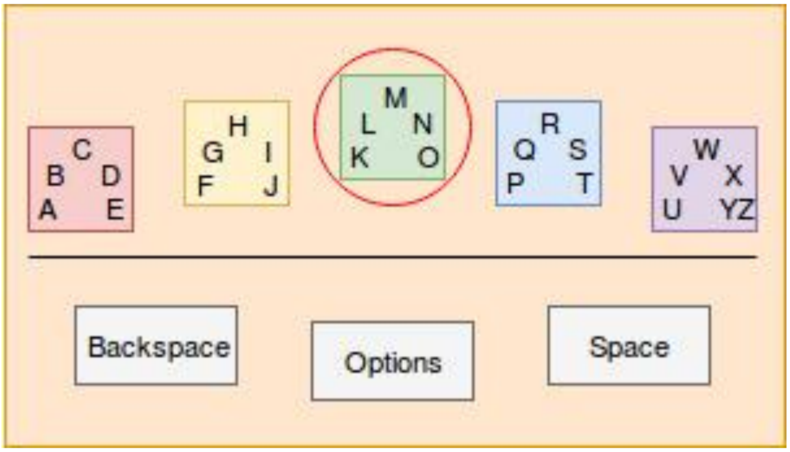

Figure 3 Dynamic Clustering - Step 1

At the beginning, the user is at the first cluster. The user will need to execute 2 successive right movements successfully to get to the cluster marked in red circle.

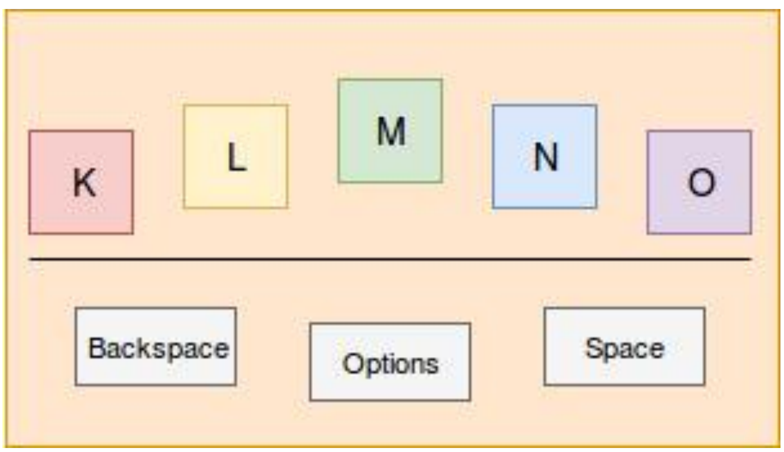

Figure 4 Dynamic Clustering - Step 2

Once the user dwells on the cluster containing N, s/he will enter this screen, and start at the first block.

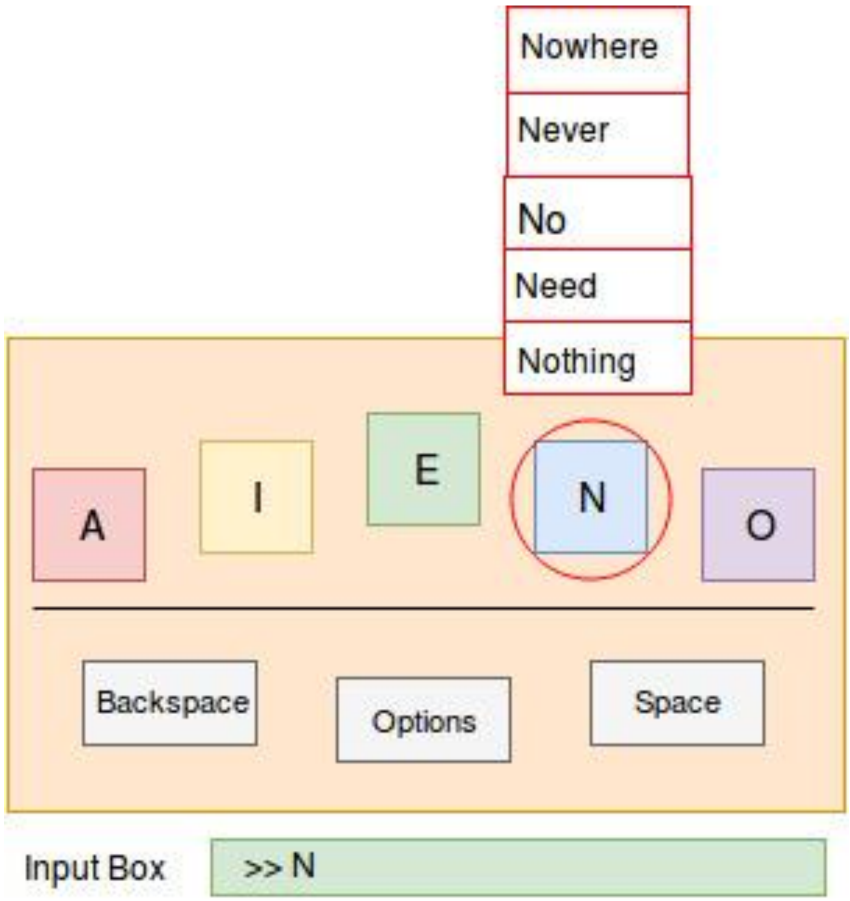

Figure 5 Dynamic Clustering - Step 3

The user performs three successive right thoughts and reaches the N block. After dwelling on N, N is written in the Input Box and vertical prediction list appears. At the same time, the other blocks of this cluster are changed according to the alphabet proximity from N.

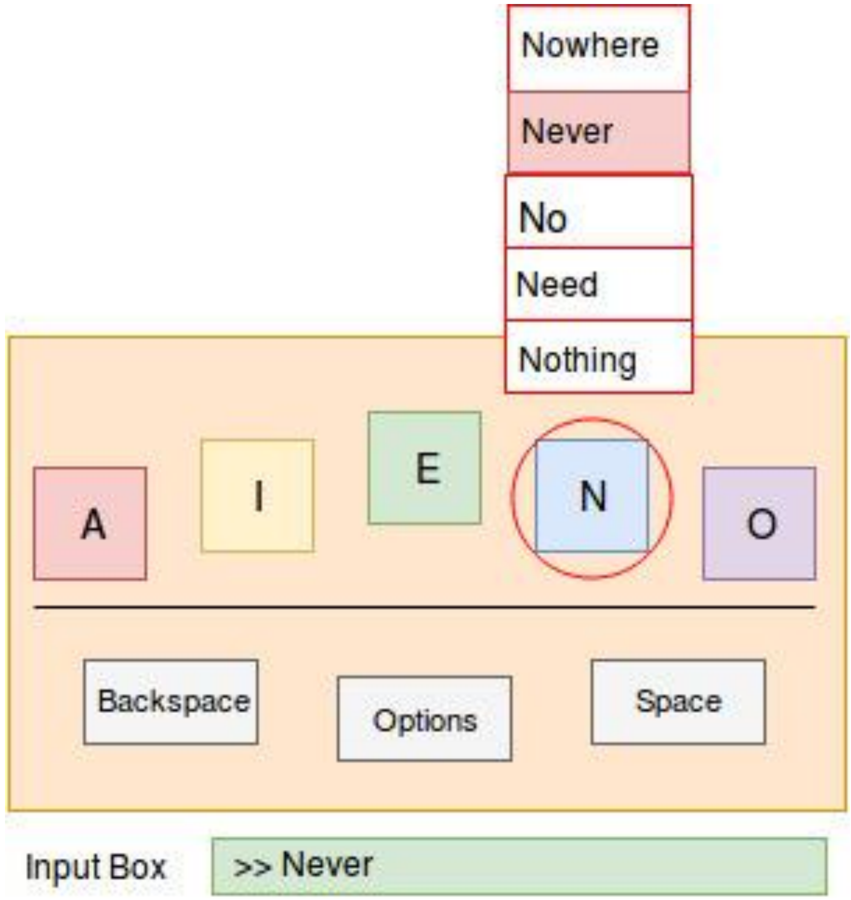

Figure 6 Dynamic Clustering - Step 4

The user performs up/down thoughts and selects a predicted word by dwelling on the word. The word is written in the Box and the user is taken to the 1st layer of the keyboard.

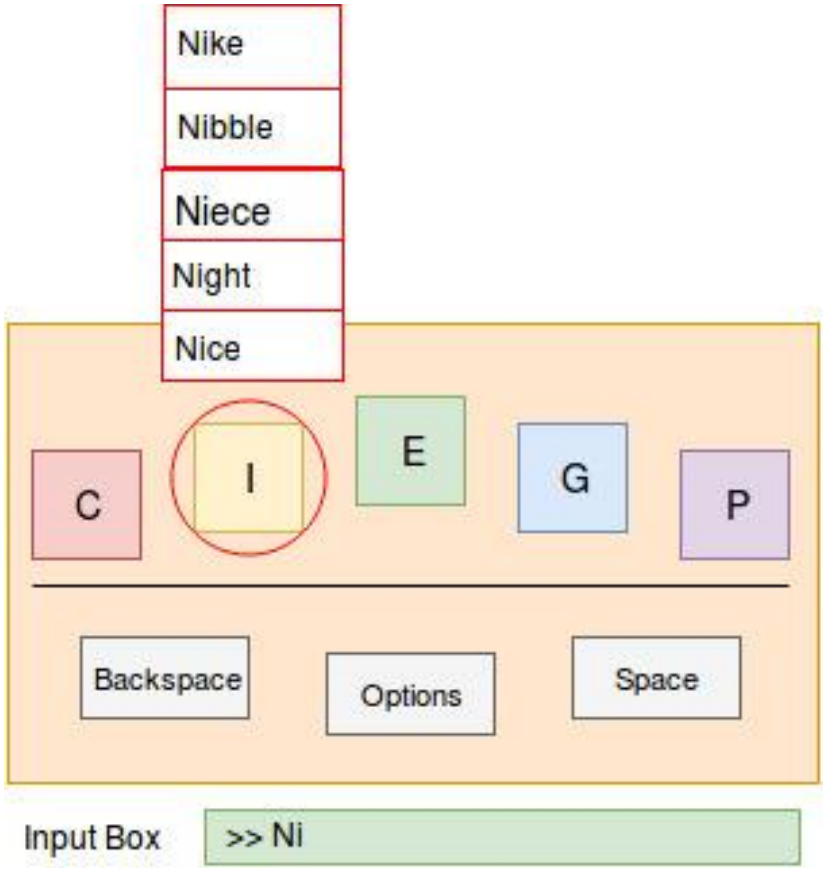

Figure 7 Dynamic Clustering - Step 5

After selecting N, the user chooses I from the new cluster and dwells on it. I is written after N in the Box and corresponding vertical prediction appears. The blocks of the cluster are again changed according to alphabet proximity from “Ni”.

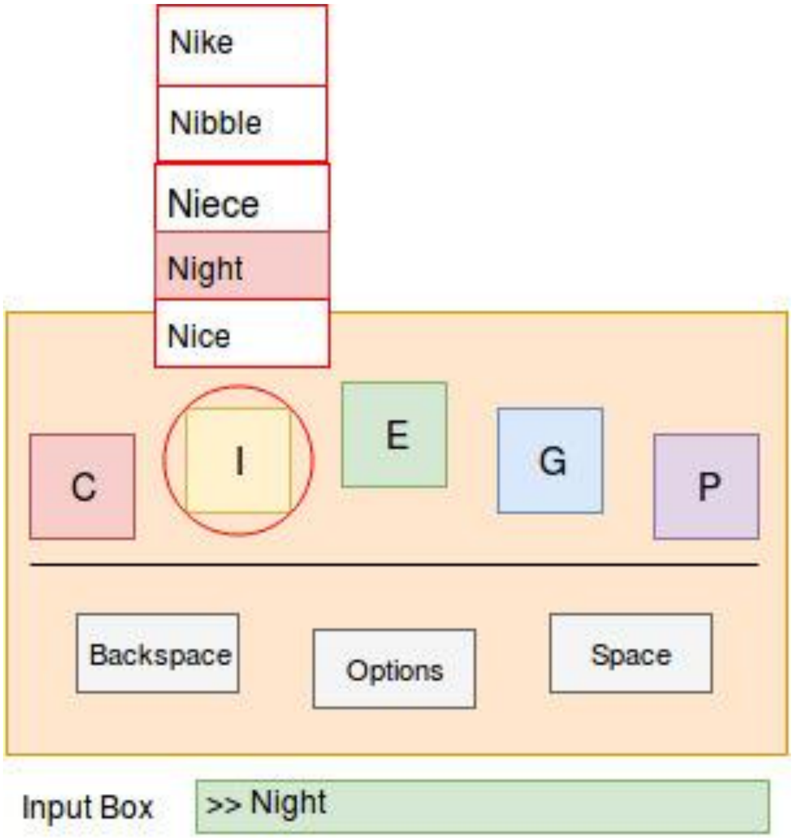

Figure 8 Dynamic Clustering - Step 6

The user selects Night from the predictions by dwelling on it. Night is written in the box and the user is taken to the 1st layer of the keyboard.

3. Experimentation

3.1. Hardware

The Emotiv EPOC+ [2] headset is placed over the scalp according to International 10–20 electrode placement system. It consists of 16 electrodes - 14 measuring electrodes and two reference electrodes. The EEG measuring sensors are AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, and O2.

The 10/20 system or International 10/20 system is an internationally recognized method to describe the location of scalp electrodes. The system is based on the relationship between the location of an electrode and the underlying area of cerebral cortex. The numbers '10' and '20' refer to the fact that the distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance of the skull. Each site has a letter to identify the lobe and a number to identify the hemisphere location.

Table 1 Electrode Placement

Electrode	Lobe
F	Frontal
T	Temporal
C	Central*
P	Parietal
O	Occipital

The reference sensors are P3 (Common Mode Sense - CMS) and P4 (Driven Right Leg - DRL). The sampling rate is 128 sps, cut-off frequency of low pass filter is 45 Hz and the resolution is 14 bits. The recorded EEG signals are transferred to the computer using wireless USB connector. The device has three types of control such as EEG, EMG and Gyroscope. It has fewer scalp contact than expensive and sophisticated devices.

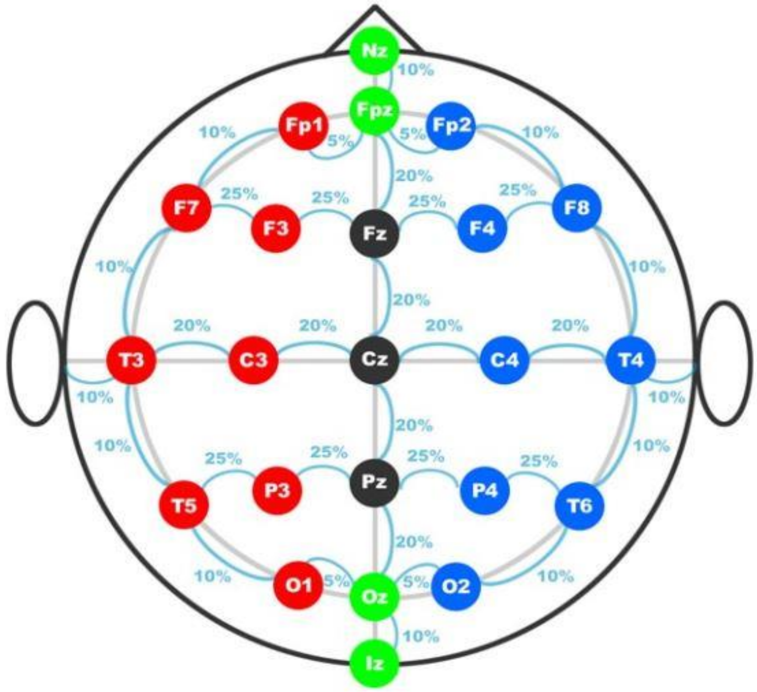

Figure 9 10-20 System for EEG electrode

3.2. Participants

Three healthy subjects, right-handed, between ages of 23 and 29, volunteered for experiment. Each subject themselves cooperated and willingly provided written informed consent before participating in the experiment. None of the subjects had prior BCI training. The subjects were provided with information only related to the activities they performed. They were not informed about the experimental design and the hypothesis/objective of the study. The experimental setup was done at the BCI LAB facility at IIT Kharagpur. Subjects were seated in chair with their arms extended, resting on the desk and their legs extended, resting on a footrest. The lab was well illuminated by artificial lights, and there was no background noise while the data was recorded. The subjects performed or kinesthetically imagined left and right hand movements, and the EEG signals were recorded on all 16 channels using the Emotiv EPOC+ device and Emotiv API for Java. The subjects were provided with stimulus on GUI (Fig. 1) designed using a custom designed tool. They were provided with two stimulus, a cross figure that marked the beginning of the trial followed by a left or a right arrow on which the task was performed, recording EEG signals for MI for nearly 4 s. The stimulus was presented on a nineteen-inch LCD monitor, kept in-front of the subject making a viewing angle of approximately 1.5◦. The subject was focused on the stimulus during the entire session. The experiments were performed in presence of a researcher, with no interaction with the subject during the recording session.

3.3. Details, Data, Tools

Table 2 Experimentation Details

Experimentation Details	Session One	Data Collection Duration 500ms Small Dataset
	Session Two	Data Collection Duration : 7s Large Dataset
Experimentation Data	Arm Lift (14 Channel)	Right, Left
	Nodding (14 Channel)	Up, Down, Left, Right
Experimentation Tools	Software	Emotiv SDK – Java Custom tool for visual cues and time synchronization
	Hardware	Emotiv EPOC+ EEG Headset

3.4. Initial Experiment Procedure:

1. All the surrounding pieces of equipment running on alternate current were switched off to reduce the effect of noise
2. The subject was seated in a relaxing position with as less distractions as possible
3. For each class of the data, the subject was shown a corresponding cue while having been thinking of the representing orientation
4. The data was recorded according to the duration specified for session one and two
5. Data tuple:

channel 1	channel 2	...	channel 14	class label

3.5. Data Acquisition

Subjects were seated in chair with their arms extended, resting on the desk and their legs extended, resting on a footrest. The lab was well illuminated by artificial lights, and there was no background noise while the data was recorded. The subjects performed or kinesthetically imagined left and right hand movements, and the EEG signals were recorded on all 16 channels using the Emotiv EPOC+ device and our custom Java tool for data acquisition. The subjects were provided with stimulus on GUI (Fig. 1) designed using Java and Emotiv SDK. They were provided with two stimulus, a cross figure that marked the beginning of the trial followed by a left or a right arrow on which the task was performed, recording EEG signals for MI for nearly 4 s. The stimulus was presented on a nineteen-inch LCD monitor, kept in-front of the subject making a viewing angle of approximately 1.5◦. The subject was focused on the stimulus during the entire session. The experiments were performed in presence of a researcher, with no interaction with the subject during the recording session.

Figure 10 Emotiv EPOC+

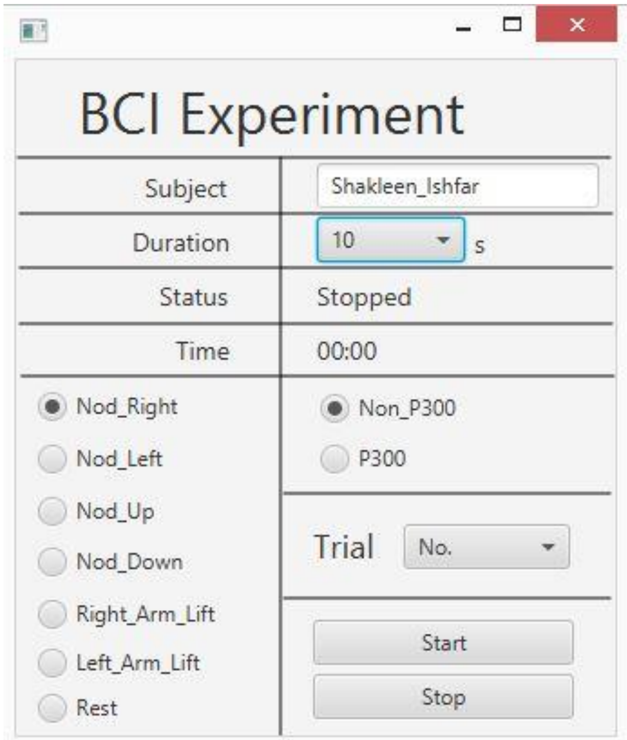

Figure 11 Custom EEG Acquisition Tool

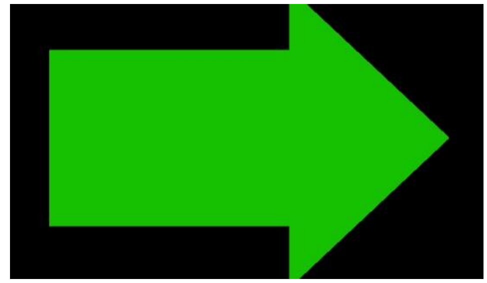

Figure 13 Right Cue

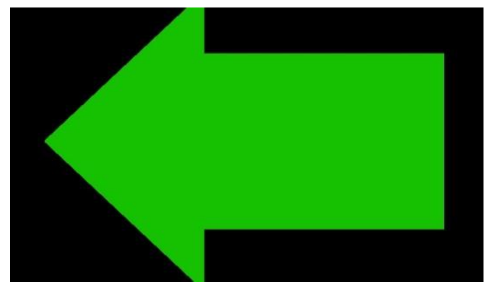

Figure 12 Left Cue

3.6. Data Labeling

We arrange two sessions with each subject. Each session lasted about 10 minutes, including the time for experimental setup and data recording. The sessions were held on different days. Each session, had 10 trials of each orientation shown in table 2. The subjects were told to imagine the corresponding movements. The subjects were shown left, right, up, down arrows in random order on the screen and subjects had to think about lifting/moving the corresponding arm. Subjects were instructed to focus on the cues to minimize noisy reading. To minimize the artifacts in the recordings, subjects were asked to minimize eye blinks, jaw and head movements during recording. They were allowed to swallow, blink and adjust to relax during the cross presentation. The duration of cross presentation was adjusted if required for subjects, and also varied randomly. The data is recorded into a single file as a continuous stream of signals.

The dataset for each direction was concatenated for all the trials. The data for the gyroscope was removed and each data tuple was labeled according to the cues.

4. Preprocessing

Signal preprocessing aims to remove the noise and enhance the information content of the EEG signal. The raw EEG signals captured are preprocessed, before performing feature extraction and classification. Another reason for preprocessing is the presence of artifacts, that is, parts of signal due to background brain activity, which may lead to incorrect conclusions. It is an important step as it increases the information content of the raw signal. The artifacts due to power line interference and other artifact above 50 Hz can be easily removed using a notch filter, or a band-pass filter that allowed only signals in the range 8–30 Hz.

ICA, theoretically is the best method among itself, CAR, CSP and regularized CSP for the preprocessing [1], but it is very memory intensive. The requirements of large computation time make it unfit to be used in such project. In our study, we experimented with CSP with respect to variations in Windowed and Global CSP.

4.1. Common Spatial Pattern

Common spatial pattern (**CSP**) is a mathematical procedure used in signal processing for separating a multivariate signal into additive subcomponents which have maximum differences in variance between two windows. The use of variance between the channels limits the random artifacts introduced such as eye-blinks. CSP has become a standard and popular preprocessing method in the domain of processing Electro Encephalography Signal.

Let X_1 of size (n, t_1) and X_2 of size (n, t_2) be two windows of a multivariate signal, where n is the number of signals and t_1 and t_2 are the respective number of samples. The CSP algorithm determines the component w^T such that the ratio of variance (or second-order moment) is maximized between the two windows:

$$w = \arg max_w \frac{||wX_1||^2}{||wX_2||^2}$$

The solution is given by computing the two covariance matrices:

$$R_1 = \frac{X_1 {X_1}^T}{t_1}$$

$$R_2 = \frac{X_2 {X_2}^T}{t_2}$$

where, t_1 and t_2 are the trace matrices of $X_1 {X_1}^T$ and $X_2 {X_2}^T$ respectively.

Then, the simultaneous diagonalization of those two matrices (also called generalized eigenvalue decomposition) is realized. We find the matrix of eigenvectors $P = p_1, \dots, p_2$ and the diagonal matrix D of eigenvalues $\{\lambda_1, \dots, \lambda_n\}$ sorted by decreasing order such that:

$$P^{-1} R_1 P = D$$

And

$$P^{-1} R_2 P = I_n$$

With I_n the identity matrix.
This is equivalent to the Eigen decomposition of ${R_2}^{-1} R_1$:

$${R_2}^{-1} R_1 = PDP^{-1}$$

w^T will correspond to the first column of P :

$$w = {p_1}^T$$

Arbitrary number of columns can be selected from P. However, the first column is the one that maximizes the variance along one class and minimizes it along another class. Hence, it isolates the different classes. The number of columns selected from P decides the new number of channels of the dataset. We have chosen different values for the number of channels and have come up with different results.

At the same time, we have performed a few variations in CSP. These are as follows:

4.1.1. Global CSP

The $X_1$and X_2 classes were respectively considered to be consisting of all the samples of one class and all the samples of the other class. The resultant weight vector w then was a $14 * c$ matrix. And, each tuple was a $14 * 1$ vector. So, performing the following computation gave us a vector consisting of c rows, where c is the number of channels selected from the CSP.

$$x' = w^T * x$$

Here, x' is the tuple in the new dimension, w^T is the transpose of the weight vector and x is the original tuple.

4.1.2. Windowed CSP

The $X_1$and X_2 classes were respectively considered to be consisting of n number of samples of one class and n number of samples of the other class (window). The resultant weight vector w then was a $14 * 1$ matrix. And, each tuple was an $n * 14$ matrix. So, performing the following computation gave us a matrix of dimension $1* n$, where $c = 1$ is the number of channels selected from the CSP.

$$x' = x * w$$

Here, x' is the tuple in the new dimension, w is the weight vector and x is the original window of the tuple. The philosophy behind this method was to account for the non-stationary characteristic of EEG signals. However, this method has the limitation that different windows give different weight vectors. Selecting the best amongst these weight vectors was difficult and the weight vectors suffered from the problem of biased locality.

4.1.3. Average of Windowed CSP

The problem with the Windowed CSP was that no suitable metric exists to choose the best weight vector from those given by the different windows. So, an average of the window weight vectors were considered as the chosen weight vector.

The $X_1$and X_2 classes were respectively considered to be consisting of n number of samples of one class and n number of samples of the other class (window). The resultant weight vector w then was a $14 * 1$ matrix. And, each tuple was an $n *$

14 matrix. So, performing the following computation gave us a matrix of dimension $1 * n$, where $c = 1$ is the number of channels selected from the CSP.

$$w' = \frac{1}{number\ of\ windows} * \sum_{1}^{p} w_p$$

Where, w_p is the weight vector from p^{th} window.

$$x' = x * w$$

Here, x' is the tuple in the new dimension, w is the average weight vector and x is the original window of the tuple. The philosophy behind this method was to account for the non-stationary characteristic of EEG signals. The size of the window can be varied.

5. Feature Extraction

The dataset has been preprocessed using CSP. Now, the dataset has c number of channels reduced from 14 channels. So, further feature extraction has to be carried out on this dataset.

Features to be used are:

5.1. Signal Power

Sum of the absolute square values of the signal's time domain samples divided by the signal length

5.2. Time Frequency Analysis

5.2.1. Hilbert Transform and short time Fourier transform

A real function f(t) and its Hilbert transform f(t) are related to each other in such a way that they together create a so called strong analytic signal. The strong analytic signal can be written with an amplitude and a phase where the derivative of the phase can be identified as the instantaneous frequency. The Fourier transform of the strong analytic signal gives us a one-sided spectrum in the frequency domain.

The Fourier functions do not adequately represent non-stationary signals.

Therefore, appropriate windows have been applied to the Fourier functions which provide short time Fourier transform (STFT) is a type of Time-Frequency Representation (TFR).

5.2.2. Wavelet

If we have a high resolution in the frequency domain (i.e. focusing on one frequency, like sin(3x)) it is hard to isolate it in time, as each frequency exists across all time. Being uncertain of the time when focusing on the frequency is the flip-side of being uncertain of the frequency when focusing on time. To overcome this resolution problem a Wavelet Transform is used to deconstruct the signal into a load of wavelets being added together. Wavelets are useful because they are limited in time and frequency.

6. Classification

Classifiers build a model based on the training data, to distinguish between classes. Once the model is created, it could be used to label unseen data i.e. testing data. In order to classify the unseen data correctly the model build must be robust and accurate. SVM has been found to be most accurate among itself and LDA [1]. The goal of SVM is to find the optimal separation hyperplane which maximizes the margin of the training data.

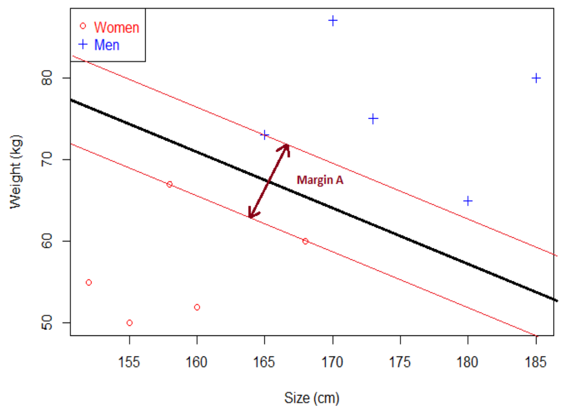

Figure 14 SVM Classification

We have used Weka API for Java as our data analysis tool. The SMO function of Weka was used to perform the classification. SMO classifier is a variant of SVM. The data tuples were not linearly separable in the original dimension. So, a kernel function had to be used to take the data to a higher dimension where they are linearly separable. Two types of kernel functions were experimented with. They are the RBF (Radial Basis Function) kernel and the Polynomial kernel. However, the RBF kernel worked better than Polynomial kernel. The parameters used to tune the SVM was the Gamma value and the Complexity value. The more complex the model was, the better it performed for the training set and worse for the test set because it could not generalize the dataset good enough to classify test data.

Since CSP is a two-class preprocessing tool, the classifier has to be binary so that it can recognize the stages of hierarchy. We intend to extend our classifier so that at first it classifies hand (left or right) or other (feet, tongue or nodding).

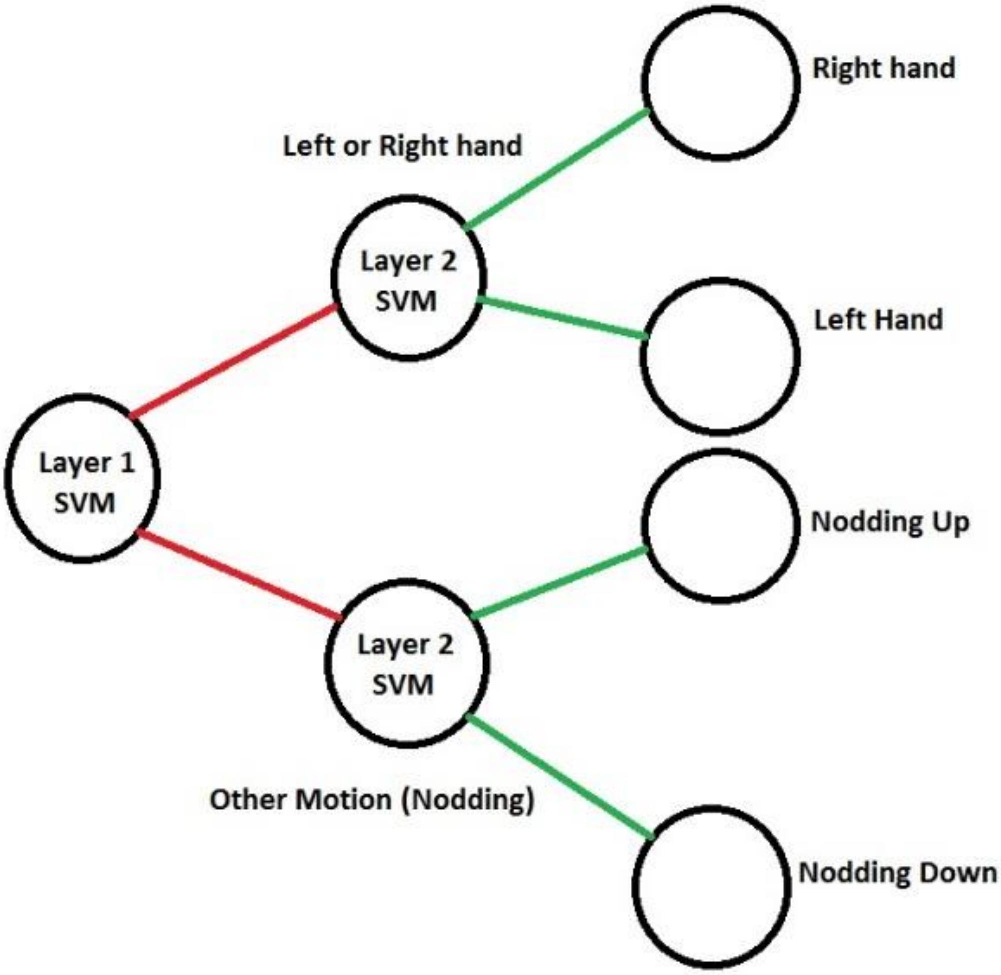

Figure 15 Hierarchical Classifier

7. Prototype

The keyboard application has been developed using Java. The keyboard has been designed according to our prototype. It is a hierarchical keyboard with three different types of clusters. The prediction cluster is at the top, the letter cluster is in the middle and the control cluster is in the bottom.

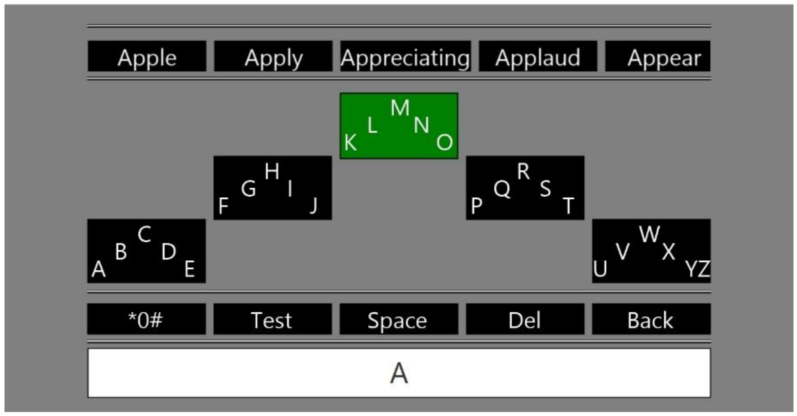

Figure 16 Soft Keyboard Prototype - Step 1

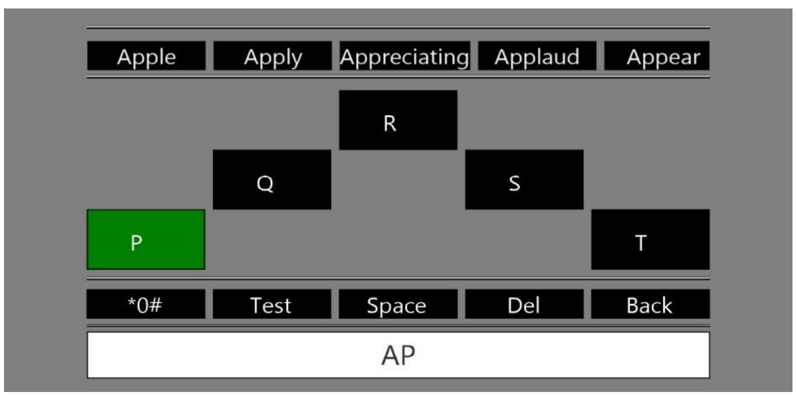

Figure 17 Soft Keyboard Prototype - Step 2

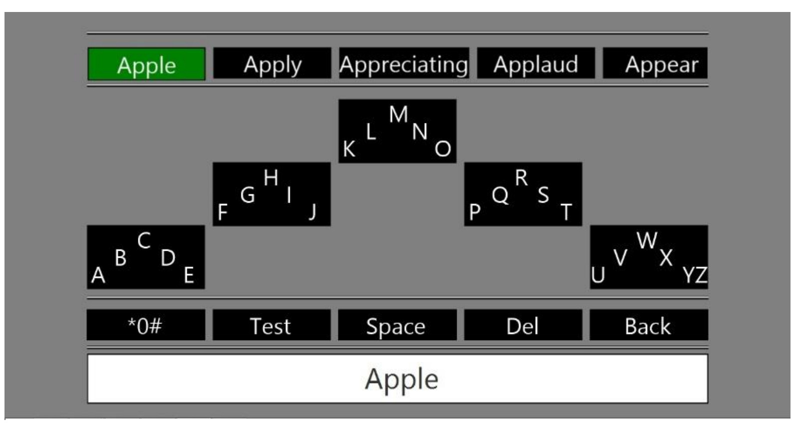

Figure 18 Soft Keyboard Prototype - Step 3

8. Results

Accuracy is higher for small dataset because an average user is only able to sustain his specific movement thought for only a few seconds (1 to 2 seconds) which corresponds to a small portion of the whole data collected over say 10 seconds.

Also a significant observation was that, with more number of data instances, the accuracy decreased. But, tuning the parameters improved the accuracy.

We have used CSP for feature extraction. We have tried different variations of CSP to preprocess data. Then, we performed SVM with Radial Basis Function as the Kernel (Gaussian). The gamma and C parameters were tuned.

For, letter and word prediction we used the probabilistic approach and achieved very good results with our initial text corpus. With time as the user uses the software, the corpus will be enhanced and performance will improve.

Table 3 Results

Serial No.	CSP Type	Results
1.	1. Global CSP	1. Correctly Classified Instances 919 52.2159 % Incorrectly Classified Instances 841 47.7841 % Kappa statistic 0.0443 Mean absolute error 0.2478 Root mean squared error 0.3575 Relative absolute error 98.989 % Root relative squared error 101.1073 % Total Number of Instances 1760

2.	2. Windowed CSP	2. No suitable weight vector selection mechanism was found
3.	3. Average of Windowed CSP	3. Correctly Classified Instances 39 48.75 % Incorrectly Classified Instances 41 51.25 % Kappa statistic -0.025 Mean absolute error 0.5125 Root mean squared error 0.7159 Relative absolute error 102.5 % Root relative squared error 143.1782 % Total Number of Instances 80

These metrics are considerable as a sign of progress. But, much higher accuracy is possible using feature extraction and optimized classifier parameters. Also, we have achieved better informal user experience using the dynamic letter prediction and word prediction.

9. Conclusion

We have completed partial background study, and formulated our problem statement. We have finalized our system design which accords with interaction design principles to meet usability and user experience goals. We have performed preliminary experiments to gather raw EEG data, and used it to develop intuitions on what methodologies to use for best results. We have done some preprocessing on the raw data, and shortlisted potential features we can extract from our dataset. We have performed some feature extractions and classified EEG signals with good accuracy. Currently, we are working on extracting more features to improve accuracy. Also, some parameter optimization technique will have to be automated to compute the classifier parameters.

We will perform comparative analyses on potential classifiers like Bayesian classifiers to see if they perform better than aforementioned classifiers like LDA and SVM. We will test our hypotheses by future training and testing, and make evaluations as to what improvements we can make in the keyboard design, and what performance parameters need to be changed to attain the ultimate goal which is to achieve a high characters per minute in our application.

10. References

1. Microsoft, Type without using the keyboard (On-Screen Keyboard). https://support.microsoft.com/en-in/help/10762/windows-use-on-screen-keyboard

2. Emotiv EPOC, Software Development Kit (2010). http://www.emotiv.com/researchers

3. World report on disability, World Health Organization (2011). http://www.who.int/disabilities/world report/2011/report/en

4. Alomari, M.H., Samaha, A., AlKamha, K.: Automated classification of L/R handmovement EEG signals using advanced feature extraction and machine learning. Int. J. Adv. Comput. Sci. Appl. 4(6) (2013)

5. Bin, G., Gao, X., Wang, Y., Li, Y., Hong, B., Gao, S.: A high-speed BCI based on code modulation VEP. J. Neural Eng. 8(2), 025015 (2011)

6. Crow, K.L.: Four types of disabilities: their impact on online learning. TechTrends 52(1), 51–55 (2008)

7. Farwell, L.A., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalogr. Clin. Neurophysiol. 70(6), 510–523 (1988)

8. Graimann, B., Allison, B., Pfurtscheller, G.: Brain-computer interfaces: a gentle introduction. In: Graimann, B., Pfurtscheller, G., Allison, B. (eds.) Brain Computer Interfaces, pp. 1–27. Springer, Heidelberg (2009)

9. Guo, L., Wu, Y., Zhao, L., Cao, T., Yan, W., Shen, X.: Classification of mental task from EEG signals using immune feature weighted support vector machines. IEEE Trans. Magn. 47(5), 866–869 (2011)

10. Long, J., Li, Y., Wang, H., Yu, T., Pan, J., Li, F.: A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. IEEE Trans. Neural Syst. Rehabil. Eng. 20(5), 720–729 (2012)

11. Prasad, G., Herman, P., Coyle, D., McDonough, S., Crosbie, J.: Using motor imagery based brain-computer interface for post-stroke rehabilitation. In: 2009 4th International IEEE/EMBS Conference on Neural Engineering, pp. 258–262. IEEE (2009)

12. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A.: Openvibe: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence 19(1), 35–53 (2010)

13. Ryan, D.B., Frye, G., Townsend, G., Berry, D., Mesa-G, S., Gates, N.A., Sellers, E.W.: Predictive spelling with a P300-based brain-computer interface: increasing the rate of communication. Int. J. Hum. Comput. Interact. 27(1), 69–84 (2010)

14. Scherer, R., Muller, G., Neuper, C., Graimann, B., Pfurtscheller, G.: An asynchronously controlled EEG-based virtual keyboard: improvement of the spelling rate. IEEE Trans. Biomed. Eng. 51(6), 979–984 (2004)

15. Spalteholz, L., Li, K.F., Livingston, N., Hamidi, F.: Keysurf: a character controlled browser for people with physical disabilities. In: Proceedings of the 17th international conference on World Wide Web, pp. 31–40. ACM (2008)

16. Wolpaw, J., Wolpaw, E.W.: Brain-Computer Interfaces: Principles and Practice. OUP USA, New York (2012)

17. Yong, X., Fatourechi, M., Ward, R.K., Birch, G.E.: The design of a point-and-click system by integrating a self-paced brain-computer interface with an eye-tracker. IEEE J. Emerg. Sel. Topics Circ. Syst. 1(4), 590–602 (2011)