Decoding Imagined Speech and Computer Control using Brain Waves

Abhiram Singh, Ashwin Gumaste

***Abstract*—In this work, we explore the possibility of decoding Imagined Speech brain waves using machine learning techniques. We propose a covariance matrix of Electroencephalogram channels as input features, projection to tangent space of covariance matrices for obtaining vectors from covariance matrices, principal component analysis for dimension reduction of vectors, an artificial feed-forward neural network as a classification model and bootstrap aggregation for creating an ensemble of neural network models. After the classification, two different Finite State Machines are designed that create an interface for controlling a computer system using an Imagined Speech-based BCI system. The proposed approach is able to decode Imagined Speech signal with a maximum mean classification accuracy of 85% on binary classification task of one long word and a short word. We also show that our proposed approach is able to differentiate between imagined speech brain signals and rest state brain signals with maximum mean classification accuracy of 94%. We compared our proposed method with other approaches for decoding imagined speech and show that our approach performs equivalent to the state of the art approach on decoding long vs. short words and outperforms it significantly on the other two tasks of decoding three short words and three vowels with an average margin of 11% and 9%, respectively. We also obtain information transfer rate of 21-bits-per-minute when using an IS based system to operate a computer. These results show that the proposed approach is able to decode a wide variety of imagined speech signals without any human-designed features.**

***Index Terms*—Brain-Computer Interface, Imagined Speech Artificial Neural Network, Electroencephalogram, Tangent Space, Principal Component Analysis, Finite State Machine.**

I. Introduction

BRAIN Computer Interface (BCI) is a combination of hardware (used to capture brain signals) and software (for analysis and understanding of different cognitive tasks). Nowadays, research in BCI is getting popular because of its possibilities to study human behavior, diagnose brain disease, and its possible use in human-computer interface (HCI) devices. A BCI system can be seen as an add-on for existing technologies such as touch screen, mouse, or keyboard. Many BCI systems exist using different paradigms such as P300 or motor imaginary for Human-Computer Interaction (HCI) [1].

Various activities generate electrical signals from the brain. Imagined speech (IS) or speech imaginary [2] is one such class of brain signals in which the user speaks in the mind without explicitly moving any articulators. IS is different from silent speech, in which a user thinks to move articulators during the imagination of words. Hence, silent speech is likely to generate signals from the motor cortex of the brain, and IS signals are usually generated from Broca's and Wernicke's area [3].

There exist different techniques to capture electrical signals from the brain. Electroencephalography (EEG) [4] is one such widely used technique that involves placing electrodes over the scalp in a non-invasive fashion. These electrodes capture voltage differences generated due to ion movement along the brain neurons. These measurements are obtained over a period to form an EEG signal. The number of electrodes can vary from sparse (just 1) to dense (256), which is usually determined based on application requirements.

This work focuses on the classification of IS signals. The reason behind using IS signals is because this technique is expected to take less training time and is more comfortable than motor imaginary tasks, as well as provides a natural way towards HCI. IS signals may lead to overall improved user experience in computer interaction. The work in this paper assumes that the data is not fully corrupted with noise. Subjects participating in IS experiments are instructed to follow specific guidelines, making this assumption feasible (though this may not always be true in a real-life scenario). So, there is a possibility of extracting useful information related to the imagined speech and, thereafter decode the signal.

This work aims to identify the discriminative features and a classification model that improves decoding performance on different IS tasks and is also robust to noise. Based on experimental results, we propose Tangent Space (TS) [5] as input features to an Artificial Feed Forward Neural Networks (ANN) [6] model. Our proposed approach improves the mean classification accuracy from 49.3% to 60.35%, 49.2% to 58.61%, 66.56% to 69.43%, and 73.27% to 78.51% on three short words, three vowels, two long words and one long vs. one short word classification tasks respectively.

We also propose two designs for computer control using IS brain signals and provide implementation details. We tested one design in a partial online setting and obtained an Information Transfer Rate of 21 bits/minute.

This paper is organized as follows. Section 2 presents the problem statement and then showcases our research contribution as well as presents an overview of existing methods for decoding imagined speech. Section 3 describes the proposed approach for feature extraction and classification. Section 4 shows the results from different feature extraction and classification models. Section 5 describes the user interface designs and pipelined system for real-time BCI system using IS for computer control. Section 6 provides discussion and

Abhiram Singh is a Ph.D student in the Department of Computer Science and Engineering, Indian Institute of Technology Bombay, India, 400076. e-mail: abhiram25.1990@gmail.com

Ashwin Gumaste is a professor in the Department of Computer Science and Engineering, Indian Institute of Technology Bombay, India, 400076. e-mail: ashwin@ashwin.name

conclusion.

II. Problem Statement, Research Contribution, And Related Work

We now formally describe our problem statement along with the high-level contributions of our work.

A. Problem Statement

Given an EEG signal, we desire to identify whether it belongs to an imagined speech category. If so, then we desire to decode the actual, imagined word or word category. Subsequently, we want to use this decoded information to take appropriate action for computer interaction.

B. Contribution

Our work leads to the following contributions:

1) We show that the proposed approach is capable of discriminating IS EEG signals from participant's rest state EEG signals. This provides the first step in decoding IS signals.
2) We consider the aspect of generalization of neural networks (NN) on IS signals. We identify the Covariance matrix as the most useful discriminative feature. Tangent Space (TS) [5] as discriminative information preservative transformation of the covariance matrix to vectors. PCA as dimension reduction technique, feed-forward Neural Network (NN) as the most successful classification model with boot-strap aggregation (bagging) scheme for combining results of multiple NN outputs. Results confirm that our proposed approach indeed improved the classifier performance significantly in comparison to existing approaches.
3) We developed an FSM to operate the computer system using only IS signals. For user interaction with the system, a new and simple Graphical User Interface (GUI) was developed corresponding to this FSM. To the best of our knowledge, this is the first-ever approach for general-purpose computer control, which is based only on the IS signals. We discuss two designs of FSM for binary classification tasks using IS signals and then focus on several improvements to build a fully functional system that can work in a real-time (online) setting. We simulated a design using available datasets for demonstration and obtained an ITR of 21-bits-per minute.

C. Related work

Nguyen et al. [7] used 64 channel EEG to capture three vowels, two long words, and three short words across different subjects. They used features from Riemann manifold (Tangent Space (TS)) [5] as an input to the Relevance Vector Machine (RVM) [8]. Their results show the mean classification accuracy of 49% for vowels and also for short words, 66% for long words across and 73% for long and short word classification tasks for different subjects. We significantly improve these results for vowels and short words classification tasks and obtains equivalent results for long words and short and long word classification tasks. This improvement was achieved by reducing the dimensions of the transformed covariance matrix using the PCA and a more powerful NN classifier combined with bagging as an ensemble classifier. Authors also suggest using Extreme Learning Machine (ELM) as a classifier, but they obtained very similar results as of using RVM as a classifier.

Tomioka et al. [9] applied Common Spatial Pattern (CSP) to data, calculated log-variance of each transformed channel to create input features, and used linear discriminant analysis (LDA) as a classifier. Here the main limitation is posed by the LDA classifier, which works well if the features of each class are generated using the normal distribution. Our approach removes this limitation by using a powerful ANN classifier for modeling complex distributions.

Dasalla et al. [10] used CSP based transformation and support vector machine (SVM) as a classifier. In particular, they used four CSP channels for transforming raw EEG signals using the training data. Then transformed both training and test data using learned parameters. Signals obtained after transformation were then stacked together to form a vector and finally given as input feature to the SVM classifier. Authors used CSP, which is known to work best for the case of motor imaginary signals. On the other hand, we used the covariance matrix as input features to our model, which captures the dependence between different channels and is able to retain some information caused by imagined speech production.

Min et al. [11] used statistical features such as mean, variance, standard deviation, and skewness and ELM as a classifier. To extract these features, they divided the signals into overlapping windows and calculated these features over each channel of each window to form a feature vector, used a sparse-regression based feature selection scheme to reduce the dimension of the features and used ELM as a classifier. In our proposed approach, we used PCA for feature dimension reduction as this method is much faster than a sparse regression-based feature dimension approach. Another difference lies in the use of the classification model. Our feed-forward neural network model is trained using gradient descent, and gradients are computed using a backpropagation algorithm (rather than random initialization of weights in the first layer of ELM). Gradient descent makes our model more powerful and shows good generalization on test data.

III. Approach

Our proposed method for IS signal decoding is summarized in the following steps. First, we create covariance matrices from the raw EEG trials. Then we project each of these covariance matrices to the tangent space (TS) to get a vector representation of the matrices. Third, we reduce the dimension of these vectors using PCA. Finally, features in the lower dimension are given as input to the ensemble of NN classifiers, and results of all classifiers are averaged to get the final prediction of the model.

A. Background

Before a detailed explanation of our approach, we briefly go through the concepts.

1) Covariance Matrix: Given an EEG trial $E \in \mathbb{R}^{n,m}$, where n is number of EEG channel and m is number of samples, covariance matrix $C \in \mathbb{R}^{n,n}$ and more specifically $C \in \mathbb{R}^{+}_{sym}$ can be represented as follows:

$$C = \frac{1}{m} E * E^T \tag{1}$$

Where, $\mathbb{R}^{+}_{sym}$ represents the space of real symmetric positive semi-definite matrices, T represents matrix transpose operation, $*$ is matrix multiplication operator and division operation scales each value of matrix by a fixed number m. In matrix form, E is represented as two dimension array of shape $[n, m]$ and C of shape $[n, n]$.

2) Tangent Space: Spatial dependency between EEG channels can be measured by estimating covariance matrix for a given EEG trial but these matrices must be transformed to form a vector for creating the low dimensional input features representation. However this transformation must preserve the discriminative information of the target class. One known way to achieve this (in an unsupervised manner) is to project covariance matrices on to the tangent space (TS) [5]. This projection is defined as follows,

$$\begin{gathered} C_i = C_m^{1/2} logm(C_m^{-1/2} C_i C_m^{-1/2}) C_m^{1/2} \\ logm(M) = V D' V^{-1},\ D'[i,i] = log(D[i,i]) \end{gathered} \tag{2}$$

Where C_i is the covariance matrix, whose projection is needed, C_m is the mean of the covariance matrices, M is a diagonalizable matrix, M^{-1} is the usual matrix inverse operation, and VDV^{-1} represents diagonalized form of the matrix M. M^{-1} and $M^{1/2}$ can be calculated in a similar way to that of $logm(M)$, where we only need to replace the log function with inverse and square root functions.

The projected matrix (in the tangent space) is then flattened to form a vector. These steps transform covariance matrices into the Euclidean space while preserving the information present in the space of covariance matrices.

3) Principal Component Analysis: In our work, we used PCA [12] for dimension reduction. The objective function of PCA is,

$$\begin{gathered} max_{u \in \mathbb{R}^n}\ u^T A u \\ subject\ to\ \|u\|_2^2 = 1 \end{gathered} \tag{3}$$

Here, A is the covariance matrix obtained as $\frac{1}{m}\sum_{i=1}^{m} x^{(i)} {x^{(i)}}^T$ and vectors $x^{(i)}, u \in \mathbb{R}^n$. The solution of this optimization problem can be obtained by solving the Eigen value problem $Au = \omega u$.

4) Artificial Neural Network: We used Artificial Neural Network (ANN) [6] model as a feed-forward model for classification task. The ANN model is loosely inspired from the human brain. It linearly combines the input and then applies non-linearity (both steps applied in a layered fashion) to generate a desired output. Connectivity between two layers is defined as follows:

$$a^l = g^l(W^l * a^{l-1}) \tag{4}$$

Here, vector a^{l-1} represents an n-dimensional input obtained from layer $l-1$, a^l represents an m-dimensional output at layer l, W^l is the weight matrix of shape $[m, n]$ between layer $l-1$, l and g^l is the non-linear activation function at the layer l. At a hidden layer, non-linear function g is either sigmoid, tanh or ReLU function applied elementwise. At the output layer, the softmax function at g is applied to compute target class probabilities.

5) Bootstrap Aggregation: We also used the Bootstrap Aggregation (Bagging) Classifier [13]. This classification method creates several base classifiers and trains each on subsets of the original dataset. Elements of each subset are chosen with replacement from the original dataset. Results are combined by averaging the results of all classifiers. Bagging has shown to increase classifier performance in terms of improved classification accuracy and reduced variance. In our case, based classifiers are ANN.

B. Proposed Approach

In this section, we describe our proposed approach in detail.

1) Feature Extraction: The following steps give details about feature extraction from raw EEG signals.

- We store raw EEG trials in the format $[n, c, s]$ where n is the number of trials, c is the number of channels, and s is the number of samples.
- Then we divide data into training and test set which is stored in the form $[n_{tr}, c, s]$ and $[n_{te}, c, s]$ where n_{tr} and n_{te} represents number of trials in the training and test set respectively.
- For each trial in train and test set, covariance matrices are calculated as described in equation 1 and stored in the form $[n_{tr}, c, c]$ and $[n_{te}, c, c]$.
- Training data in the above step is used to find the mean of covariance matrices represented as C_m in equation 2.
- Each trial of training and testing data is then projected to tangent space as defined in equation 2. Then projected matrices are converted to vector representation by concatenating rows of the matrix to form the matrices of dimension $[n_{tr}, n_f]$ and $[n_{te}, n_f]$ where n_f denotes the number of features.
- Feature dimension is then reduced by PCA as defined in equation 3. Training data $[n_{tr}, n_f]$ is used for learning vectors u. Then dimension of training and testing data is reduced using the learned vectors u to form the matrices of dimension $[n_{tr}, n_{rf}]$ and $[n_{te}, n_{rf}]$ where n_{rf} denotes the number of features after dimension reduction (rf stands for reduced features).

2) Feature Classification: Following steps give details about feature classification.

- ANN takes $[n_b, n_{rf}]$ dimensional matrix as input and generates $[n_b, n_o]$ dimensional matrix as output by repetitively applying equation 4 in layered fashion. Here, n_b denotes the batch size and n_o denotes the number of output classes. ANN intermediate values are represented by $[n_{nb}, n_f^l]$ where, n_f^l denotes the number of features at layer l. Cross entropy loss is computed through the ANN output matrix O_{pred} of dimension $[n_b, n_o]$ and

true output represented by one hot matrix O_{true} of dimension $[n_b, n_o]$. Each row of output matrix O_{pred} generates target class probability and sums to 1. Each row of one hot matrix O_{true} has all zeros but one for the class in which input belongs. The required gradients were calculated with respect to cross entropy loss, ANN weights are updated using the gradient descent variant Adam optimizer [14] and gradients of each layer are computed using the backpropagation algorithm.

- We use a bagging classifier with k number of ANNs as its base classifier. Each ANN weights are randomly initialized and trained in parallel with the input batch size $n_b = n_{tr}$. The output matrix of each ANN is averaged along each row to give the final output matrix of the bagging classifier.

The computation of obtaining covariance matrices from raw EEG signals and covariance matrix transformation to vectors was performed using the Pyriemann library [15]. PCA implementation of sklearn [16] is used to project vectors into lower-dimensional space. ANN and bagging classifier training is also performed using sklearn library.

IV. Results

In this section, we provide details of the experiment, dataset description, and results obtained with our proposed approach and approaches proposed in the literature.

A. Experiment and dataset details

Nguyen et al. [7] experimented with IS-related brain signals using an EEG device. They divided experiment into four tasks namely short words $\{in, out, up\}$, vowels $\{a, i, u\}$, long words $\{independent, cooperate\}$ and short and long words $\{in, cooperate\}$.

In each experiment, subjects focus on a computer screen to receive the visual cue about the word to be imagined along with periodic beep indicating the start of imagination. Each trial consists of 7 periods of T seconds. Starting 4 periods consists of the visual cue with an audio to imagine the word while the last three periods include only visual cue for imagined speech. The trial ended with 2 seconds of rest state condition without any beep sound or any visual cue. For vowels and short words, T was 1 second, and for long words and long and short words tasks, T was 1.4 second. Each task has 100 trials for the target class for each subject.

EEG signals were captured using 64 electrodes and down-sampled to 256 Hz. Out of 64 channels, 60 channels were used for recording EEG signals of IS tasks. The dataset contains nine subjects for vowels IS task, six subjects for short words IS task, six subjects for long words IS task, and seven subjects for long and short words IS task. Each subject is having 100 trials for every target class except for two subjects in the short and long words task where two subjects are having 80 trials each. We rejected data of one subject from short words IS task, three subjects from vowels IS task, one subject from long words IS task, and one subject from long and short words IS task due to mismatch between the number of channels in the subject's data.

Within each trial, subjects performed three repetitive thinking processes under the imagined speech condition. Hence each trial gave rise to three different $[c, t]$ dimensional matrices with $c = 60,\ t = 256$ for vowels and short words and $c = 60,\ t = 360$ for long words and short and long words tasks. Here for each subject, we have $[900, 60, 256]$ or $[600, 60, 360]$ dimensional matrix as input (except for two subjects in short and long words task where dimension is $[480, 60, 360]$) and 3 or 2-dimensional one hot vector as target labels depending on the 3 vowels/short words category or 2 long words/short and long word category.

B. Results

In this section, we first report results based on our proposed approach and then compare it with existing approaches for decoding the IS task.

1) Performance metric: We used classification accuracy (CA) to check model performance. CA measures the number of predicted outputs equal to actual outputs divided by the number of predictions. This quantity lies between 0 and 1. $(1 - CA)$ denotes the misclassification rate of the model. We report results using 10-fold cross-validation. Before train and test set separation, the dataset was randomly shuffled. Then in each fold, data were divided into train and test set using stratified sampling. This sampling method maintains the sample proportion of data points of each class in the train and test sets.

2) Results: Now we first show results for the classification of IS signals from rest state signals. Using experimental results, we show that our proposed approach is able to separate IS signals from brain rest state signals with very high accuracy. We show these results on the classification task of long words, short words, and vowels in Figure 1. For comparison, we extracted IS signals of the 5th period from the dataset and 2 seconds of rest state brain signals.

We also performed a significance test of our proposed approach with chance level classification accuracy. We report p-values using 2 tailed t-test in Table I. Small p-values show that results obtained using our proposed approach is significantly different from the chance level classification accuracy. In table II, we report the mean classification accuracy and standard deviation of all the subjects calculated for each classification task.

TABLE I: P-Values For Proposed Approach and Chance Level Accuracy.

t-test	p-values
Vowels	0.0005
Short words	0.0083
Long words	0.001

TABLE II: Classification Accuracy Across Subjects on Different IS Classification Tasks and Rest State Brain Signals.

IS task vs rest state	Mean accuracy	Standard deviation
Vowels	0.8033	0.0858
Short words	0.794	0.1355
Long words	0.858	0.0927

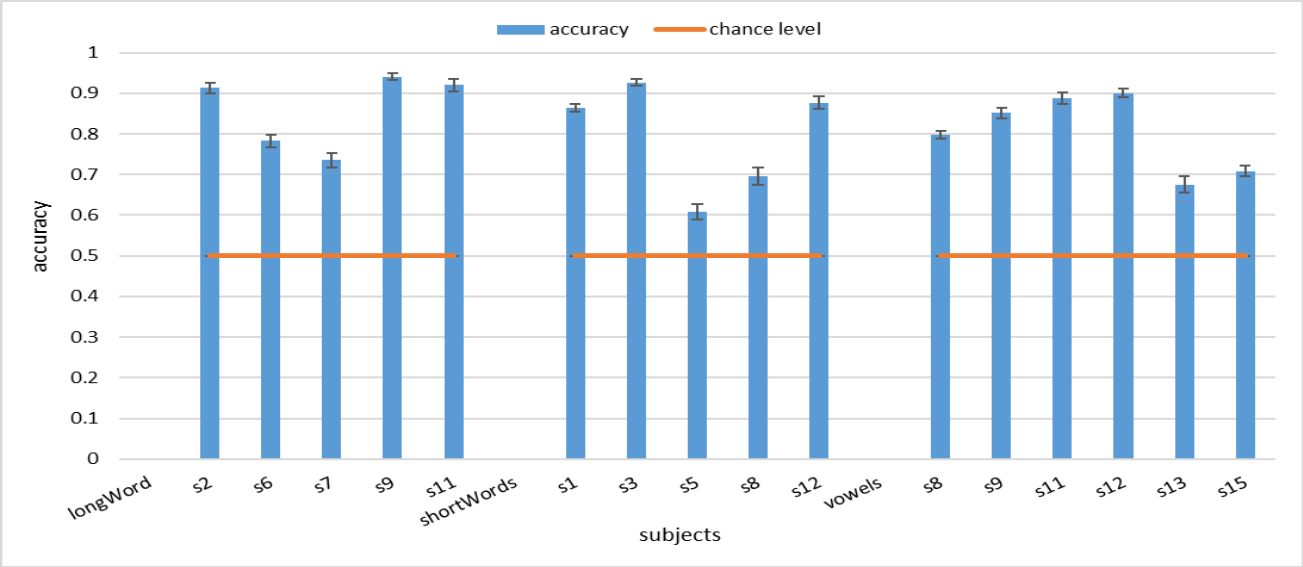


Fig. 1: Classification accuracy on long words imagined speech, short words and vowels vs rest state brain signals. Each column represent mean classification accuracy and error bars show standard error of mean. Name of each task is followed by the participants id.

TABLE III: Mean, Standard Deviation (Std), Standard Error Of Mean (Sem), Maximum (Max) And Minimum (Min) Classification Accuracy for all Subjects on Different IS Tasks.

Long words vs rest	MEAN	STD	SEM	MAX	MIN
s2	0.9125	0.0406	0.0135	0.95	0.825
s6	0.7825	0.0461	0.0153	0.875	0.7
s7	0.735	0.0538	0.0179	0.825	0.675
s9	0.94	0.0254	0.0084	0.975	0.875
s11	0.92	0.0471	0.0157	0.975	0.8
Short Words vs rest	MEAN	STD	SEM	MAX	MIN
s1	0.8633	0.0296	0.0098	0.9166	0.8166
s3	0.9266	0.0249	0.0083	0.95	0.8666
s5	0.6083	0.0597	0.0199	0.7166	0.55
s8	0.695	0.0628	0.0209	0.7666	0.5333
s12	0.8766	0.0454	0.0151	0.95	0.8166
Vowels vs rest	MEAN	STD	SEM	MAX	MIN
s8	0.7983	0.0292	0.0097	0.85	0.75
s9	0.8516	0.039	0.013	0.9166	0.8
s11	0.8866	0.042	0.014	0.9666	0.8
s12	0.9	0.0324	0.0108	0.9666	0.8666
s13	0.675	0.0606	0.0202	0.7666	0.55
s15	0.7083	0.0389	0.0129	0.7666	0.6666

The high accuracy of many subjects on three different tasks shows that our proposed approach is able to successfully differentiate IS signals from rest state brain signals. Table III shows the mean classification accuracy, the standard deviation of the mean values, standard error of the mean, the maximum and minimum value of each subject in different tasks. This gives us some idea of model performance on different non-overlapping subsets of data. Note that the minimum classification accuracy of all subjects is well above chance level for long words classification tasks in comparison to short words and vowels. This suggests that long words carry a lot more information that short words which are used by the model to differentiate from rest state brain signals.

Now we report results using our proposed approach on different IS tasks. Figure 2 shows the mean classification accuracy of the proposed approach for different subjects. We report maximum mean classification accuracy of 0.85 for subject s9 on short and long words IS classification task and 0.5378 for subject s12 on vowels IS classification task. Note that classification accuracy is above the chance level for each subject. High performance on short and long word classification tasks across all subjects states that long word imagination leads to the EEG patterns that are very different from short words imagination. In table IV, we report mean classification accuracy and standard deviation obtained on each IS task. Due to different complexity of words in short and long words IS task, the highest mean classification accuracy is obtained in this task.

TABLE IV: Mean Classification Accuracy And Standard Deviation on four Different IS Tasks using our Proposed Approach of ts+ann.

IS task	Mean accuracy	Standard deviation
vowels	0.586083	0.038881
shortWords	0.6035	0.044183
shortLong	0.785117	0.049689
longWords	0.6943	0.055531

Now we compare our proposed approach of using TS+ANN with existing approaches on different IS tasks. Figure 3 shows the performance of our proposed approach with existing approaches. We compare our approach of using ts+ann with: (a) ts as features with rvm as a classifier approach suggested by Nguyen et al. [7], (b) ts as features and elm as a classifier approach also suggested by Nguyen et al. [7], (c) use of statistical features with elm as a classifier suggested by min et al. [11], (d) CSP based transformed signal with SVM as a classifier approach of Dasalla et al. [10] and (e) variance of CSP transformed signal with LDA as a classifier suggested by

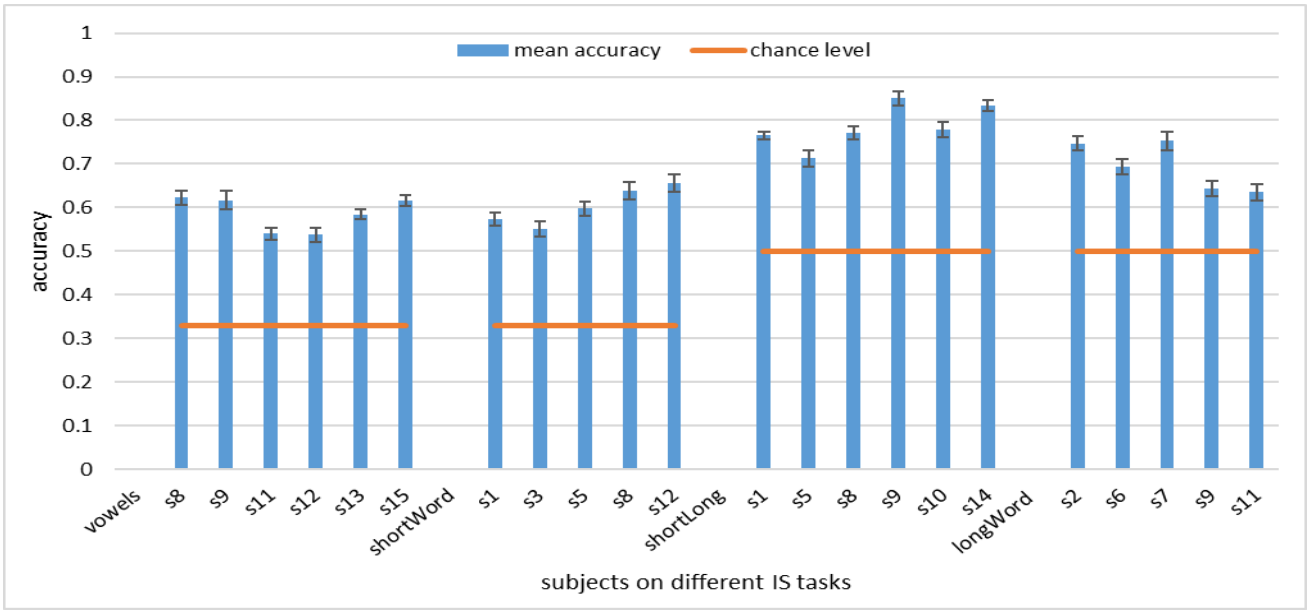


Fig. 2: Classification accuracy of our proposed approach (ts+ann) on four different IS tasks. Error bars represent standard error of mean. Name of each task is followed by the participants id. Chance level accuracy is same for vowels and short words task and for short-Long and long words tasks.

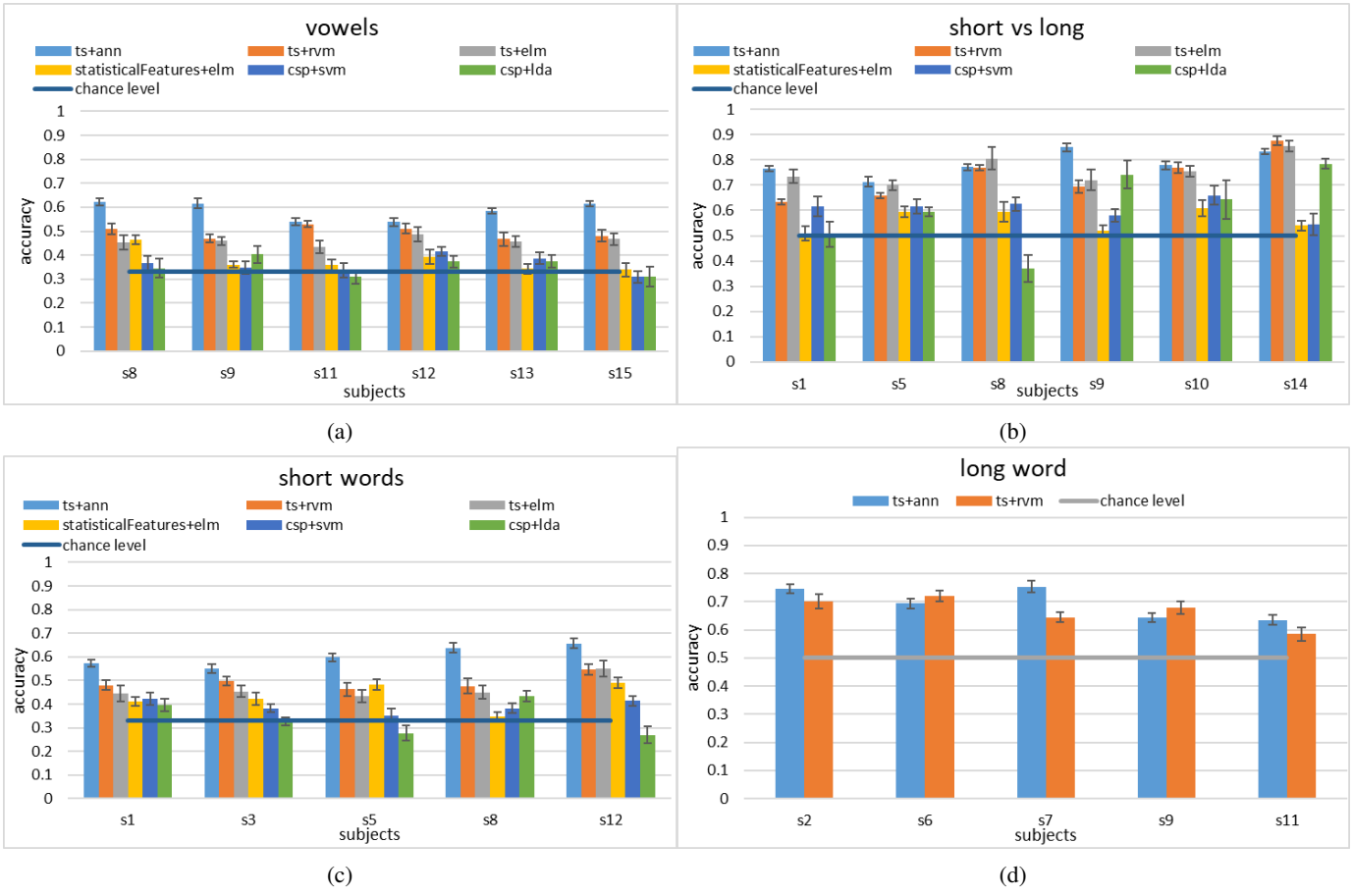


Fig. 3: Classification accuracy of different approaches on vowels, short vs long, short and long words classification tasks. Error bars show standard error of mean.

Tomioka et al. [9] on three different IS tasks of vowels, short words and short vs long word. Due to the unavailability of

TABLE V: Detailed Comparison with Different Approaches (as Reported in the Work of Nguyen et al. [7]). Result are in the format: Mean Accuracy + Standard Deviation, Minimum Accuracy - Maximum Accuracy

Vowels IS task						
Subjects	s8	s9	s11	s12	s13	s15
csp+lda [9]	34.6 + 11.8 16.6 - 60	40.3 + 10.4 13.3 - 46.6	31.0 + 8.5 20.0 - 46.6	37.3 + 7.1 30.0 - 53.3	37.5 + 8.1 26.6 - 53.3	31.0 + 12.7 10.0 - 50.0
csp+svm [10]	36.7 + 9.2 30.0 - 60	34.7 + 7.7 30.3 - 53.3	33.7 + 8.7 23.3 - 53.3	41.7 + 5.7 36.7 - 56.7	38.7 + 7.6 30.3 - 56.7	31.0 + 7.4 23.3 - 46.7
statF+elm [11]	46.5 + 5.6 36.7 - 56.7	36.1 + 4.4 26.6 - 43.3	36.0 + 7.1 30.0 - 50.0	39.3 + 9.4 30.0 - 60.0	34.3 + 6.5 23.3 - 43.3	34.0 + 8.3 23.3 - 46.7
ts+elm [7]	45.3 + 8.9 30.0 - 56.7	46.0 + 5.1 36.7 - 53.3	43.3 + 7.9 33.3 - 53.3	48.6 + 8.9 36.7 - 60.0	45.7 + 7.2 36.7 - 63.3	46.7 + 7.5 36.7 - 60.0
ts+rvm [7]	51.0 + 6.7 43.3 - 63.3	47.0 + 5.5 36.7 - 53.3	53.0 + 4.0 46.7 - 60	51.0 + 6.3 43.3 - 63.3	46.7 + 8.2 33.3 - 60.0	48.0 + 7.2 33.3 - 56.7
ts+ann (proposed)	**62.0** + 4.68 53.33 - 68.8	**61.66** + 6.46 54.44 - 72.22	**54.0** + 4.16 45.55 - 61.11	**53.78** + 4.84 43.33 - 58.88	**58.44** + 3.52 53.33 - 64.44	**61.55** + 3.69 55.55 - 66.66
Short words IS task						
Subjects	S1	S3	S5	S8	S12	
csp+lda [9]	39.6 + 7.6 26.6 - 53.3	32.6 + 4.9 26.6 - 43.3	27.7 + 9.8 20.0 - 50	43.3 + 7.0 36.6 - 53.3	27 + 10.8 13.3 - 43.3	
csp+svm [10]	42.3 + 8.2 33.3 - 56.7	38.3 + 5.3 33.3 - 50.0	35.3 + 8.3 30.0 - 56.7	38.3 + 6.1 33.3 - 53.3	41.33 + 6.7 33.3 - 53.3	
statF+elm [11]	41.0 + 5.5 46.7 - 56.7	42.3 + 8.0 26.7 - 56.7	48.3 + 7.2 36.7 - 60.0	34.7 + 5.9 26.7 - 46.7	49.0 + 6.7 36.7 - 56.7	
ts+elm [7]	44.6 + 10.3 33.3 - 60.0	45.3 + 7.4 33.3 - 56.7	43.4 + 7.7 30.0 - 56.7	45.0 + 8.5 30.0 - 56.7	55.0 + 9.8 40.0 - 70.0	
ts+rvm [7]	48.0 + 6.1 40.0 - 56.7	49.7 + 5.5 40.3 - 56.7	46.3 + 8.2 36.7 - 66.7	47.7 + 9.8 36.7 - 66.7	54.7 + 6.9 43.3 - 66.7	
ts+ann (proposed)	**57.44** + 4.55 48.88 - 64.44	**55.0** + 5.28 43.33 - 61.11	**59.77** + 4.91 54.44 - 72.22	**63.88** + 6.25 53.33 - 75.55	**65.66** + 5.99 57.77 - 76.66	
Short vs long words IS task						
Subjects	S1	S5	S8	S9	s10	s14
csp+lda [9]	50.5 + 14.8 30.0 - 72.5	59.5 + 5.7 52.5 - 70.0	36.9 + 15.9 21.9 - 71.9	74.1 + 16.6 31.3 - 87.5	64.3 + 23.0 20.0 - 80.0	78.5 + 6.3 70.0 - 90.0
csp+svm [10]	61.5 + 12.0 50.0 - 85.0	61.5 + 8.8 50.0 - 80.0	62.5 + 8.3 50.0 - 81.3	58.1 + 7.2 50.0 - 75.0	66.0 + 11.5 50.0 - 85.0	54.5 + 13.2 45.0 - 90.0
statF+elm [11]	51.0 + 8.4 40.0 - 65.0	59.5 + 6.4 50.0 - 70.0	59.4 + 11.5 43.8 - 81.3	51.9 + 6.6 43.8 - 68.8	61.0 + 9.7 45.0 - 75.0	54.0 + 6.1 50.0 - 70.0
ts+elm [7]	73.5 + 8.2 60.0 - 85.0	70.0 + 6.2 60.0 - 80.0	**80.6** + 13.2 62.5 - 93.8	72.5 + 12.2 43.7 - 87.5	75.5 + 6.8 65.0 - 85.0	85.5 + 6.8 75.0 - 95.0
ts+rvm [7]	63.3 + 2.9 60.0 - 70.0	65.8 + 3.1 62.5 - 70.0	76.9 + 3.0 71.8 - 81.3	69.4 + 7.5 59.4 - 81.3	76.8 + 6.2 67.5 - 85.0	**87.5** + 5.5 75.0 - 92.5
ts+ann (proposed)	**76.5** + 2.83 73.33 - 81.66	**71.33** + 5.66 60.0 - 78.33	77.08 + 4.26 68.75 - 85.41	**85.0** + 4.73 77.08 - 91.66	**77.83** + 5.16 68.33 - 85.0	83.33 + 3.57 78.33 - 90
Long words IS task						
Subjects	S2	S6	S7	S9	s11	
ts+rvm [7]	70.0 + 7.8 55.0 - 80.0	**72.0** + 0.6 65.0 - 85.0	64.5 + 5.5 59.0 - 75.0	**67.8** + 6.8 55.0 - 80.0	58.5 + 7.4 50.0 - 77.5	
ts+ann (proposed)	**74.66** + 4.76 66.66 - 81.66	69.33 + 5.53 60.0 - 76.66	**75.33** + 6.35 65.0 - 86.66	64.33 + 5.12 56.66 - 71.66	**63.5** + 5.39 53.33 - 73.33	

results on other approaches for long words classification task, we compare the results of our proposed approach with one approach ts+rvm approach suggested by Nguyen et al. [7].

As reported in Figure 3, our approach outperforms existing approaches on decoding vowels and short words IS tasks and performs equivalent to the ts+elm approach on decoding short vs. long words IS task and ts+rvm approach on decoding long word IS task. We report the highest classification accuracy of 0.8333 on short vs. long words IS task for subject s14 and minimum classification accuracy of 0.54 for subject s11 on vowels IS task. Results obtained using our approach of ts+ann is well above chance level for all subjects on all four IS tasks. Chance level accuracy is 0.33 for vowels, and short words and 0.5 for short vs. long words and long words IS tasks. Due to dimension reduction with the help of PCA and the generalization capability of the ANN model, our proposed approach is able to outperform other approaches with a significant margin on vowels, and short words IS tasks. For short vs. long words and long words classification tasks, our approach performs equivalent to the approach proposed by Nguyen et al. [7] but the standard error of the mean (SEM) is still lower in our approach. Bagging classifier helps in reducing the variance in predicting the output, and hence, results are more stable in terms of SEM in comparison to the other approaches.

Now we compare mean classification accuracy, standard deviation, maximum, and minimum accuracy obtained using our approach with existing approaches in Table V. For the long words IS task, we compared our approach with one approach proposed by Nguyen et al. citeNguyen2018IS. Authors have not published results of other approaches for long words IS task.

As we can see from Table V, our proposed approach is able to outperform other approaches in short words and vowel

TABLE VI: P-Values Obtained After Two Tailed Paired T-Test.

t-test	ts+ann, chance level	ts+ann, ts+rvm	ts+ann, ts+elm	ts+ann, statF+elm	ts+ann, csp+svm	ts+ann, csp+lda
vowels	0.000016673	0.0117196	0.000780616	0.000246665	0.00038811	0.0000936821
short words	0.000157915	0.00385681	0.001383572	0.005317578	0.000641303	0.002537817
Short vs long	0.0000327993	0.159018237	0.372650079	0.001139683	0.002841504	0.019606281
long words	0.001440737	0.34204982	-	-	-	-

classification tasks and performs equivalent with the approach suggested by Nguyen et al. [7] on short vs. long words and long words classification task. Our proposed approach of using ts+ann also has less deviation in comparison to the other approaches. This is achieved by using an ensemble of ANN classifiers and averaging the results of each.

Now we perform significance testing of our proposed approach with chance level accuracy and other approaches. Table VI shows the p-values after performing the two-tailed pairwise t-test. Results in Table VI show very low p-value when comparing our proposed approach ts+ann with chance level accuracy. Hence, our approach performs well above the chance level on all four IS tasks.

In comparison to the approach proposed by Nguyen et al. [7], results obtained using our approach are significantly different for vowels, and short words IS tasks. This is verified by the low p-values of 0.01171 and 0.00385 for a 0.05 significance level. On the other hand, for short vs. long words and long words IS tasks, p-values 0.15901 and 0.34204 shows the equivalence of results between our proposed approach (ts+ann) and Nguyen et al. [7] (ts+rvm). Similar behavior is also observed for the ts+elm approach also suggested by Nguyen et al. [7].

For all other approaches, we see that p-values are far below the significance level. Hence, it shows that results obtained with our approach (ts+ann) are significantly different from approaches suggested by min et al. [11] (statistical features with elm as a classifier), Dasalla et al. [10] (CSP based transformed signal with SVM as a classifier) and Tomioka et al. [9] (variance of CSP transformed signal with LDA as a classifier) on three different IS tasks of vowels, short words and short vs. long word.

From Table V, we see that the performance of each method varies significantly across subjects. To compare different approaches, we require to have a result from each of the considered approaches. To this end, we need to average out the performance of each approach across all subjects. This will give us one performance measure for each classification task. Table VII summarizes these results.

From Table VII, it is clear that our proposed approach gives the highest accuracy across all the IS classification tasks. By examining the standard deviation of our approach, it is clear that the ANN model does not show much variability across subjects. Other approaches show either low mean accuracy with low variance over all the subjects or high accuracy with high variance across the subjects. Hence existing approaches are either unable to extract useful discriminative information thereby resulting in low accuracy and low deviation or these approaches are able to decode IS signals of some subjects and therefore achieving high accuracy however with high variance. The approach with high mean accuracy and low variance (when calculated across all subjects) is desired.

TABLE VII: Mean Classification Accuracy (Mean) And Standard Deviation (Std) Computed Across all Subjects for Each IS Task.

Task		Short words	Vowels	Short vs Long	Long words
csp+lda	Mean	34.04	35	64.83	-
	Std	7.21	3.91	10.12	
csp+svm	Mean	39.1	35.5	61	-
	Std	2.77	3.8	4.64	
statF+elm	Mean	43.06	37.5	56.16	-
	Std	5.86	4.71	4.75	
ts+elm	Mean	46.66	45.5	75	-
	Std	4.71	1.81	5.25	
ts+rvm	Mean	50.1	49.0	73.3	66.2
	Std	3.5	2.4	8.86	4.8
ts+ann	Mean	**60.16**	**57.83**	**79.3**	**69.43**
	Std	4.32	3.49	5.08	5.553

This shows the generalization capability of ANN models over other models when given the same input data. One thing to note here is that classification accuracy of words in the long words IS task and short vs. long words IS task is much higher than vowels, and short words IS tasks. This suggests that the proper choice of words based on word length and complexity provides useful discriminative information and improves the recognition power of the models.

V. Imagined speech for computer interaction

This section provides details for computer interaction using Imagined Speech signals captured from an EEG device. We propose two designs, one that is based on creating a new Graphics User Interface (GUI) to click anywhere on a computer screen and a second design that uses the functionality of Arrow, Enter, and Backspace keys of a computer keyboard to perform various actions.

A. Design

To control a computer, the first requirement is to locate desired content displayed on a computer screen. So there must be some provision with which a user is able to reach the target location. Currently, this step is done by the movement of a mouse that is shown on the screen as a change in the cursor position. A keyboard may also be used by using the Tab or arrow keys to reach the target. Since cursor control requires continuous input from the user, and imagined speech classifier output is discrete, hence it does not make sense to control continuous movement using discrete steps. Hence the design has to be done by considering the type of classifier output to improve the system performance.

Assumption: The binary classifier is used for imagined speech decoding. This assumption is due to simplicity in GUI

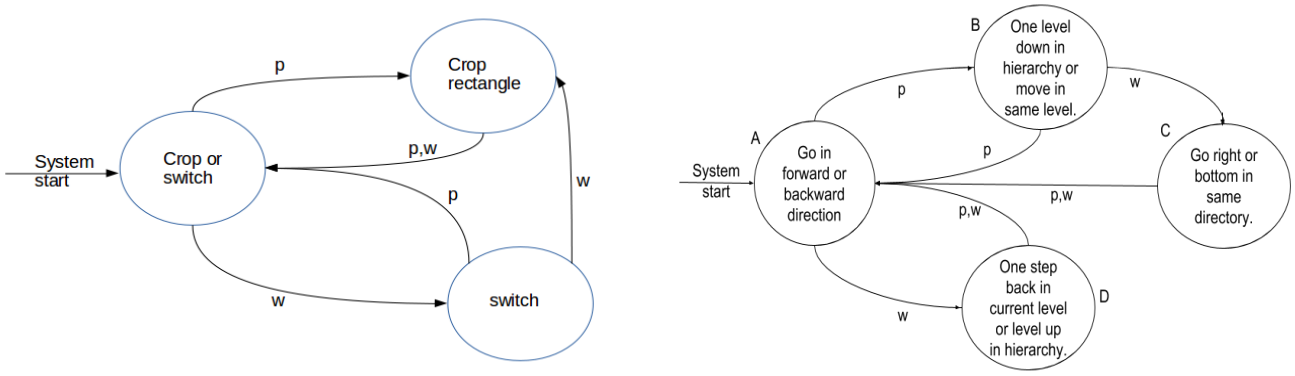


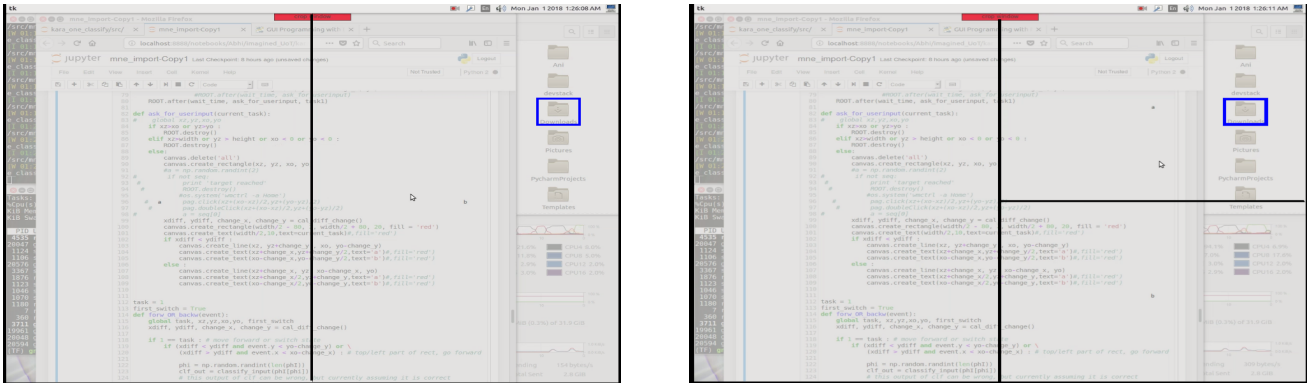
Fig. 4: (a) State diagram 1 of computer control application. (b) State diagram 2 of computer control application. p stands for phoneme and w for word.

demonstration and high classification accuracy obtained by the binary classifier for the datasets. Also, the classifier is trained such that the 0 output corresponds to a phoneme, and the one output corresponds to a word. These assumptions can be relaxed, and several extensions can be proposed, as shown in the later sections.

Navigation steps: The following steps are performed in each iteration to open a folder currently being displayed on the screen. These steps are also shown in the FSM of Figure 4a.

We obtain the screen resolution and create a rectangular window with partial transparency of the same size as that of the screen resolution. 2) We then divide the current rectangle into two halves. If the length is greater or equal to the breadth, then we further divide rectangle along the length, otherwise divide the rectangle along breadth. 3) We then display one phoneme on one half of the rectangle and one word on the other side of the rectangle. For consistency, if the rectangle is divided along its length, then the phoneme is always displayed on the left part of the rectangle, and the word is always displayed on the right part of the rectangle. Similarly, if the rectangle is divided along its breath, then the phoneme is displayed on the top part, and the word is displayed on the lower part of the rectangle. The phoneme and word are chosen randomly from their respective sets. 4) A display response is used for ensuring that the user starts thinking of either the phoneme or word in a given time-interval leading to the capture of the corresponding brain signals. The user thinks either phoneme or words by looking at the part of the rectangle under which the target folder is located. 5) The captured signal is pre-processed, features are extracted and given to the classifier to decode the user imagined word. If a classifier generates an output of 0 then, the rectangle part (either right or bottom) representing word is removed, whereas if the output is a one then, the rectangle part (either left or top) corresponding to a phoneme is removed. 6) Repeat until the rectangle becomes small enough to cover the folder fully. At this stage, the user needs to switch the window and double click on the folder. However, until this stage, the system only recognizes one action, which is to crop the current window to reach a target location. To solve this problem, at the starting of each step, two options are displayed to the user, either to go to the crop state or switch state. The crop option is selected by thinking of a phoneme, and this leads to the system state where all the above-defined steps can be performed. The switch option is selected by thinking a word, and it switches the window and double clicks the folder behind the current rectangle. 9) If at a switch state, the user at any time instance thinks of any phoneme then, the system state is reset, the rectangle is set to the full-screen resolution, and the whole process restarts to select a different folder. However, if the classifier made a mistake on the previous crop state then, the user can go to a switch state, think of a word, and the system recreates the rectangle of the last crop rectangle state and goes to the crop rectangle state again.

Second design (shown by FSM in Figure 4b) converts user imagined speech into keyboard actions. Here we demonstrate one application to utilize a tree file directory structure. The tree structure can be divided into multiple levels with a root at the top and leaves at the bottom. Files here represent a leaf of the tree, and the root is the root directory of the computer system. Initially, a user decides to open a particular file in the computer system. Then computer control is shifted to the root of a directory tree. There might be multiple directories at the root. So the first among them is selected. Now, based on the target file location, the user either can navigate at the same level of the tree or go a level down. To achieve this, a user thinks about one phoneme to change the system state from A to B (see Figure 4b). In state B, the user can either move in the same directory level or go one level down. A user can think of a phoneme to go one level down along the directory tree hierarchy. This user imagined speech is converted into action corresponding to pressing an Enter key. In another case, a user can think of a word to switch state from B to C and navigate in the same directory level by thinking of a phoneme for the right arrow key action and a word for the bottom arrow key action. Then the system goes to state A. It is possible that the classifier has made a mistake or that the user wants to go up the directory tree. Hence, in state A, the user thinks of a word to change state from A to D and thinks of a phoneme to revert to the previous action or a word to go a level up in the tree. In



(a) (b)

Fig. 5: (a) First division of rectangle from crop action. To open Downloads folder (in blue box), user thinks one word so that right part of rectangle is selected and left part is cropped. (b) Second division of rectangle from crop action. To open Downloads folder (in blue box), user thinks one phoneme so that top part of rectangle is selected and bottom part is cropped.

this way, this design provides navigation among directories in the computer system and provides a simple way for computer interaction.

Two designs presented here, alternate between user input for 1 second and user rest state for 1 second. Here the maximum time is consumed in taking user input. Another 1 second is taken so that the user can decide to navigate within a directory. After waiting for the initial 1 second, the system pre-processes the signal, extracts useful features, classifies it to one of the categories, and takes appropriate action according to a classifier output and chosen design implementation (either design1 or design2). All the processing steps can be performed in milliseconds by the computer except for taking user input.

Now we compute average information, denoted as I, for each selection of the IS based BCI system as follows:

$$I = log_2|C| + alog_2 a + mlog_2 \frac{m}{|C|-1} \tag{5}$$

Where, $|C|$ is number of classes in the target class set C, a is the classification accuracy and m is the misclassification rate of the classifier computed as $1-a$. Now information transfer rate (ITR) can be obtained as,

$$ITR = I/T \tag{6}$$

Where, I is average information in bits per trial and T is the total time of each trial. For our analysis, we have $|C| = 2$ since $C = \{0, 1\}$, $a = 0.95$, $m = 0.05$ and $T = 2$ seconds.

Hence, ITR in our case is 0.35 bits per second or 21 bits per minute.

B. Implementation Details

GUI implementation (design1) for displaying rectangles is performed using the Tkinter library in Python. Before starting the GUI, we train the classifier on the training data. In this implementation, 60% of the given data was used for training the CSP and ANN parameters, and then the remaining 40% of the data was used during the testing state. At the starting of each step, the system displays an option to select between crop and switch action, as well as for the user to start thinking. The system then waits for 1 second to capture the EEG recording. In the offline analysis, to select between phoneme and word, the user clicks using the mouse in one part of the rectangle. Then, one trial of either phoneme or word from test data is selected at random. Selection from test data is based on the location of the click in the rectangle. If the location of the click is inside the top or left part of the rectangle, then a random test trial from the phoneme set is selected.

Otherwise, a random test trial from the word set is selected. This data is then pre-processed, CSP transformed, features are extracted, and finally decoded by the classifier. All these steps are repeated for taking inputs for other states. Figure 5 shows the display, rectangle division, and the target folder Downloads (in blue) by two repeated crop actions. Details of the experiment are provided subsequently. By cropping the rectangle, the user reaches the target location and selects the switch window option to double click for the desired folder. The window in which the rectangle is shown has been kept partially transparent so that the user can see the location of the target folder and crop the rectangle accordingly. GUI Design Considerations: many techniques can improve GUI performance as this design provides only a starting point for the creation of online computer control using imagined speech. 1) Dividing the rectangle into multiple parts instead of two. This implies that each step reduces the rectangle size by k instead of 2. For example, if k = 4 then, this new method is twice faster than the method with k = 2. However, a higher value of k requires the high performance of the multiclass classifier. 2) When a classifier does an incorrect output generation, then care has to be taken to circumvent this situation. For this situation to be rectified, the switch option is preferred. If a user detects that the last crop was incorrect, then in the next step, the user decides to switch to the previous rectangle. This regenerates the previous rectangle, and the user can select to crop the rectangle in the next step. 3) Prior to each crop step with crop/switch state, the decision of whether to crop or switch can be skipped for k steps, and the value of k can be decreased with each decision. 4) This design considers only opening a folder by performing a double

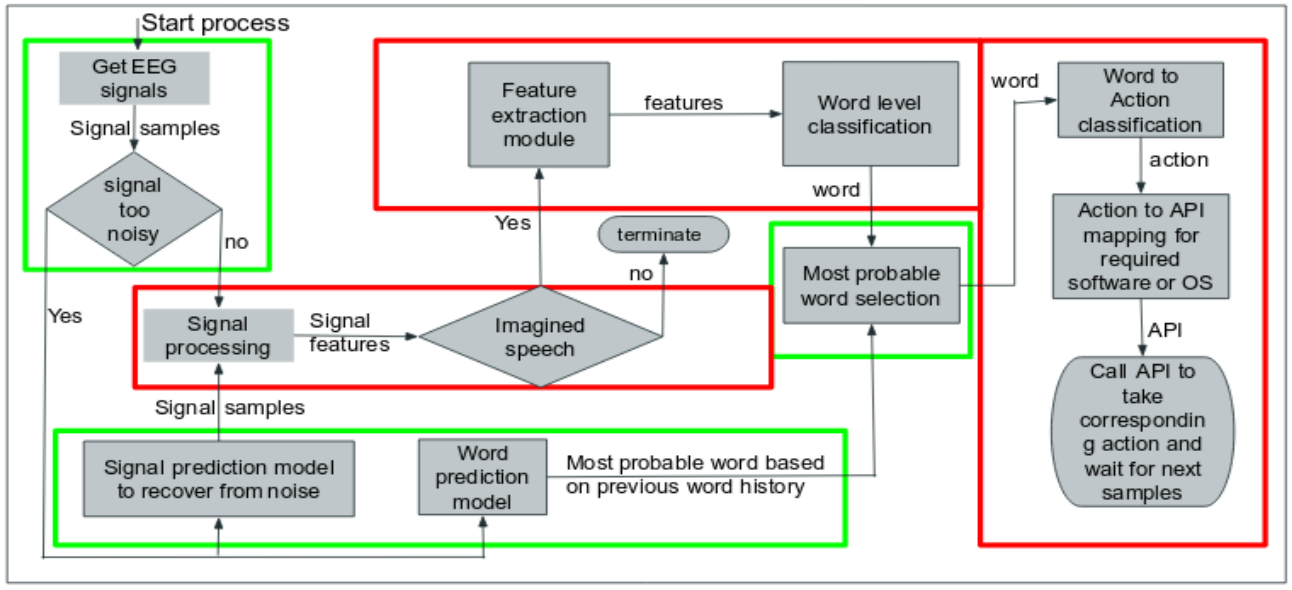


Fig. 6: Flow inside a pipelined system for computer control using IS signals.

mouse click operation. Other options can also be provided for feature enhancement such as a single mouse click, right-click of a mouse, and then creating a new window dictated from the size of the right-click menu. These features take the BCI system towards practical realization. 5) This implementation was done in the Linux Operating System (OS). This means a few components are OS-dependent. Implementations can be made OS independent or developed for multiple operating systems.

C. BCI Pipeline

Based on the proposed approach, we can design a BCI system that identifies the rest state brain condition from the IS condition. If the brain signal corresponds to the IS condition, then it can decode the target word. Here, we have two modules performing two different categorization tasks. Similarly, a BCI system has other components related to artifact detection and removal and for OS interaction. By combining all the components, we propose a data flow framework of the IS based BCI system (Figure 6). This framework is essential for the real-time functioning of the IS based BCI system. The detail of each component is as follows:

After reading brain signals from the EEG device, it is necessary to identify whether the given signal is clean or corrupted from noise. If the signal is clean, then the useful frequency components are extracted. After that, the filtered signal is examined to identify if the IS components are present within the signal. If the components are present, then the useful features are extracted, and the classification model is built to decode the imagined word. Initially, if the signal is noise corrupted, then noise removal or signal reconstruction should be performed. A noisy signal also triggers the word prediction model. This model works based on the word's history. The classifier outputs and word prediction model outputs are compared to identify the most probable word. This word is then mapped to the intended user action. Action is mapped to an Application Program Interface (API). The called API then changes the current computer system state. The modified system state again asks the user for some input and hence provides a new way of brain-computer interaction. As seen in Figure 6, we have implemented processes inside red boxes. Implementation of processes inside green colored boxes is left as future work.

VI. Discussion and Conclusion

In this section, we discuss a few points related to performance on different datasets and also a few design aspects. This we provide concluding remarks.

A. Discussion

In the previous section, we have shown that our proposed approach is equally capable of decoding long words and short vs. long words. Also, our approach generalizes to vowels, and short words IS task by improving the results. We observe that using appropriate features and classification model, word decoding capability can be improved. It happens because words or vowels are different in their speech signal representation. So the process that generates these sounds inside the brain must generate different activation patterns. These activation patterns lead to discriminative EEG signals. Long words are more difficult than vowels and short words in terms of imagined pronunciation. Hence, this additive complexity in silently speaking long words provides more discriminative information and hence improving the classification results for short vs. long words and long words decoding tasks.

This behavior of sophisticated features representation is also supported by the results obtained for the IS signal vs. rest state brain signal. We observe high classification accuracy for long words vs. rest state in comparison to short words vs. rest and vowels vs. rest state brain signals. We observe a difference

of around 5% in classification accuracy when decoding long words IS signals from rest state in comparison to vowels/short words IS signals from rest state brain signals. We believe that model performance will further increase if the time difference of capturing IS signal and rest state signal is increased. In the experiment by Nguyen et al. [3], the rest state condition was immediately followed by the IS condition. So there is some chance that subjects are still in imagination state, and thus feature representation is the same.

This work also presents two new interface designs for computer control with the following benefits and differences from existing designs: 1) This interface design is generic. Though we illustrate for IS based EEG tasks, the design can be expanded for use in other BCI paradigms such as motor imaginary. However, the interface currently being used in P300 speller or motor imaginary to control mouse cannot be easily used in IS tasks. 2) The interface here is shown for binary classification tasks, but it can easily be extended to a multiclass setup to provide faster navigation or providing more features to the user. 3) Design 2 provides an easy way of navigation within a file structure. One the other hand, design 1 is generic. It can be used in a wide variety of computer applications such as folder navigation, browser-control, or navigating through documents (reading). 4) Designs 1 and 2 can be easily extendible for providing more functionality to users, such as providing right-click features of a mouse or double-click and single-click features. 5) One important difference between existing designs and proposed design is that designs presented in this work are reactive as opposed to existing proactive designs.

In the case of reactive design, we wait for the user signal to modify the current system while in the case of proactive design, we keep the system active by automatically moving over available options on the computer screen with the user requiring to provide an input when the target location is reached. The second proactive design is the movement of the horizontal line from top to bottom and vertical line from left to right on the computer screen. These lines form various intersection points on the computer. When the intersection of lines is at the target location, then the user provides an input, and the system state changes accordingly. After that, the whole process repeats. The third design might be a circular rotation of a line segment starting from the center of a computer screen up to the end of a computer screen. When the line intersects with a target location such as some folder, then the user provides input, and the line rotation stops (say at an angle of degree θ from positive-x/horizontal direction). Then line segments of different lengths are displayed from small lengths up to a size of max screen resolution along the direction θ. The user provides second input whenever a new line segment reaches the target location. The above are few examples of proactive interface design to operate the computer system.

Note that the designs presented here were tested in a partial online setting. In a partial online setting, rather than taking input from an EEG device, the input was taken from a user mouse click. Based on the location of the click, a corresponding trial data from the test set was picked, processed, classified, and system state was changed. The new system state was shown to the user, and then the user again provides input to reach a target location. This provides a closed-loop of the BCI system for human-computer interaction.

B. Conclusion

This work shows that with machine learning methods, it is possible to design an imagined speech signal based brain-computer interface system for human-computer interaction. In doing so, we have presented an approach using covariance matrix as input feature and ANN as classification model for decoding IS signal. This approach outperformed existing means when applied on an IS dataset. We show that IS signals can be differentiated from other brain signals, and the length of words is a useful criterion in discriminating words. In the future, we will work on improving model performance, developing new ways of computer interaction, and IS signal prediction models to recover from high noise scenarios.

REFERENCES

[1] J. Wolpaw and E. Wolpaw, *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, USA, 2012. [Online]. Available: https://books.google.co.in/books?id=tC2UzuC_WBQC

[2] G. Hickok and D. Poeppel, "The cortical organization of speech processing," *Nature Reviews Neuroscience*, vol. 8, pp. 393–402, 2007.

[3] N. T. Sahin, S. Pinker, S. S. Cash, D. L. Schomer, and E. Halgren, "Sequential processing of lexical, grammatical, and phonological information within broca's area." *Science*, vol. 326 5951, pp. 445–9, 2009.

[4] M. Teplan, "Fundamental of eeg measurement," *MEASUREMENT SCIENCE REVIEW*, vol. 2, 01 2002.

[5] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass braincomputer interface classification by riemannian geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, April 2012.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[7] C. H. Nguyen, G. K. Karavas, and P. K. Artemiadis, "Inferring imagined speech using eeg signals: a new approach using riemannian manifold features." *Journal of neural engineering*, vol. 15 1, p. 016002, 2018.

[8] I. Psorakis, T. Damoulas, and M. A. Girolami, "Multiclass relevance vector machines: Sparsity and accuracy," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1588–1598, Oct 2010.

[9] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. r. Muller, "Optimizing spatial filters for robust eeg single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.

[10] C. S. DaSalla, H. Kambara, M. Sato, and Y. Koike, "Single-trial classification of vowel speech imagery using common spatial patterns," *Neural Networks*, vol. 22, no. 9, pp. 1334 – 1339, 2009, brain-Machine Interface. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608009000999

[11] B. Min, J. Kim, H. jun Park, and B. Lee, "Vowel imagery decoding toward silent speech bci using extreme learning machine with electroencephalogram," in *BioMed research international*, 2016.

[12] I. Jolliffe, *Principal Component Analysis*. John Wiley & Sons, Ltd, 2014. [Online]. Available: http://dx.doi.org/10.1002/9781118445112.stat06472

[13] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug 1996. [Online]. Available: https://doi.org/10.1007/BF00058655

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[15] A. Barachant *et al.*, "pyriemann: Biosignals classification with riemannian geometry," 2015, library available from http://pyriemann.readthedocs.io/en/latest/index.html. [Online]. Available: https://github.com/alexandrebarachant/pyRiemann

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.