CLASSIFICATION OF POSITIVE AND NEGATIVE STIMULI USING EEG DATA FUNCTIONAL CONNECTIVITY AND MACHINE LEARNING

by

Sai Jaya Sasanka Bhamidipati, B.Tech.

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Engineering

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

DECEMBER, 2018

CLASSIFICATION OF POSITIVE AND NEGATIVE STIMULI USING EEG DATA FUNCTIONAL CONNECTIVITY AND MACHINE LEARNING

by

Sai Jaya Sasanka Bhamidipati

APPROVED BY

_____

Unal 'Zak' Sakoglu, Ph.D., Chair

_____

Jiang Lu, Ph.D., Committee Member

_____

Liwen Shih, Ph.D., Committee Member

APPROVED/RECEIVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

_____

Said Bettayeb, Ph.D., Associate Dean

_____

Ju H. Kim, Ph.D., Dean

Acknowledgments

During my master's program, I learned a lot of things it would not have been possible without the help and support of the great number of individuals. Firstly, I would like to thank my advisor, Dr. Unal 'Zak' Sakoglu, for his continuous support and guidance during my graduate studies and my thesis work. His valuable guidance, constant encouragement added considerably to my research experience. He consistently allowed me to work on my thesis and guided me in the right direction whenever I needed it.

I would also like to acknowledge Dr. Jiang Lu and Dr. Liwen Shih for serving on my committee. I am gratefully for their valuable comments before and during my thesis defense.

Special thanks to Prof. Dr. Hakduran Koc, program chair and associate professor of Computer Engineering.

I extend my thanks to the Dean's Office, Librarians, Writing Centre, and UHCL staff.

I would also like to thank the College of Science and Engineering at the University of Houston Clear Lake University for providing me with the graduate research assistantship opportunity for my master's education.

Finally, my deepest gratitude goes to my family. This thesis would not have been possible without the help, support and continuous encouragement of my beloved family.

ABSTRACT

CLASSIFICATION OF POSITIVE AND NEGATIVE STIMULI USING EEG DATA FUNCTIONAL CONNECTIVITY AND MACHINE LEARNING

Sai Jaya Sasanka Bhamidipati
University of Houston-Clear Lake, 2018

Thesis Chair: Dr. Unal 'Zak' Sakoglu

Electroencephalography (EEG) provides electrical measures of brain activity by monitoring voltage fluctuations of the collective neural activity in different parts on the cortex of the brain. Recently, there have been numerous applications of machine learning techniques to classify events or participants based on EEG data in the biomedical field. EEG data are rich in the sense that one can extract many features from the data. This makes feature selection and reduction an important step in EEG based classification. Feature selection and correlations between features for classification of EEG data typically depend on time-frequency characteristics of the EEG channels, which represent data from different parts of the brain cortex.

In this proposed work, we calculated functional connectivity (FC) between different EEG channels as our features for classification and applied it for classification of positive and negative visual stimuli. Previously, EEG data were collected from 12 participants (6 females and 6 males) while they were observing positive and negative images in a random order and the data were completely de-identified. After filtering of the noise in the data, we extracted FC features. From these FC features for each of the

stimuli, we reduced the number of features using techniques which included correlation-based and principal components based methods. Once the features were selected, we implemented classification of positive vs. negative stimuli using classification techniques support vector machines, decision trees, random forests, k nearest neighbors, Gaussian process, Adaboost, quadratic discriminant analysis and logistic regression. We compared the classification accuracy results Support vector machine and Logistic regression provided the highest classification accuracy of whether a participant was seeing a positive or negative image, with accuracies of up to 71.9% and 71.4% for each of the participant, respectively.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Machine Learning

Over the last few years, Machine learning is defined by the Encyclopedia Britannica [1] as "an artificial intelligence discipline concerned with the implementation of computer software that can be learned autonomously." Machine learning is the study and development of algorithms that can learn and make predictions form a set of data. To make the best predictions based on a set of rules, a machine learning algorithm makes the data-driven prediction based on a model, a pattern or a structure that summarizes a set of input and output data. [2]

Machine learning tasks can be classified into three broad categories based on the nature of the model learning feedback available to the system: Supervised learning, unsupervised learning, and reinforcement learning. [3]

In supervised learning, the computer is given a set of input samples and designated output samples or "labels" by a "teacher" (hence the term "supervisor") with the goal of computer program learning a rule to map the inputs to the outputs [2]. This type of learning is used to model a problem that may be simple for humans to understand, but too complex for a set of rules to code. An example problem is to be presented with a lineup of pictures and to be asked to identify each picture. It's simple for humans to solve this problem, but difficult to describe it into an algorithm for a computer to solve. Types of supervised learning are classification and regression.

In classification, inputs are divided into two or more classes (or categories, or groups). The model then places incoming data in one or more of these categories. Based on which categories the data is placed in, it makes a classification. Prediction is similar to classification but is used with continuous data to predict future data points, often using

some type of regression [10]. The difference between classification and regression is shown in Figures 1.1 and 1.2. In Figure 1.1, the red line divides the two classes. When new data are given to the classification algorithm to get predicted, the predicted outcome is determined by where the data lies on the graph. In Figure 1.2, the red line shows the trend of the data. Using this line, future predictions can be made.

In our proposed work, we performed binary (two categories) classification using features were extracted from different channels of electroencephalogram (EEG) data. The details of the EEG and features to be extracted are presented in the next section.

Figure 1.1

Example of classification learning

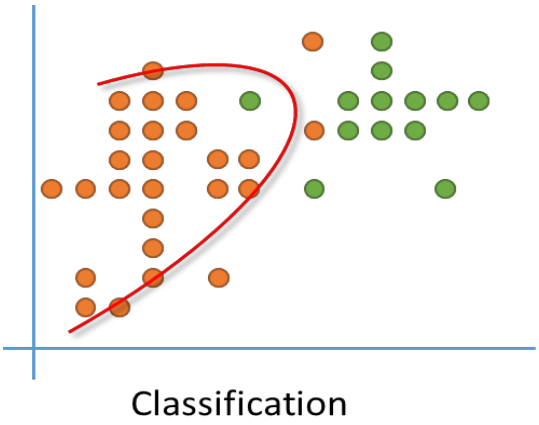
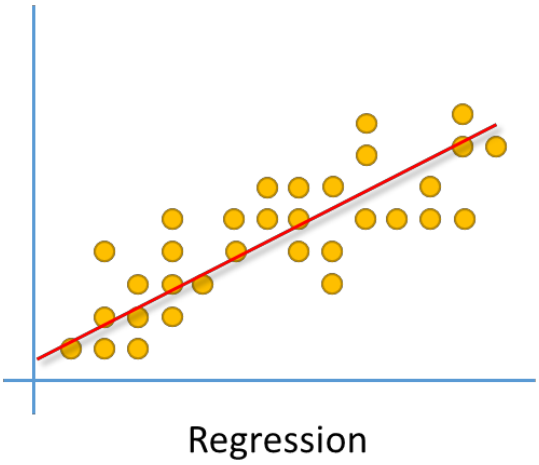


Figure 1.2

Example of regression learning



1.2 Electroencephalography (EEG)

EEG is an electrophysiological monitoring technique for recording and interpreting electrical activity in the brain. This phenomenon was first observed in 1875 by Richard Caton, a physician practicing in Liverpool, who presented his findings of the electrical activity of rabbits and monkeys in the British Medical Journal [19]. The first recording of human EEG was done in 1924 by a German physiologist and psychiatrist Hans Berger. Berger also invented the first electroencephalogram [5].

The nerve cells of the brain generate electrical impulses that fluctuate in distinct rhythmic patterns. EEG waves are measured with typically with 16 to 128 pairs of electrodes, placed on the scalp. The difference in voltage between the pairs is recorded as the signal. Nowadays, 32, 64, 128 and even 256 channel EEG systems are available for research and clinical purposes. Typical interpretations of EEG signals are done by spectral analysis. A spectral analysis of EEG signals shows the brain pattern in the frequency domain [6]. When looking at the frequency domain, various frequency bands

are associated with different rhythmic activities. Five commonly bands are known as alpha, beta, theta, delta, and gamma and associated brain activities are summarized in Table 1.1. An example of each band is plotted in Figure 1.3. It presents the original EEG wave captured and filtered into different bands. The EEG has DC offset of 4200 uV removed.

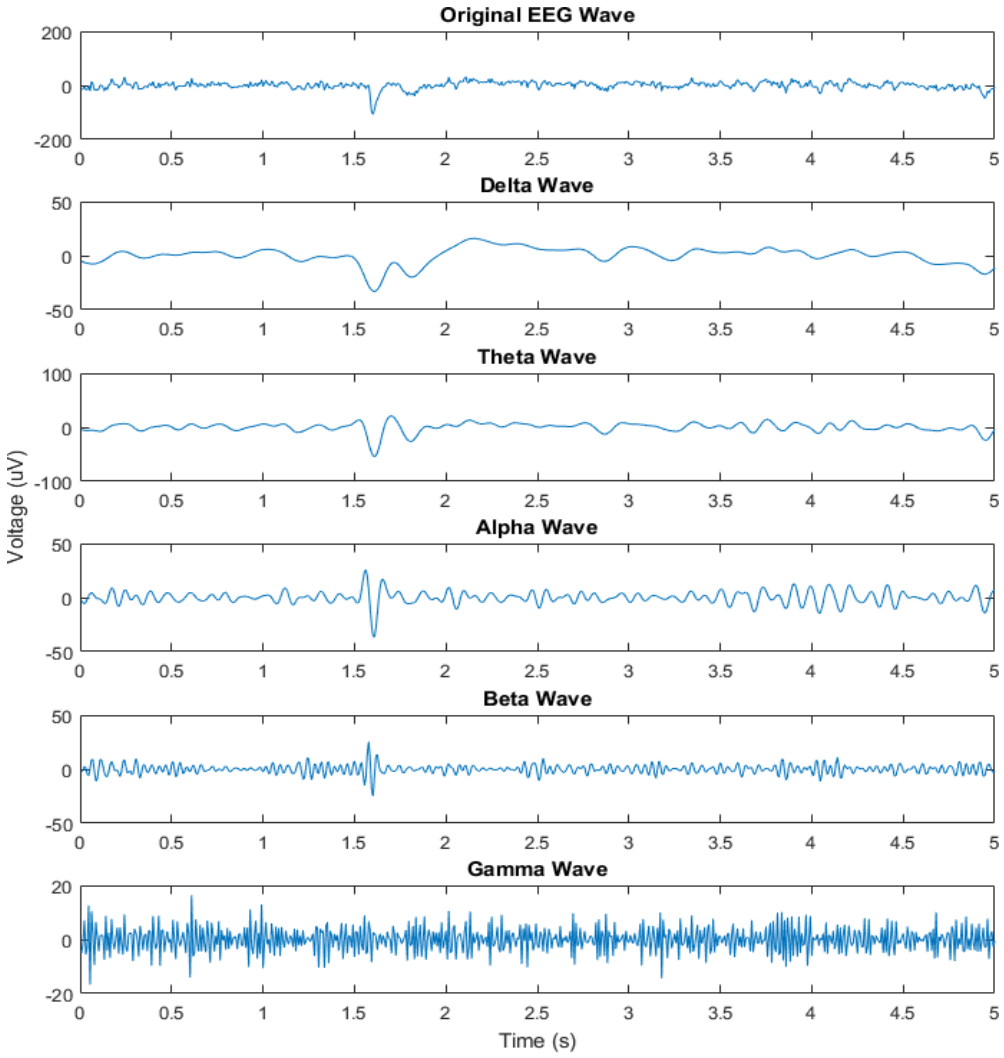
Table 1.1

EEG Bands and Frequencies

Band	Frequency (Hz)	Associated Activity [16]
Delta	1-4	Deep meditation, sleep, and source of empathy
Theta	4-7	Learning, and memory
Alpha	8-15	Mental coordination, calmness, and alertness
Beta	16-31	Problem solving, judgment, decision making, focused mental activity
Gamma	31+	Love, high altruism, and higher virtues

Figure 1.3

Comparision of EEG waves



1.3 Classification Based on EEG Data

The classification of the EEG signals is highly challenging due to the variability of the EEG signals and various sources of noise in EEG signals. There are different ways to handle feature selection and classification of EEG signals. During this thesis work, we referred to multiple IEEE papers on the classification of EEG signals using machine learning techniques. We used some of the widely used ones, which are listed below.

In a recently published paper [21], Amin et al. describe the classification of EEG signal using discrete wavelet transform-based feature extraction. In their paper, the discrete wavelet transform is applied to EEG signals and calculated relative wavelet energy in terms of detailed coefficients and the approximation coefficients of the last decomposition level. During the process, they applied frequency range 0.53 Hz-3.06 Hz for approximation coefficients and 3.06 Hz-6.12 Hz for detailed coefficients. The extracted relative wavelet energy features are passed to classifiers for the classification purpose. They have evaluated four different measures i.e., accuracy, sensitivity, specificity and precision values.

In an EEG classification method [11] using the same EEG data as in this thesis, the authors applied EEG classification by using features such as alpha, beta, and gamma bands. The purpose of that study was to develop the method by extending (augmenting) the spatio-temporal data either directly in the data space or in the feature space using the temporally-augmented versions of data or the features. They introduced maximum temporal lag value, denoted with L, which was studied by picking different L values, and the most optimal L value was determined for the best EEG classification for different classification algorithms and data size, controlled by the parameter N, the number of data points. They obtained around 60% average classification accuracy for binary

classification of positive vs negative images, significantly higher than a chance accuracy of 50%.

In this thesis, we develop and apply of a new classification method by considering functional connectivity between EEG channels as feature sets and apply different machine learning algorithms to obtain better accuracy results.

1.4 MATLAB

The name MATLAB stands for Matrix Laboratory [7]. MATLAB is a high-level scientific programming language developed by Math Works, Inc., which is widely used in the scientific community. MATLAB integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for electrical engineering, computer science, mathematics, finance, biology, statistics, geology, aerospace, controls, meteorology, bioinformatics, and medical imaging. We utilized MATLAB and its various toolboxes, such as signal processing toolbox, for preprocessing and filtering, calculation of functional connectivity metrics, and extraction/selection of features [4]. A detail of how MATLAB was used in this work's methodology provided in Chapter 2 below.

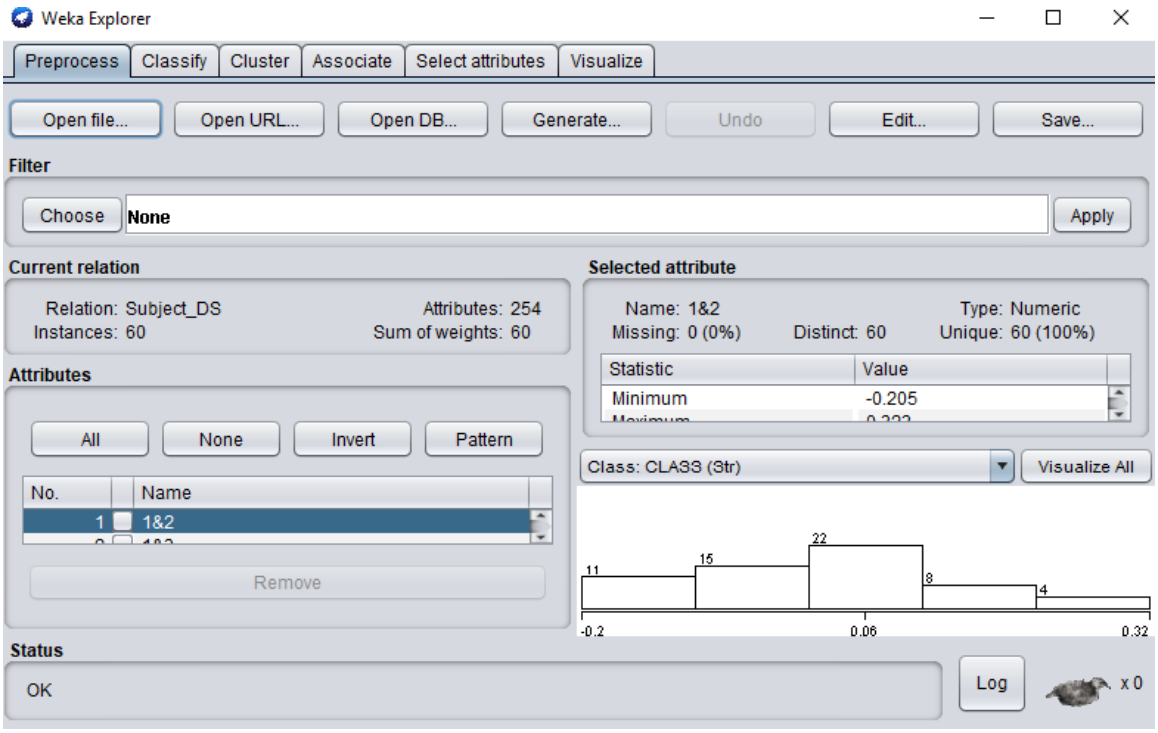
1.5 WEKA

Waikato Environment for Knowledge Analysis (WEKA) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand [8]. It is free software licensed under the GNU General Public License. It contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions. WEKA supports several standard data mining tasks, more specially, data preprocessing,

clustering, classification, regression, visualization and feature selection user interface of WEKA. We utilized WEKA for classification of our EEG data. Figure 1.4 provides the main GUI of WEKA. A detail of how WEKA was used in this work is provided in Chapter 2 below.

Figure 1.4

GUI of WEKA tool



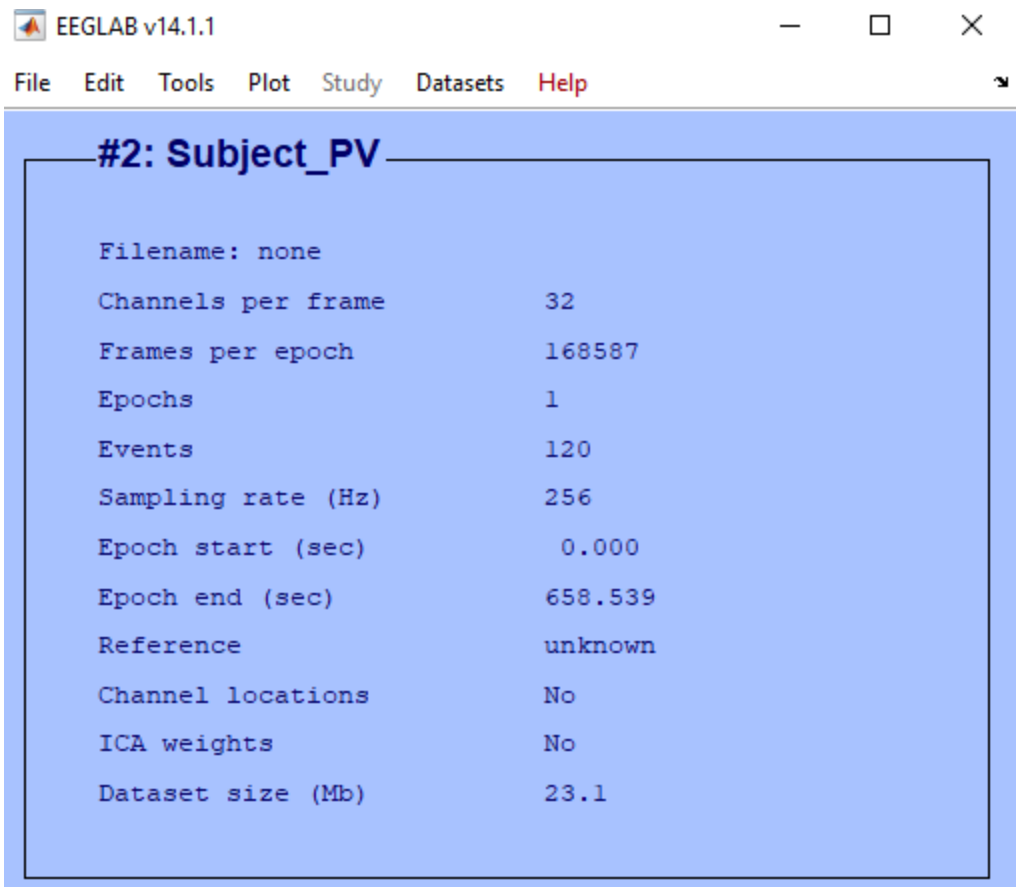
1.6 EEGLAB

EEGLAB is an open source MATLAB-based toolbox for electrophysiological signal processing developed by the University of California at San Diego [7], [9]. It provides an interactive graphical user interface (GUI) and command-line options to process the continuous EEG signals. It has built-in functionality to visualize EEG signals, filtering the EEG signals, rejecting artifacts like eye blinks and muscle movements, and

extracting the epochs in the EEG data. Through its visualization tools, cerebral cortex can be visualized to understand the regions where the neurological activity is concentrated during various neurological and physiological tasks respectively. EEGLAB is the most widely used EEG data processing toolbox in the world and it has been used for our research. Figure 1.5 provides the main GUI of EEGLAB. We used EEGLAB for extracting negative and positive epoch data, removing bad connecting channels which were not in contact with cerebral cortex during the experiments, specialized preprocessing and filtering of EEG data. Details of how EEGLAB was used are provided in Chapter 2 below.

Figure 1.5

GUI of EEGLAB tool



CHAPTER 2: METHODOLOGY AND DESIGN

2.1 EEG Data Collection

EEG data were previously collected and de-identified as part of a previous project. Sakoglu et al. had previously obtained IRB approval from the local IRB committee of the institution where the EEG data collection experiment was done [11]; the details of the experiment are described in detail in [11]. During the experiment, negative (disturbing) and positive (pleasant) pictures, selected from Geneva Affective Picture Database (GAPED) were shown to participants. GAPED is a collection of pictures which were shown to people in an emotional research experiment [20]. These images are categorized into negative, positive and neutral categories in the database. The content of the negative images is mainly the scenes which induce negative emotions due to animal mistreatment and human rights violation. The content of positive images is mainly the animal and human babies, nature scenery, etc. Neutral images are everyday images like houses, general tasks, etc. The images in the GAPED database had been rated by the experiment participants on valence, arousal, and the congruence of the represented scene with internal (moral) and external (legal) norms. Images which had high and low valence scores had been selected, for the positive and negative categories, respectively [20]. The EEG data were collected from 12 young adult participants (6 males, 6 females, ages 22 to 38), and completely de-identified. During the EEG data collection, each participant was shown 30 positive and 30 negative images, and each image was shown for duration of 3 seconds, with 7 seconds-long interval between the images. In this work, each of these 3-seconds is referred to as an “epoch”. The images were shown to the participants in a random order so that they could not predict the category of the next image (i.e. whether it is negative or positive). A blank black image with a fixation cross at the center was shown during the 7-second interval. A sample timeline and two sample images are

presented in Figure 2.1 [20]. Continuous EEG data were recorded at a sampling rate of 256 Hz by the EEG device (32 channel, wireless, dry-contact, Mindo Sepia 32H, made by BRC-Taiwan) while the participants were being presented the images. The data were completely de-identified. An image of the EEG device is shown in Figure 2.2 [11].

Figure 2.1

A sample snapshot of the timeline and some samples of the pictures that were shown to participants during the EEG experiment [20].

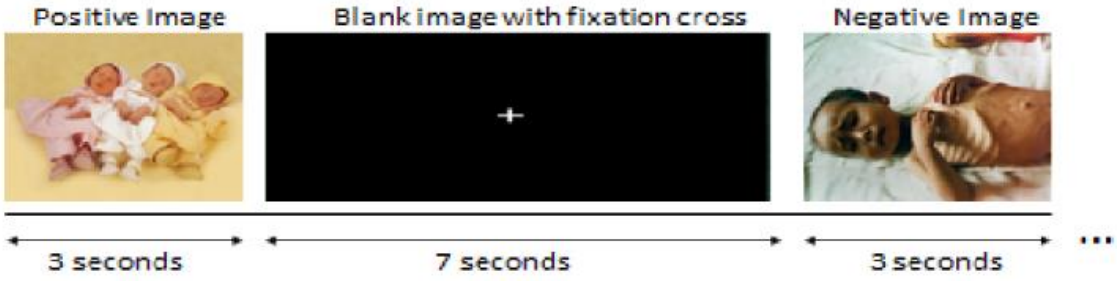


Figure 2.2

32 channel Mindo dry-contact bluetooth EEG system with which the dataset was collected [11]

2.2 Channel Location

The 32 electrode channel locations for the Mindo Sepia 32H EEG device are presented in Figure 2.3. Some of the channel locations, O1, Oz, O2, located on the back of the scalp, near the visual areas of the brain, are associated with processing of visual stimuli such as images, and those channels may contribute more to the classification when compared with other channels, which correspond to other brain regions. In order to visualize the cerebral cortex we supplied the electrode location file to the EEGLAB. Each site has a letter to identify the lobe and a number to identify the hemisphere location. The letters F, T, C, P, and O stand for frontal, temporal, central, parietal, and occipital lobes, respectively even numbers (2, 4, 6, and 8) refer to electrode positions on the right hemisphere, whereas odd numbers (1, 3, 5, and 7) refer to those on the left hemisphere. Figure 2.4 and figure 2.5 represents the 2D and 3D view of channel locations.

Figure 2.3

Channel location of EEG data

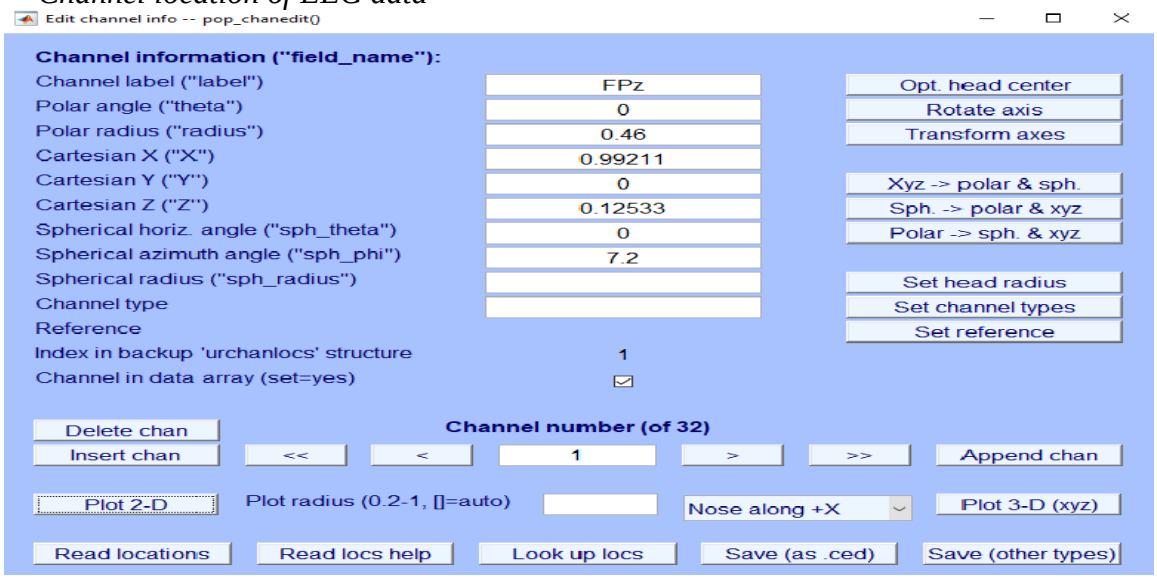


Figure 2.4

2D view of channel location

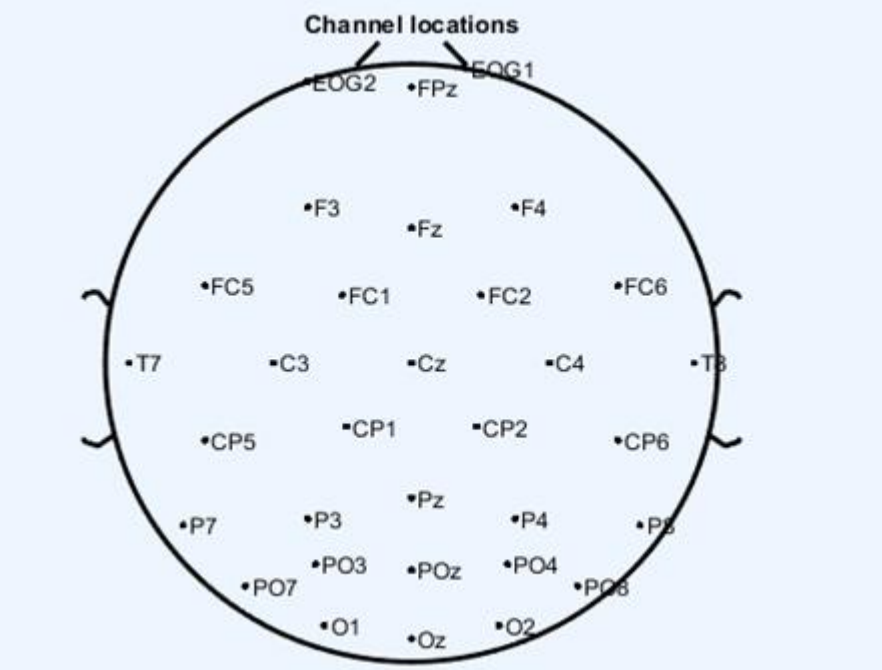


Figure 2.5

3D view of channel location

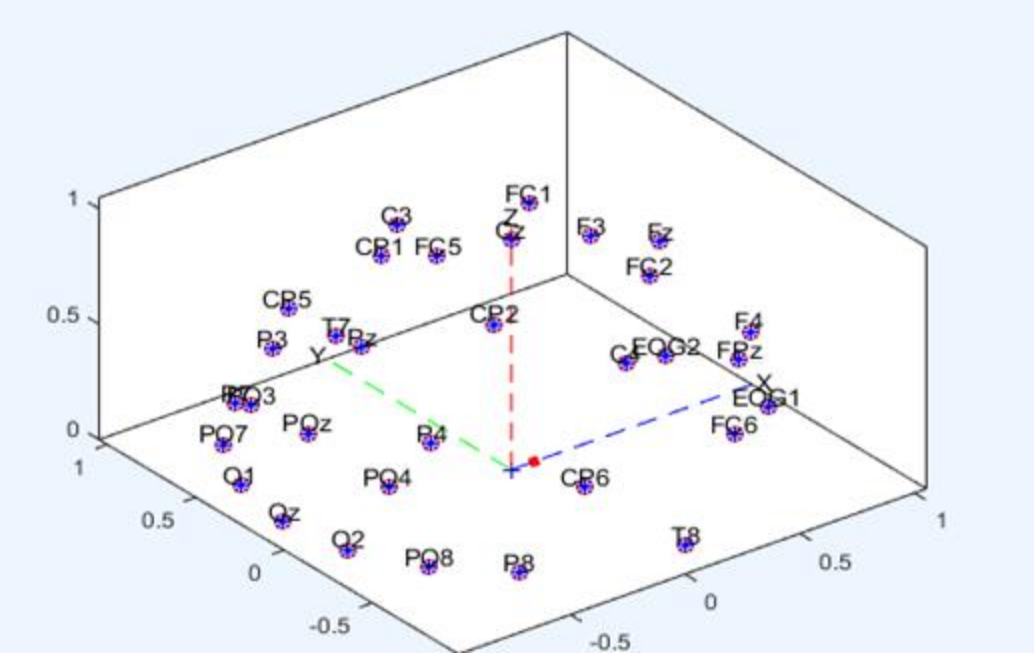
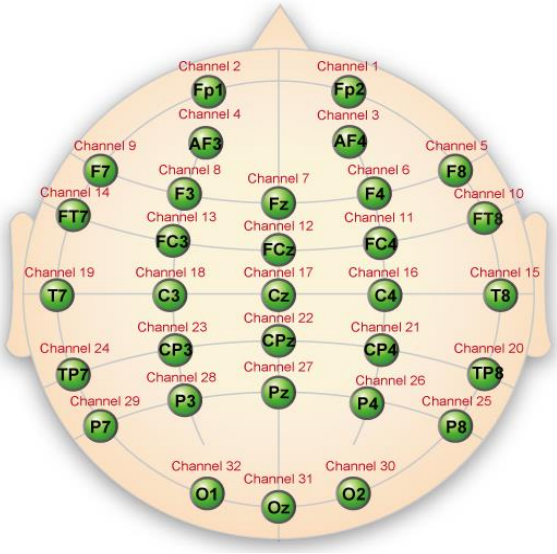


Figure 2.6

Electrode channel locations of 32-channel Mindo Sepia 32H model EEG device [11]



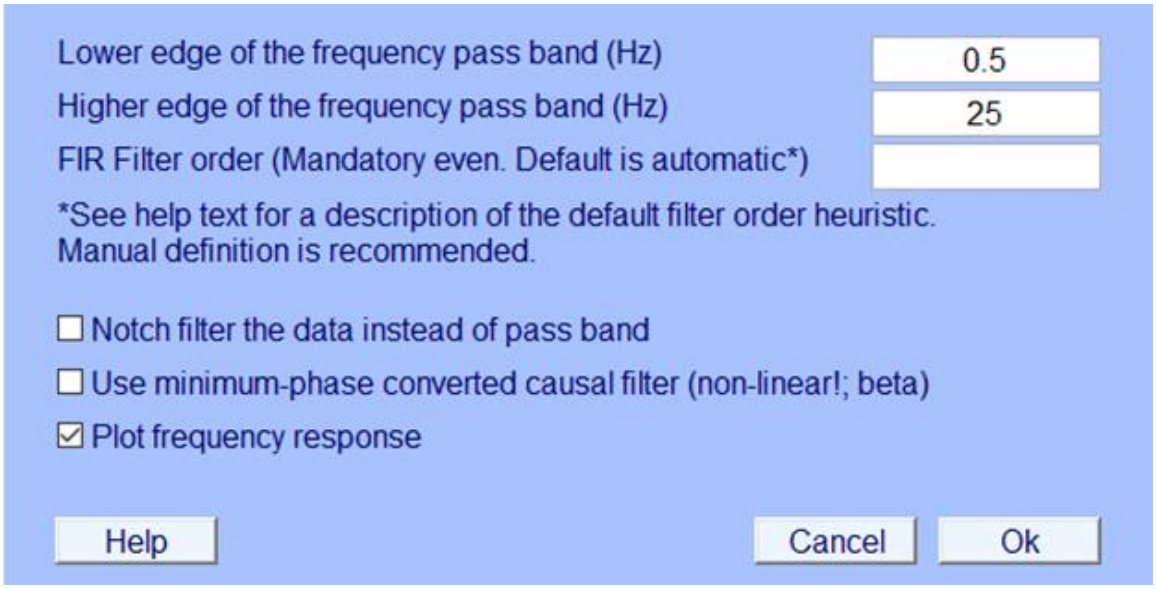
2.3 Preprocessing of EEG RAW DATA

The EEG data were preprocessed with EEGLAB toolbox developed in MATLAB. The pre-processing stage generally includes initializing input parameters (sampling frequency, number of channels), epoch extraction (negative and positive epoch data) and filtering the collected EEG data with the EEGLAB [5] [9]. The sampling rate is the number of samples of EEG signal carried per second, and it is measured in Hz. 256 Hz sampling frequency was used to record EEG signal for all of the 12 participants.

Filtering is used to remove linear trends, noise and spikes, it is often desirable to high-pass and low pass filter the data. Filtering continuous data minimizes the introduction of filtering artifacts at epoch boundaries. We used 0.5 Hz for lower frequency range, and two cases of high-frequency cut-off: first 45 Hz, and second 25 Hz, to remove more noise.

Figure 2.7

EEGLAB's graphical user window interface for filtering



During EEG data collection for most of the participants, a group of 9 channels were mostly not properly in contact with cerebral cortex during the experiments for each of the participants. These groups of 9 channels were channels 8,9,13,14,18,22,24,25,29. By using the EEGLAB, we removed these respective 9 channels' data from every participant, to keep the same channels for every subject for analysis. We extracted the remaining 23 channels and extracted data from EEGLAB into a text file. We then performed band-pass filtering on the data using the parameters mentioned above.

Figure 2.8

EEGLAB GUI interface with 23 channels sample data

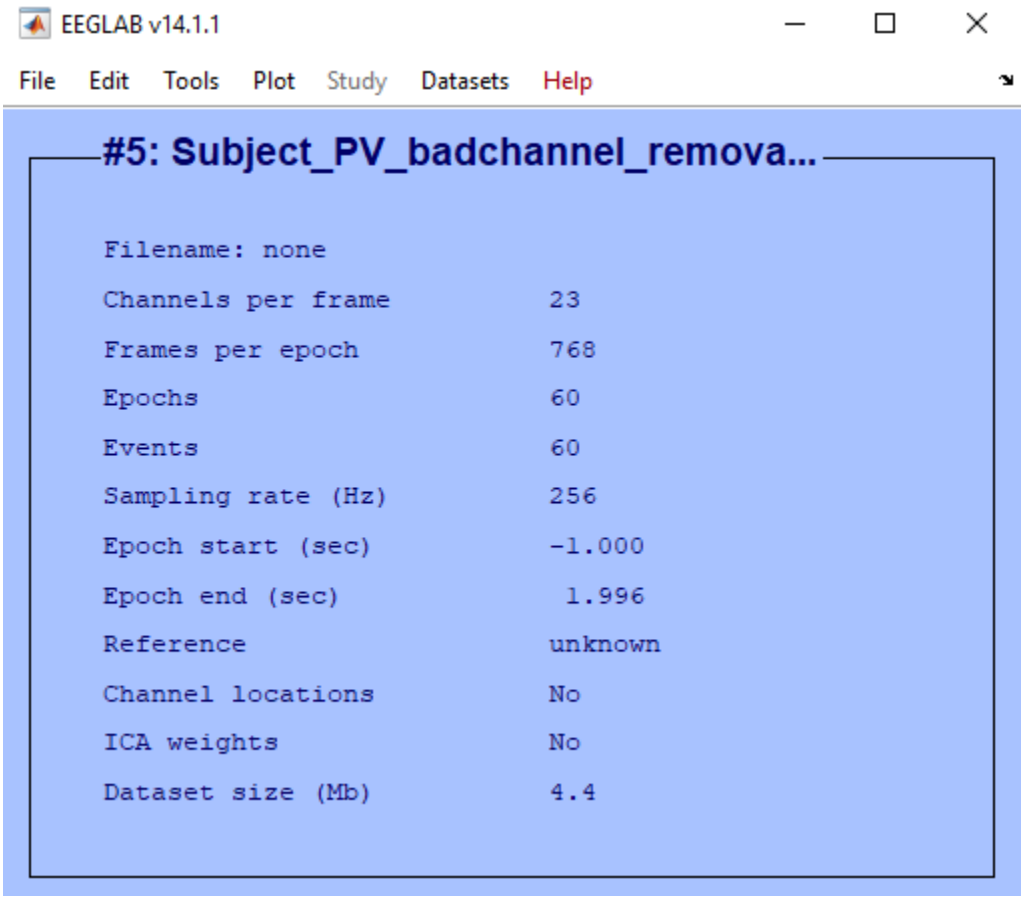
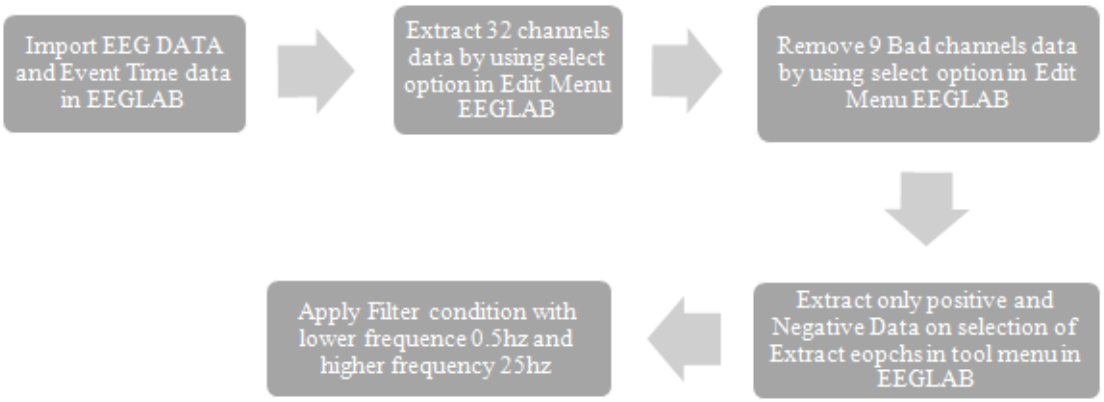


Figure 2.9

Block diagram describing the procedure followed in EEG preprocessing steps.



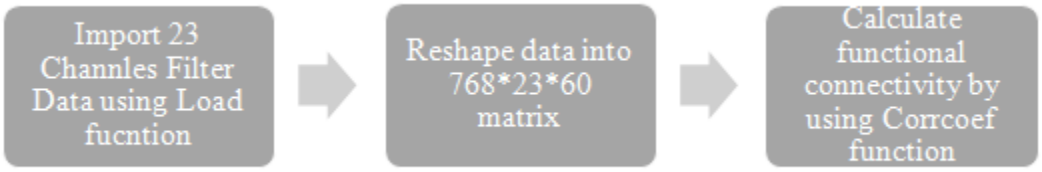
2.4 Functional Connectivity Features

After the preprocessing step, functional connectivity (FC) metrics between each channel was evaluated for each of the 30 positive and 30 negative visual stimuli, for each of the 12 participants. The correlation coefficient between each pair constituted the measure of FC. Duration of each stimulus, which is 3 seconds, is called an “epoch”. Since the sampling frequency of the EEG data is 256 Hz, each channel’s single epoch has 768 data points. Since there are 60 images, there are 60 epochs, for each channel and for each participant. With 23 channels, there are (23, 2) = 253 paired combinations of channels, for each epoch, for each participant. Therefore, for each epoch (which corresponds to a positive or a negative image), we have 253 FC numbers, which can potentially be used as raw features or attributes for the classification of positive vs negative image viewing.

Functional connectivity script was custom-developed in MATLAB programming language. By using correlation coefficient function between 23 channels for all 60 epochs, 253 functional connectivity features were extracted using MATLAB.

Figure 2.10

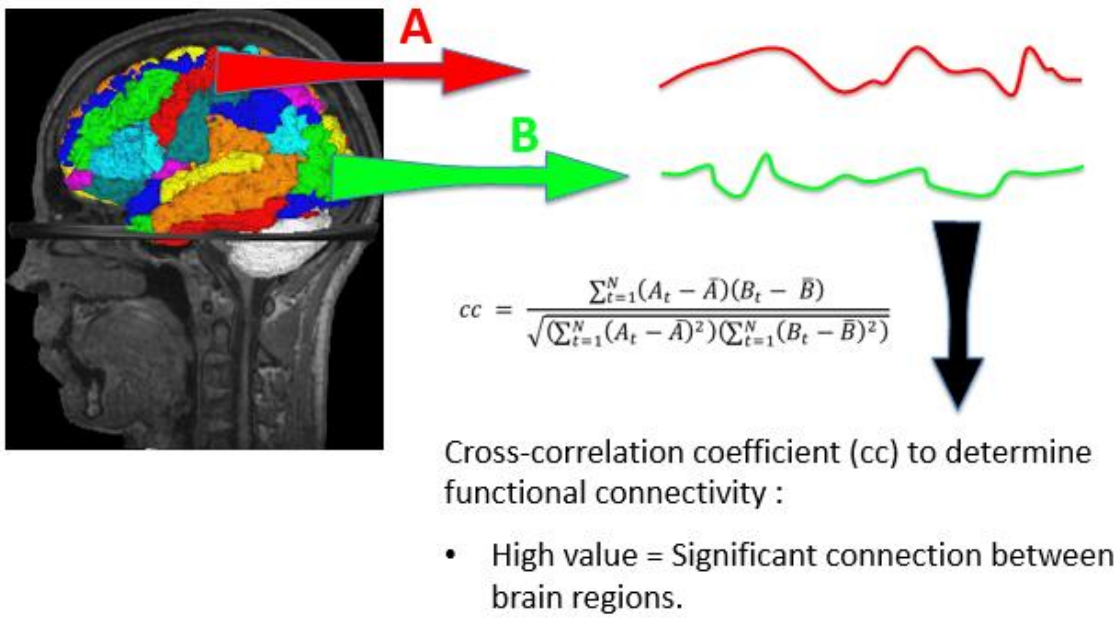
Block diagram describes calculation of functional connectivity using MATLAB



Using all of the raw features would likely yield low classification results since some of the channels are noisy and also since there are much more features than the items to be classified (253 vs. 60), which is notoriously known as “curse of dimensionality” in machine learning and classification. Therefore, we utilized feature reduction techniques to find the most contributing FCs before utilizing them as features in the classification algorithms; Figure 2.11 summarizes computation of FC for a channel-pair. FC is basically the correlation coefficient between the pair of EEG signals for each epoch.

Figure 2.11

Computation of functional connectivity (FC) metric between two hypothetical EEG signals from two hypothetical channels A and B, which correspond to neural electrical activity of two different brain regions.



2.5 Recursive Feature Elimination

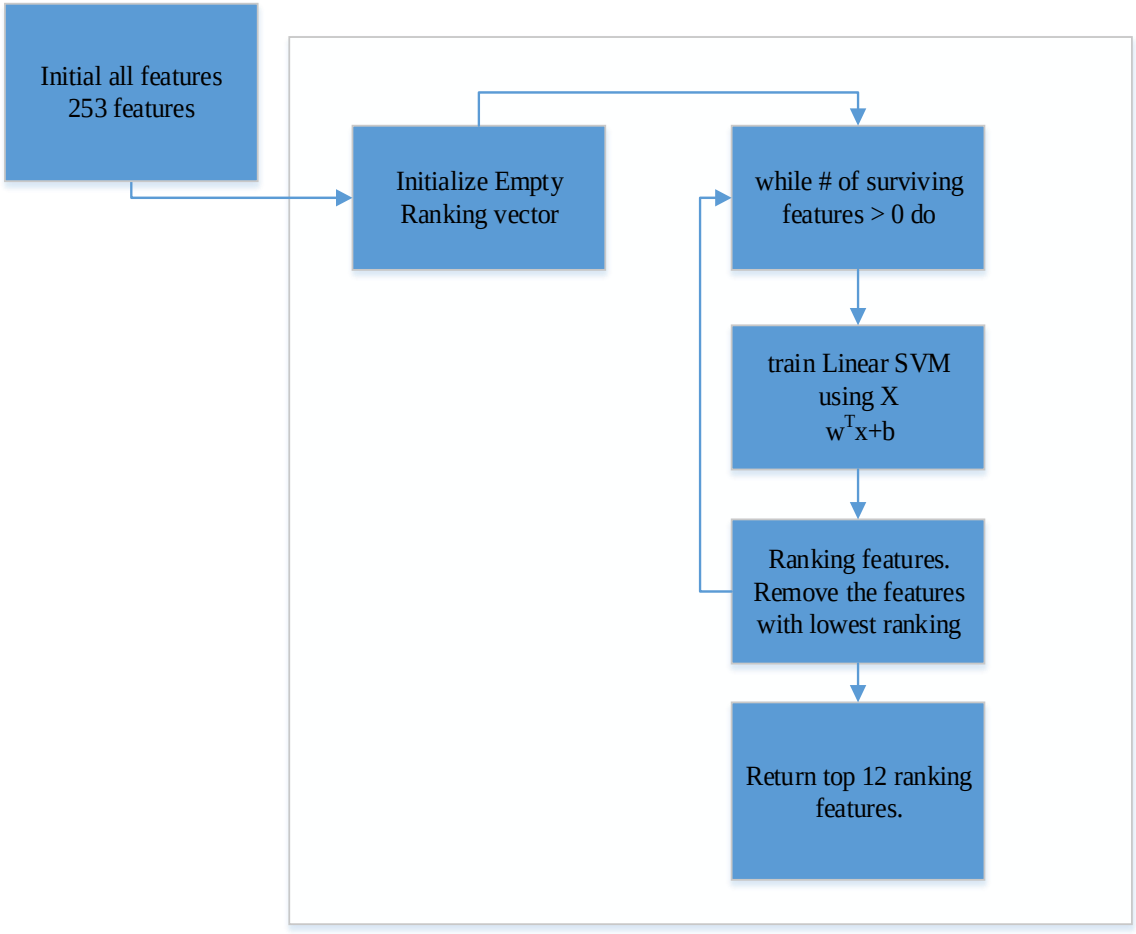
Feature selection is to select a subset of relevant features from a larger set of raw/original ones in terms of some pre-defined criteria such as classification performance or maximum variance. It plays a significant role in machine learning applications. For classification with the small number of training samples (in our case, 60) and high dimensionality of raw features (in our case, 253), feature selection plays an important role in avoiding over fitting problem and improving classification performance.

One of the commonly used feature selection methods for small samples problems is recursive feature elimination (RFE) method [17]. The RFE works by recursively removing attributes and building a model on those attributes that remain. It uses the

model accuracy to identify which attributes (and the combination of attributes) contribute the most to predicting the target attribute. In this thesis work, the RFE method utilizes the generalization capability embedded in support vector machines and is thus suitable for small samples problems. Therefore RFE tends to discard "weak" features which make it. Removing features of low importance can improve accuracy, and reduce both model complexity and over fitting. Figure 2.12 explains support vector machine (SVM) based recursive feature elimination algorithm. It is difficult to analyze high dimensional EEG datasets, since it contains very large feature sets, which causes model learning to be more difficult and also degrades the generalization performance of the learned models.

Figure 2.12

Flow chart of recursive feature elimination SVM

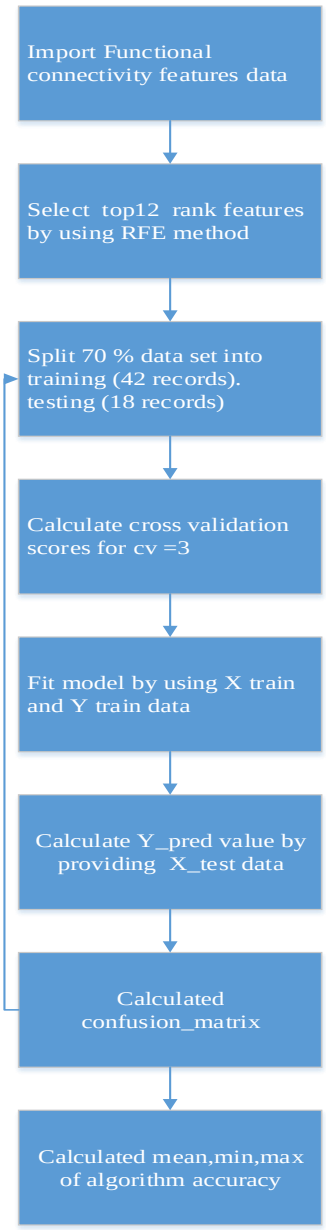


In our case, we have a total of 253 input features, so it is very hard to represent 253 high dimensions. Not all of the 253 features contribute significantly to the classification result. We used Python [12] Scikit to teach feature selection class by importing RFE method and support vector classifier model. After experimenting with the number of selected features, we observed that by selecting 12 features, the classification accuracy was improved and it also reduced both model complexity and over-fitting.

Figure 2.13 describes step-by-step process followed in K-fold testing in our EEG data classification. “Records” in the figure refers to the epochs.

Figure 2.13

End to End classification flow chart



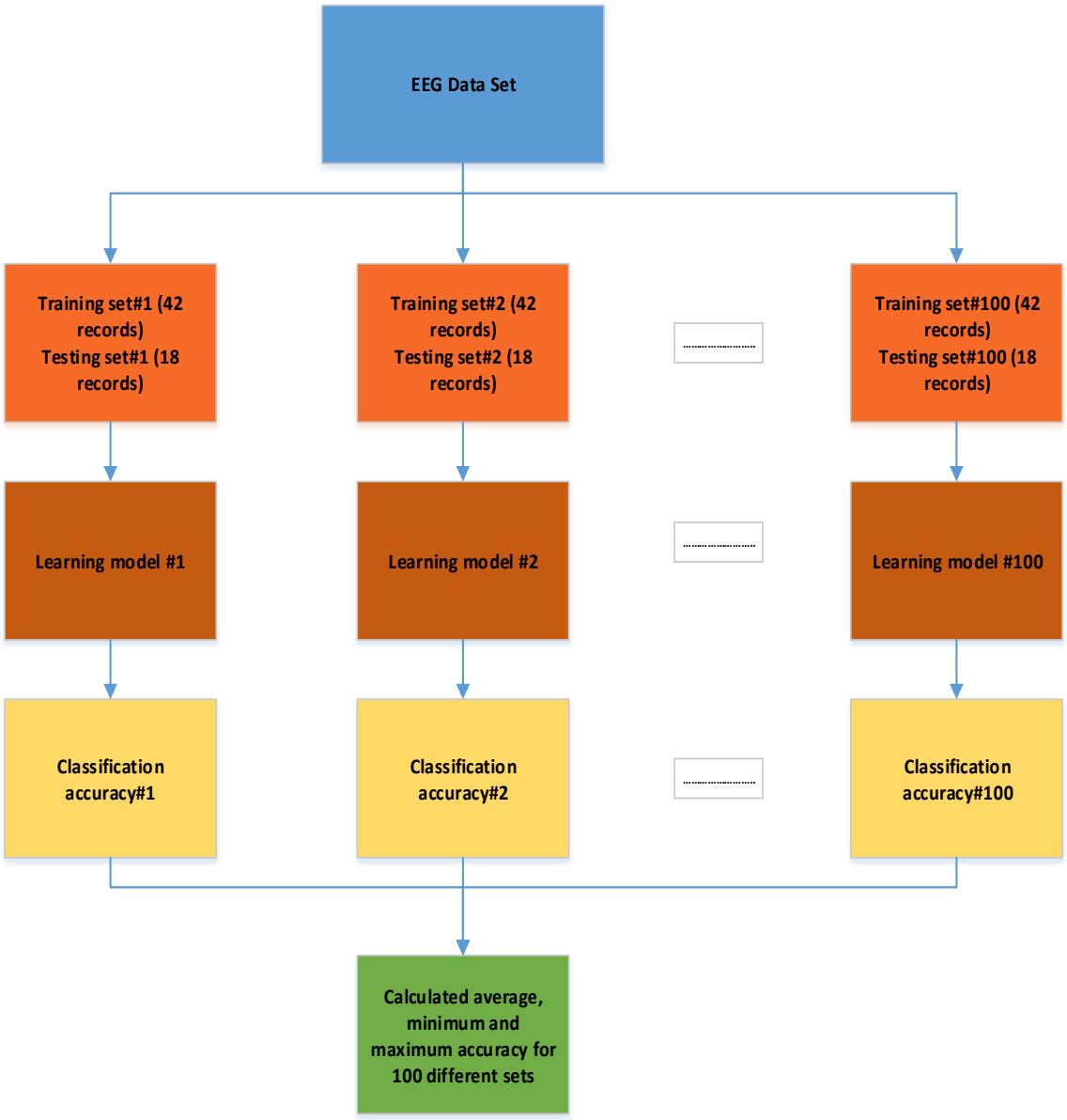
2.6 Bootstrapping

For each of the 12 participants of the EEG dataset, we have limited a number of functional connectivity samples (60 total, 30 positive and 30 negatives). Bootstrapping method [18] is a useful approach to use when classifying based on the limited number of inputs or samples. In bootstrapping, one randomly splits the data into training, validation, and test datasets, builds, validates and tests a model, keeps repeating this process many times in order obtain how robust the validation and testing performance of the models are.

Specifically, for each participant dataset, we randomly generated (sampling with replacement) 70% of training samples and 30% test samples. This was repeated 100 times. Thus, for each dataset, we have 100 sub-groups of a training set and a test set, and therefore test results are averaged over the 100 randomly generated sub-groups of test sets. For classification, original data were split into training data and testing data. The training data set is a sample of data used to fit the model. It contains a known input and output variables and the model learns on this data. We have the test dataset in order to test our model's prediction. In our experiment original dataset contains 60 samples for each participant (30+30=60 records), we used 70% split for training dataset (21+21=42 records) and 30% for testing data (9+9=18 records) in order to train and test the model. This procedure was repeated 100 times and average classification accuracy performance across 100 times (average testing accuracy) was calculated for each participant. These steps are summarized with a block diagram in Figure 2.14.

Figure 2.14

Block diagram of bootstrapping steps



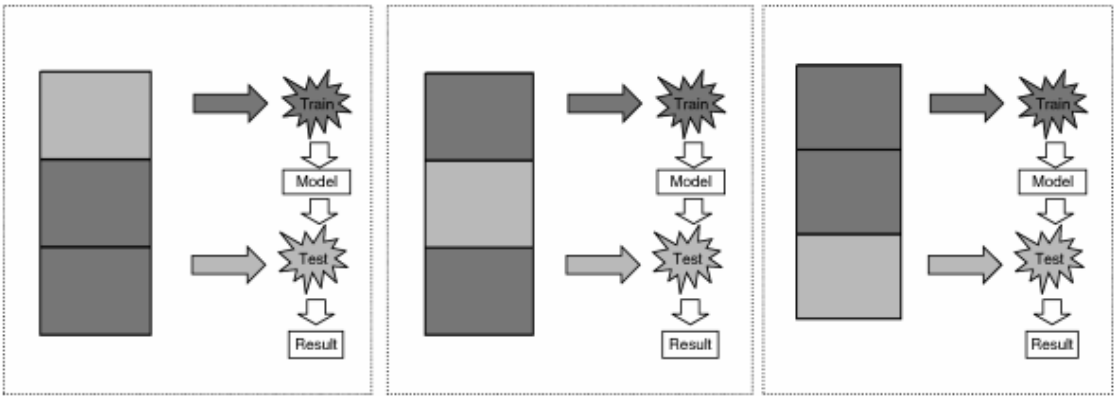
2.7 Cross-validation

Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model (i.e. training) and the other segment is used to validate the model (i.e. validation). In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k-fold cross-validation, which is described in the next paragraph.

In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or "folds". Subsequently, k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning. Figure 2.15 demonstrates an example with k = 3. The darker section of the data is used for training while the lighter sections are used for validation.

Figure 2.15

Block diagram describes 3 fold cross-validation



Cross-validation is a method applied to a model and a dataset in an effort to estimate the error due to sampling. Different methodologies such as averaging can be used to obtain an aggregate measure from all of the different sampling results from all of these different "folds". The average value for each of the 3 cross-validation folds are calculated as

$$\text{Accuracy value} = \text{Average (Fold1 + Fold2 +Fold3)} \quad (1)$$

In this thesis to improve the effectiveness of model performance, we used 3-fold cross-validation. In this 3-fold we split the training data (42 records) into three equal subsets (14 records). Two subsets (28 records) were used for the train the model and one subset used for validation purpose (14 records). We repeated this cross-validation process for 100 different datasets (bootstrapping method) which leads to higher variation in validating model effectiveness and hence the overall model bias was low.

2.8 Machine Learning Algorithms

Statistical classification is defined as the categorization of the data into predefined classes. It is a supervised machine learning methodology. In this methodology, a known set of input and responses are used to build a model. This model is then used to generate the responses of the test data. There are many types of classification algorithms; we used some of the widely used ones, which are listed below.

Support Vector Machines:

Support vector machines (SVM) are statistical supervised machine learning models. They are widely used for classification problems. It is discriminative classifier formally defined by a separating hyper plane. We used this algorithm in this thesis work with the parameters kernel as 'linear'. Table 2.1 listed all the parameters which were used in this thesis work.

Logistic Regression:

Logistic regression (LR) is a predictive analysis method. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables. We used this algorithm in this work with the parameters listed in Table 2.1.

K nearest neighbors:

K nearest neighbors (KNN) is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). Mostly KNN [14] has been used in statistical estimation and pattern recognition. We used this algorithm in this work with the parameters listed in Table 2.1.

Decision Trees:

Decision trees (DT) are one of the more widely used and simple supervised classification algorithms. The tree is defined by the root node, internal node and leaf node. The leaf nodes are the class identifiers and root nodes and internal nodes test the attribute conditions. A decision tree model is constructed by the training set of data. The constructed model is then tested by the test data on how well the classification is done. We used this algorithm in this work with the parameters listed in Table 2.1.

Perceptron:

Perceptron is a widely used algorithm for supervised learning of binary classifier. This algorithm requires input parameters as vector numbers for best prediction. Preceptron is a linear classifier, which means the classification algorithm that makes its predictions based on a linear predictor function. We used this algorithm in our thesis work by initializing parameters, such as the number of iterations = 10. Table 2.1 listed all the parameters which were used in this thesis work.

Gaussian Process:

Gaussian process is a statistical process of random variables by time or space collection. This algorithm is mainly used for multivariate normal distribution i.e. for every linear combination data should be normally distributed. In machine-learning algorithm Gaussian process involves the measurement of the similarity between input datasets by using a kernel function in order to predict the output value for an unseen dataset from testing data. We used this algorithm in this work with the parameters listed in Table 2.1.

Random Forest:

Random decision forest algorithm is used ensemble learning method for classification problem. It constructs a multitude of decision trees on training datasets. In order to classify output prediction it applies on the mode of the classes or mean prediction of individual trees. A random forest model generally mitigates any over-fitting problems. We used this algorithm in this work with the parameters listed in Table 2.1.

AdaBoost:

Adaptive Boosting (AdaBoost) is an ensemble classifier, i.e. the algorithm is made up of multiple classifier algorithms whose output is combined result of the output of those classifier algorithms. This algorithm works iteratively by choosing the training set based on the accuracy of previous training. We used this algorithm in this work with the parameters listed in Table 2.1.

Naïve Bayes:

Naïve Bayes (NB) is a simple probabilistic classifier. This algorithm is based on the assumption of independence among the predictors. NB classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. NB model is easy to build and particularly useful for very large data sets. This algorithm

is easy and fast to predict the class of test data set and also perform well in multi-class prediction. We used this algorithm in this work with the parameters listed in Table 2.1.

Quadratic discriminant analysis:

Quadratic discriminant analysis (QDA) is used to separate measurements of two or more classes of objects by a quadric surface. QDA is a generalization of linear discriminant analysis (LDA), where it is assumed that the measurements from each class are normally distributed. The hypothesis is calculated by measurement of the given class is the likelihood ratio test. This QDA algorithm is the most commonly used method for obtaining a classifier. We used this algorithm in this work with the parameters listed in Table 2.1.

Table 2.1

Machine learning Algorithms and parameters used in classification

Algorithm	Parameters
Perceptron	n_iter = 10 (The number of passes over the training data (aka epochs).)
Logistic Regression	C=4, C is inverse of regularization strength
Nearest Neighbors	n_neighbors = 3 (Number of neighbors)
Linear SVM	kernel="linear", C=4
RBF SVM	C=4 , C is Inverse of regularization strength
Gaussian Process	1 * RBF(0.5)
Decision Tree	max_depth=5 (The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples)
Random Forest	max_depth=5, n_estimators=10, max_features=1
AdaBoost	base_estimator=LogisticRegression(C=4)
Naive Bayes	no parameters, probabilistic
QDA	tol=1e-11

After the feature generation and selection step, different commonly used feature extraction and classification algorithms were applied to the preprocessed EEG data to classify the 60 different epochs (time periods) into "positive" or "negative" categories. The total accuracy of the classification algorithms (i.e. the combined sensitivity and specificity) were calculated according to the following formula, given in Equation 2:

$$ACC = (TP+TN) / (P+N) \quad (2)$$

In Equation 2, P represents the number of positive instances, N represents the number of negative instances, and TP represents the number of "true positives," and TN represents the number of "true negatives". In our case, $P = N = 30$ for each participant. Average classification accuracy values across bootstrapping testing results were calculated with different classification algorithms we employed in this thesis. The results are reported in the next chapter.

CHAPTER 3: RESULTS

3.1 Classification results for all 32 channels using WEKA, 0.5-45Hz filter

We have applied 3-fold cross-validation on EEGLAB preprocessed data for all 12 participants. Table 3.1 summaries the classification accuracy results of different classification algorithms we used in WEKA, with the filtering range of 0.5Hz-45Hz. Each individual participant different features selection to classification results. Among the different classification algorithms, the decision trees algorithm resulted in the highest average classification accuracy, 70.7%, on the average (across the participants). The highest classification accuracy for any participant was 86.7% when the Naïve Bayes algorithm was used.

Table 3.1

Average and individual classification results for all 12 participants with filter frequency 0.5Hz to 45Hz for EEG preprocessing method

Subject	Naïve Bayes	Logistic Regression	SVM	Simple Logistic	Random Forest	Decision Trees	Gaussian process	Nearest Neighbors
1	75.0	66.7	65.0	68.3	65.0	71.7	68.3	71.7
2	66.7	73.3	70.0	71.7	70.0	70.0	60.0	60.0
3	65.0	61.7	61.7	70.0	61.7	76.7	70.0	73.3
4	70.0	71.7	71.7	68.3	75.0	78.3	73.3	63.3
5	71.7	70.0	71.7	70.0	63.3	71.7	65.0	50.0
6	45.0	51.7	55.0	40.0	48.3	50.0	50.0	38.3
7	86.7	78.3	76.7	80.0	78.3	83.3	75.0	48.3
8	70.0	66.7	70.0	68.3	61.7	55.0	55.0	46.7
9	76.7	73.3	71.7	68.3	73.3	73.3	66.7	55.0
10	76.7	83.3	81.7	81.7	85.0	76.7	70.0	71.7
11	75.0	70.0	68.3	68.3	63.3	75.0	71.7	56.7
12	60.0	56.7	58.3	55.0	50.0	66.7	55.0	73.3
Avg	69.9	68.6	68.5	67.5	66.2	70.7	65.2	59.1

3.2 Classification results for all 32 channels using Python, 0.5-45Hz filter

We followed 3-fold cross-validation on Python preprocessed data for all 12 participants From Table 3.2 we can refer to results of different classification algorithms. On comparing with different classification results from the table Gaussian process algorithm showed 63.9% classification on Python processed data. The highest classification accuracy for a participant was 90% in the Random Forest algorithm.

On comparison of 3.1 section results with 3.2 sections, we observed accuracy results were a little bit different. We used two different software's in preprocessing 3.1 section data was preprocessed by EEG toolbox all parameters was initialized by the tool itself. In Python preprocessing was done manually, we built a script to handle negative and positive epochs and to initialize filter parameters. These differences may have caused difference in results.

Table 3.2

Average and individual classification results for all 12 participants with filter frequency 0.5Hz to 45Hz for Python preprocessing method

Subject	Naïve Bayes	Logistic Regression	SVM	Simple Logistic	Random Forest	Decision Trees	Gaussian process	Nearest Neighbors
1	55.0	51.7	48.3	51.7	51.7	53.3	61.7	53.3
2	68.3	56.7	56.7	65.0	73.3	53.3	71.7	68.3
3	61.7	71.7	73.3	71.7	63.3	73.3	75.0	71.7
4	60.0	52.3	53.3	55.0	70.0	55.0	70.0	55.0
5	58.3	51.7	57.3	54.3	57.3	57.7	50.0	55.0
6	60.0	61.7	65.0	68.3	75.0	73.3	65.0	63.3
7	85.0	76.7	85.0	86.7	90.0	81.7	83.3	76.7
8	53.3	51.7	58.3	50.0	56.7	55.0	50.0	51.7
9	66.7	58.3	56.7	56.7	70.0	56.7	58.3	66.7
10	51.7	53.3	55.0	53.3	53.3	60.0	63.3	51.7
11	55.0	50.0	53.3	58.3	53.3	58.3	56.7	51.7
12	53.3	53.3	56.7	53.3	51.7	61.7	61.7	60.0
Avg	60.7	57.4	59.9	60.4	63.8	61.6	63.9	60.4

3.3 Classification results for all 32 channels using WEKA, 0.5-25Hz filter

We have applied 3-fold Cross-validation on EEGLAB preprocessed data for all 12 participants, From Table 3.3 we can refer to results of different classification algorithms. Each individual participant different features selection to classification results. On comparing the different classification results from the table decision trees algorithm [13] showed a 71.2% classification result. The highest classification accuracy for a participant was 80% in Naïve Bayes algorithm.

On comparing classification result sets of 0.5-45Hz filter data, 0.5-25Hz filter data we observed the model performance of filter range 0.5-25Hz data input was significantly high to 0.5-45Hz data model accuracy. Because with a higher pass-band frequency we were including more noise such as movement artifacts and other sources of physiological and electrical noise, which may have affected model training and testing performance and hence the classification performance.

Table 3.3

Average and individual classification results for all 12 participants with filter frequency 0.5 Hz to 25 Hz for WEKA preprocessing method

Subject	Naïve Bayes	Logistic Regression	SVM	Simple Logistic	Random Forest	Decision Trees	Gaussian process	Nearest Neighbors
1	75.0	71.7	71.7	68.3	76.7	65.0	68.3	55.0
2	71.7	66.7	70.0	66.7	66.7	75.0	71.7	55.0
3	71.7	71.7	68.3	73.3	70.0	66.7	53.3	65.0
4	75.0	78.3	80.0	76.7	75.0	80.0	78.3	65.0
5	51.7	66.7	61.7	60.0	63.3	76.7	70.0	61.7
6	63.3	58.3	65.0	55.0	61.7	66.7	68.3	53.3
7	60.0	63.3	56.7	65.0	56.7	66.7	68.3	58.3
8	80.0	70.0	75.0	78.3	78.3	76.7	60.0	66.7
9	53.3	51.7	51.7	41.7	56.7	66.7	68.0	59.0
10	68.3	68.3	68.3	70.0	73.3	58.3	73.3	70.0
11	78.3	66.7	66.7	71.7	66.7	80.0	70.0	58.3
12	53.3	60.0	65.0	65.0	56.7	75.0	68.0	61.0
Avg	66.8	66.1	66.6	65.9	66.8	71.2	68.1	60.8

3.4 Classification results for all 32 channels using Python, 0.5-25Hz filter

We followed 3-fold cross-validation on Python preprocessed data for all 12 participants From Table 3.2 we can refer to results of different classification algorithms. Summaries the classification accuracy results of different classification algorithms we used in WEKA, with the filtering range of 0.5Hz-25Hz. On comparing with different classification results from the table random forest algorithm showed 63.3% classification on Python processed data. The highest classification accuracy for a participant was 90% in the Random Forest algorithm.

Table 3.4

Average and individual classification results for all 12 participants with filter frequency 0.5Hz to 25Hz for Python preprocessing method

Subject	Naïve Bayes	Logistic Regression	SVM	Simple Logistic	Random Forest	Decision Trees	Gaussian process	Nearest Neighbors
1	54.7	56.7	50.0	52.7	59.3	50.0	52.7	59.3
2	68.3	52.7	70.0	58.3	71.7	58.3	66.7	70.0
3	65.0	63.3	66.7	61.7	66.7	70.0	65.0	58.3
4	61.7	61.7	50.0	55.0	58.3	63.3	60.0	56.7
5	60.0	50.0	51.7	50.0	68.3	60.0	55.0	68.3
6	53.3	50.0	66.7	61.7	75.0	75.0	65.0	76.7
7	83.3	76.7	80.0	81.7	85.0	68.3	81.7	78.3
8	55.0	54.0	59.0	50.0	61.7	50.0	50.0	61.7
9	53.3	77.7	57.7	60.0	53.3	50.0	67.0	53.3
10	55.0	50.0	50.0	51.7	53.3	50.0	54.3	53.3
11	55.0	51.7	52.3	50.0	51.7	56.7	54.3	51.7
12	58.3	52.3	52.7	50.0	55.0	65.0	51.7	55.5
Avg	60.2	58.1	58.9	56.9	63.3	59.7	60.3	61.9

3.5 Classification results for 23 channels using Python, 0.5-45Hz filter

We applied 3-fold cross-validation on functional connectivity data for all 12 participants, From Table 3.5a and Table 3.5b we can refer to results of different classification algorithms. Summaries the classification accuracy results of different classification algorithms we used in Python, with the filtering range of 0.5Hz-45Hz Each individual participant different features selection to classification results. On comparing the different classification average results from the table Linear SVM [15] showed 77.6% classification result.

Table 3.5a

Average and individual classification results for all 12 participants with filter frequency 0.5 Hz to 45 Hz for EEG preprocessing method

	Perceptron			Logistic Regression			Nearest Neighbors			Linear SVM			RBF SVM		
Subjects	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
1	88.8	38.8	65.1	100.0	33.3	74.1	83.3	33.3	60.3	94.4	33.3	75.5	100.0	50.0	73.8
2	94.4	50.0	70.6	100.0	61.1	78.3	100.0	55.5	76.5	94.4	61.1	80.2	88.8	55.5	75.6
3	100.0	66.7	83.6	100.0	66.6	83.6	88.8	55.5	73.4	100.0	66.6	84.8	100.0	61.1	81.6
4	100.0	50.0	71.0	100.0	55.5	78.3	88.8	55.5	70.0	100.0	61.1	78.6	100.0	55.5	77.0
5	94.4	50.0	70.7	94.4	61.1	80.9	88.8	33.3	67.4	100.0	61.1	82.6	94.4	55.5	78.7
6	94.4	50.0	69.1	94.4	55.5	79.5	88.8	44.4	71.8	94.4	55.5	78.2	94.4	61.1	78.3
7	94.4	50.0	70.8	94.4	50.0	75.4	94.4	50.0	69.0	94.4	55.5	75.8	88.8	33.3	73.2
8	94.4	50.0	67.0	94.4	55.5	72.4	88.8	50.0	70.9	88.8	50.0	70.9	88.8	50.0	70.5
9	100.0	50.0	77.7	100.0	61.1	80.4	88.8	44.4	68.3	100.0	61.1	82.0	94.4	61.1	78.4
10	100.0	50.0	73.1	100.0	55.5	80.3	88.8	33.3	63.7	100.0	61.1	78.8	94.4	55.5	78.1
11	88.8	50.0	64.8	88.8	50.0	70.4	83.3	33.3	60.3	88.8	50.0	70.2	83.3	50.0	68.6
12	94.4	50.0	65.5	94.4	44.4	72.5	83.3	44.4	65.6	94.4	44.4	73.2	94.4	44.4	71.6
Avg	**95.3**	**50.5**	**70.7**	**96.7**	**54.1**	**77.2**	**88.8**	**44.4**	**68.1**	**95.8**	**55.1**	**77.6**	**93.5**	**52.8**	**75.5**

Table 3.5b

Average and individual classification results for all 12 participants with filter frequency 0.5 Hz to 45 Hz for EEG preprocessing method (Table 3.5a continued)

	Gaussian Process			Decision Tree			Random Forest			AdaBoost			Naive Bayes			QDA		
Subjects	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
1	88.8	33.3	72.2	83.3	33.3	57.6	88.8	33.3	63.5	100.0	38.8	74.1	94.4	27.7	74.7	77.7	27.7	53.7
2	94.4	55.5	77.1	83.3	33.3	57.8	88.8	44.4	70.3	94.4	61.1	78.7	94.4	33.3	67.7	83.3	38.8	64.9
3	100.0	66.6	82.2	88.8	38.8	62.1	88.8	50.0	71.3	100.0	66.6	83.6	100.0	55.5	79.2	94.4	44.4	69.8
4	100.0	61.6	78.3	77.7	33.3	57.6	94.4	38.8	65.5	100.0	55.5	77.9	100.0	44.4	74.5	94.4	38.8	66.5
5	94.4	55.5	79.2	83.3	27.7	57.1	88.8	38.8	66.7	94.4	61.1	80.4	94.4	38.8	72.2	88.8	38.8	63.3
6	100.0	50.0	78.7	83.3	33.3	60.1	100.0	44.4	71.4	94.4	55.5	79.7	94.4	50.0	74.8	83.3	38.8	59.9
7	94.4	55.5	76.8	77.7	33.3	60.1	94.4	38.8	66.6	94.4	50.0	75.5	88.8	38.8	70.4	83.3	33.3	63.3
8	94.4	50.0	74.2	83.3	33.3	63.2	88.8	38.8	66.8	94.4	55.5	73.1	88.8	50.0	73.8	83.3	27.7	63.2
9	100.0	61.0	79.9	77.7	33.3	59.1	88.8	38.8	66.7	100.0	61.1	79.8	88.8	44.4	70.6	94.4	50.0	71.6
10	100.0	55.5	78.7	83.3	27.7	54.2	88.8	33.3	63.0	100.0	61.1	80.5	88.8	50.0	72.2	83.3	27.7	58.3
11	88.8	44.4	69.8	88.8	27.7	55.9	88.8	27.7	61.4	88.8	50.0	70.3	88.8	38.8	70.7	83.3	38.8	63.2
12	94.4	50.0	73.3	88.8	33.3	60.7	88.8	27.7	68.3	94.4	44.4	72.3	94.4	55.5	76.5	83.3	27.7	62.3
Avg	**95.8**	**53.2**	**76.7**	**83.3**	**32.4**	**58.8**	**90.7**	**37.9**	**66.8**	**96.3**	**55.1**	**77.2**	**93.0**	**43.9**	**73.1**	**86.1**	**36.0**	**63.3**

3.6 Classification results for 23 channels using Python, 0.5-25Hz filter

We applied 3-fold cross-validation on functional connectivity data for all 12 participants, From Table 3.6a and Table 3.6b we can refer to results of different classification algorithms. Summaries the classification accuracy results of different classification algorithms we used in Python, with the filtering range of 0.5Hz-25Hz. Each individual participant different features selection to classification results. Each individual participant different features selection to classification results. On comparing the different classification average results from the table Linear SVM showed 71.9% classification result.

Table 3.6a

Average and individual classification results for all 12 participants with filter frequency 0.5 Hz to 25 Hz for EEG preprocessing method

	Perceptron			Logistic Regression			Nearest Neighbors			Linear SVM			RBF SVM		
Subjects	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
1	88.0	44.0	68.7	94.4	61.1	78.5	94.0	44.0	71.1	94.0	55.0	77.3	94.0	55.0	75.2
2	100.0	50.0	75.6	100.0	61.0	82.1	94.0	44.0	70.4	100.0	61.0	84.1	100.0	61.0	79.1
3	94.0	44.0	65.3	88.0	55.0	72.8	83.0	44.0	60.7	100.0	55.0	75.1	94.0	44.0	72.5
4	94.0	44.0	71.4	94.0	55.0	77.9	94.0	44.0	72.9	100.0	55.0	77.2	100.0	55.0	77.0
5	100.0	44.0	68.9	94.0	55.0	75.6	94.0	44.0	74.2	94.0	50.0	75.7	94.0	50.0	75.8
6	83.0	44.0	61.3	88.0	38.8	65.6	88.0	38.8	63.7	88.0	44.0	67.6	83.3	38.8	62.9
7	83.0	27.0	57.1	83.0	27.0	63.9	77.0	33.0	52.5	83.0	33.0	63.5	83.0	27.0	63.1
8	88.0	33.0	57.9	83.0	38.0	64.5	77.0	27.0	54.3	83.0	38.0	64.4	88.0	33.0	62.4
9	94.0	44.0	66.9	88.0	50.0	72.3	88.0	44.0	67.8	94.0	50.0	72.8	88.0	44.0	69.6
10	88.0	44.0	68.1	100.0	50.0	77.1	94.0	50.0	71.0	100.0	50.0	78.5	100.0	50.0	76.6
11	94.0	44.0	57.7	88.0	33.0	62.3	77.0	33.0	55.8	88.0	33.0	62.2	77.0	38.8	61.1
12	77.0	33.0	54.7	88.0	44.0	63.9	72.0	27.0	48.8	88.0	44.0	65.2	88.0	38.0	62.8
Avg	**90.2**	**41.2**	**64.5**	**90.7**	**47.3**	**71.4**	**86.0**	**39.4**	**63.6**	**92.7**	**47.3**	**71.9**	**90.8**	**44.5**	**69.8**

Table 3.6b

Average and individual classification results for all 12 participants with filter frequency 0.5 Hz to 25 Hz for EEG preprocessing method (Table 3.6a continued)

	Gaussian Process			Decision Tree			Random Forest			AdaBoost			Naive Bayes			QDA		
Subjects	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
1	94.0	61.0	77.1	83.0	33.0	59.4	88.0	33.0	68.5	94.0	61.0	78.1	94.0	44.0	73.1	83.0	38.0	65.7
2	100.0	61.0	79.3	77.0	22.0	56.6	94.0	38.0	63.4	100.0	55.0	78.2	94.0	55.0	78.2	83.0	44.0	65.5
3	88.0	38.0	68.1	72.0	27.0	50.2	77.0	27.0	56.2	94.0	50.0	71.6	88.0	44.0	67.7	77.0	27.0	53.0
4	94.0	55.0	77.0	83.0	22.0	57.3	94.0	38.0	65.1	94.0	55.0	77.8	94.0	50.0	77.8	88.0	38.0	63.4
5	88.0	55.0	74.7	83.0	27.0	54.6	88.0	33.0	63.6	94.0	55.0	76.0	94.0	44.0	71.7	88.0	44.0	64.5
6	77.0	38.8	61.3	83.3	33.3	59.1	88.8	38.8	61.6	88.8	39.0	65.2	88.8	38.8	65.6	77.7	38.8	58.2
7	77.0	27.0	59.7	77.0	27.0	51.5	77.0	27.0	56.5	83.0	27.0	63.9	83.0	38.0	63.5	77.0	33.0	52.5
8	77.0	38.8	57.1	77.0	22.0	53.9	77.0	33.0	56.2	83.0	33.0	64.5	83.0	44.0	63.7	83.0	27.0	55.3
9	88.0	50.0	70.8	83.0	33.0	61.1	88.0	38.0	64.5	88.0	50.0	72.2	88.0	50.0	70.3	77.0	22.0	56.6
10	88.0	44.0	73.2	77.0	22.0	53.6	88.0	38.8	61.3	100.0	50.0	76.6	100.0	38.8	70.9	88.0	44.0	62.1
11	77.0	33.0	59.6	83.0	38.0	57.4	88.0	33.0	60.5	88.0	33.0	62.5	88.0	38.0	64.7	83.0	33.0	54.4
12	72.0	27.0	54.1	72.0	27.0	52.8	94.0	27.0	55.7	88.0	44.0	64.7	83.0	38.0	62.8	77.0	16.6	47.6
Avg	**85.0**	**44.1**	**67.7**	**79.2**	**27.8**	**55.6**	**86.8**	**33.7**	**61.1**	**91.2**	**42.2**	**70.9**	**89.8**	**39.4**	**69.2**	**81.8**	**33.7**	**58.2**

CHAPTER 4: CONCLUSION AND FUTURE WORK

In this work, classification accuracy with different algorithms was calculated for all 12 participants with all 32 channels and reduced 23 channels of the EEG data. EEG time series data were preprocessed by using EEGLAB and Python for two different frequency filter ranges of 0.5 Hz to 45 Hz, and 0.5 Hz to 25 Hz separately. One main observation from the classification results was that participants had different classification accuracy results when considered all 32 channels data and 23 channels data separately and these "best accuracy" methods differed between the participants.

In this thesis, the data preprocessing technique was implemented using two different methods, to study differences in processing by different software. By using the EEGLAB toolbox method, data was easily handled on implementing filter range and selecting the channels. In Python preprocessing, data was handled manually in the programming script.

Functional connectivity feature extraction was taken place in a MATLAB environment. Features were extracted for both 32 channels and 23 channels separately. By using correlation coefficient method total 497 features, 254 features were extracted for 32 and 23 channels data. These extracted features were used to build a machine learning model. The main novelty of this work was that functional connectivity between EEG channels were computed and used as features to perform classification of what category if images the participants were viewing.

We used two different software's WEKA and Python to handle classification problem. In WEKA tuning parameters were considered by the tool itself whereas in Python manually we installed tuning parameters into the program. We implemented the bootstrapping method and 3 fold cross-validation techniques to improve model

performance. We observed different accuracy results from this software when provided same input data set.

By using WEKA classification tool for 32 channels EEGLAB preprocessed data of applied filter frequency range 0.5 Hz to 45 Hz: Highest average classification accuracy was 70.7% with a combination of decision trees algorithm with train split 70% and 3-fold cross-validation. For 32 channels Python preprocessed data of applied filter frequency range 0.5 Hz to 45 Hz: Highest average classification accuracy was 63.9% with a combination of Gaussian process algorithm with train split 70% and 3-fold cross-validation.

For 32 channels EEGLAB preprocessed data of applied filter frequency range 0.5 Hz to 25 Hz: Highest average classification accuracy was 71.2% achieved with a combination of decision trees algorithm with train split 70% and 3-fold cross-validation using WEKA tool.

By using Python Scikit learn classification for 23 channels EEGLAB preprocessed data of applied filter frequency range 0.5 Hz to 25 Hz: Highest average classification accuracy was 71.9% with a combination of linear SVM algorithm with train split 70% and 3-fold cross-validation. For frequency range 0.5 Hz to 45 Hz data: Highest average classification accuracy was 77.6% with a combination of linear SVM algorithm with train split 70% and 3-fold cross-validation.

Future work will mainly focus on analyzing EEG data using deep learning models. Continued exploration of deep learning algorithms in the classification of EEG gives better guidelines and model performance. A variety of methods can implement in feature extraction from EEG signals such as principal component analysis (PCA), time-frequency distributions (TFD), fast Fourier transform (FFT) and wavelet transform (WT).

There are two main avenues for the extension of this research: application to different datasets, expanded model search. Currently, these techniques were only applied to a single dataset. In order to show generalizability, it is desirable to replicate this research on further datasets with different machine learning tuning properties. The dataset explored in this thesis was collected on a 32-channel dry-contact Bluetooth EEG device. Dry-contact EEG devices have much more noise than the traditional gel-contact EEG devices, so removal of noise was an issue. Also, many modern gel-contact EEGs have 128 or even 256 channels, leading to several times the number of features, and better-quality signal. Thus, it is important to examine feature extraction technique on more features to improve the performance of classification accuracy.

The work presented in this thesis covered on the machine learning classification problem. Implement a Deep Learning algorithm can enable many practical applications of Machine Learning and by extension the overall field of Artificial intelligence.

REFERENCES

[1] W. L. Hosch (2016) “Machine Learning” Britannica [Online]. Available: https://www.britannica.com/technology/machine-learning

[2] R. Kohavi (1998), F. Provost “Glossary of Terms: Special Issue on Applications of Machine Learning and the Knowledge Discovery Process,” robotics.stanford.edu [Online]. Available: http://robotics.stanford.edu/~ronnyk/glossary.html

[3] S. Russell, P Norvig, “Learning from Observations,” Artificial Intelligence, A Modern Approach, 1st ed. Upper Saddle River, Prentice Hall, 2013, ch. 18, sec 1 pp. 525 - 529

[4] MATLAB. “Machine Learning with MATLAB,”Matworks [Online] https://www.mathworks.com/solutions/machinelearning/?requestedDomain=www.mathworks.com#classification

[5] L. F. Hass. “Neurological Stamp,” National Center for Biotechnology Information. [Online] Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1738204/pdf/v074p00009.pdf

[6] Editors of Encyclopedia Britannica. “Electroencephalography,” Britannica [Online]. Available: https://www.britannica.com/science/electroencephalography.

[7] MATLAB (2015) http://mathworks.com/MATLAB.

[8] https://www.cs.waikato.ac.nz/ml/WEKA/downloading.html.

[9] Arnaud Delorme, Scott Makeig (2004), “EEGLAB an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis.” J Neuroscience Methods, Mar 15;134(1):9-21, 92093-0961.

[10] Alpaydin, E. (2010). “Introduction to Machine Learning (2nd Ed.)”. Cambridge, MA, USA: The MIT Press.

[11] Seetaramaraju Jampana, Master's Thesis (2015) "A novel multivariate analysis method to classify different tasks based on rapid electroencephalography (EEG) data." (Adviser: Dr. Unal "Zak" Sakoglu), Computer Science Department, Texas A&M-University at Commerce, Commerce, TX.

[12] Python: https://www.Python.org/

[13] A Master Thesis Presented by Roman Timofeev (2004) "Classification and Regression Trees (CART) Theory and Applications". (Adviser: DrWolfgang H¨ardleCenter of Applied Statistics and Economics, Humboldt University, Berlin.

[14] https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/

[15] J.A.K. Suykens and J. Vandewalle Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT-SISTA Kardinaal Mercierlaan 94, B–3001 Leuven (Heverlee), B "Least Squares Support Vector Machine Classifiers"

[16] Brainworks. "What are brainwaves," Brainworks [Online]. Available: http://www.brainworksneurotherapy.com/what-are-brainwaves.

[17] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

[18] Kevin K Dobbin and Richard M Simon (2011). Optimally splitting cases for training and testing high dimensional classifiers.

[19] A Coenen and O Zayachkivska (2013).A pioneer in electroencephalography in between Richard Caton and Hans Berger.

[20] Dan-Glauser, E. S., & Scherer, K. R. (2011). The Geneva affective picture database (GAPED): a new 730-picture database focusing on valence and normative significance. Behavior research methods, 43(2), 468-477.

[21] Hafeez Ullah Amin, Aamir Saeed Malik (March 2015), Australasian Physical & Engineering Sciences in Medicine, Volume 38, Issue 1, pp 139–149. Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques.