Sukkur Institute of Busiess Administration

Master Thesis

Robust Error Detection During P300 Speller using Ensemble Learning

Author:
Attaullah

Supervisor:
Dr. M. Abdul Rehman

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

in the

Department of Computer Science
Sukkur Institute of Busiess Administration

December 2023

Robust Error Detection During P300 Speller using Ensemble Learning

by

Attaullah

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE
at the

SUKKUR INSTITUTE OF BUSINESS ADMINISTRATION

December 2023

Signature of Author

Attaullah
Department of Computer Science

Certified by:

Internal Examiner Signature

Dr. M Abdul Rehman
Associate Professor

External Examiner Signature

Dr. Tariq Mehmood
Associate Professor
PAF-KIET Karachi

Accepted by:

Dr. Qamar Uddin Khand
MS/Ph.D Coordinator

Exordium

In the name of Allah, the Compassionate, the Merciful.

Praise be to Allah, Lord of Creation, The Compassionate, the Merciful, King of Judgment-day!

Contents

Acknowledgements

I would like to thank my supervisor, Dr. M Abdul Rehman, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly.

Completing this work would have been all the more difficult were it not for the support and friendship provided by the other members of the Department of Computer Science in Sukkur-IBA. I am indebted to them for their help.

List of Figures

List of Tables

Abbreviations

BCI	**B**rain **C**omputer **I**nterface
SVM	**S**uport **V**ector **M**achine
LDA	**L**inear **D**iscriminant **A**nalysis
ERP	**E**vent **R**elated **P**otential
EEG	**E**lectro**e**ncephalo **g**ram
ICA	**I**ndependent **C**omponent **A**nalysis
PCA	**P**rincipal **C**omponent **A**nalysis
MI	**M**otion **I**magery
MEG	**M**agneto**e**ncephalo**g**raphy

Dedication

To my parents who taught me to think clearly & motivated me to try my hardest in everything I do, without whom I could not have reached my goals

and

To all those who believe in the power of learning.

Abstract

Electroencephalogram (EEG) is a measure of electrical activity of brain. A brain-computer interface (BCI) uses EEG signals to provide a non-muscular communication channel for motor-impaired patients. One of the most important EEG paradigm that has been explored in BCI systems is the P300 signal. The P300 component of an ERP is widely used in BCI to translate the subject's intention by thoughts into commands to control electro mechanical devices and artificial human body parts. One of the challenge of P300 signals is low signal-to-noise ratio (SNR). Low SNR is due to other ongoing activities and artifacts of brain. This contaminates true P300 signal with noise. machine learning algorithms can be used to design BCI system efficiently. In order to address this challenge in a proper way, we propose a system consists of three main steps, in first step we preprocess the data. In second step feature extraction is performed and in final step classification is performed. We propose performance efficient method to construct a decision model by using P300 data for classification and feature extraction technique used in our approach is based on spatial filtering using xDAWN, Covariances for handling high dimensional data and tangent space mapping for converting from Riemannian manifold to euclidean space, then considering the elastic net as learning algorithm. Our results show that our proposed system gains better AUC (Are under Curve) as compared to state-of-the-art , decreases time drastically and it performs better during the inter-session and inter-subject variability. Proposed model achieves 0.79 AUC in just 37 seconds as compared to state of the art having AUC 0.80 in 224 seconds.

Chapter 1

Introduction

1.1 Brain Computer Interface

Electroencephalogram (EEG) is a measure of electrical activity of brain. This measured activity can be used in many useful ways: physicians can use for diagnosing neurological disorders, Brain Computer Interface scientists can use for means of communication. Nowadays Brain Computer Interface (BCI) has proved to be very helpful for human beings, such as controlling wheel chairs, spelling devices, video games, entertainment and other assistive technologies [1] . It is more beneficial for those who are paralyzed and lost all means of communications. Other patients may also get benefit from it who have got spinal cord injury or patients with amyotrophic lateral sclerosis (ALS) which is neuro-degenerative disease that can result in loss of voluntary control of their muscles. In such situations BCI can be used as a reliable source of communication or control of some external devices. However, in some patients, disease can progress to a point that will cause "locked-in" syndrome, which is a condition where patient is awake and fully aware but cannot communicate due to complete paralysis. In these cases BCI has a potential to establish

a communication channel directly from patient's brain signals to the computer [2] . In order for BCI system to work properly, identifying a brain signal that the patient can use reliably and voluntarily for control without use of any muscular movements is of utmost importance.

Brain signals used for BCI can be obtained either through invasive or non-invasive methods. Invasive recording method requires implantation of electrodes inside the skull, this involves complex surgical process. Implanted electrodes give accurate results, but Wolpaw et al. [3] has shown that non-invasive method of recording signals can also perform better as compared to that of invasive; avoiding surgical process.

1.2 P300 Speller

P300 is one of the well-studied non-invasive BCI paradigm. P300 response is usually elicited by oddball paradigm in which low probability of desired items are mixed with high probability of undesired items. The P300 response is an event-related-potential (ERP) which is recorded during the process of decision making as a subject reacts to a stimulus. P300 speller [4] has 6X6 matrix of characters containing alphabets and numbers. This paradigm uses visual stimuli, also other type of stimuli exists like auditory stimuli [5]. In this scenario subject wears a cap on scalp containing electrodes as shown here in Figure 1.1. Electrodes are connected to Amplifier (as signals generated by activities of neurons are of low potential), which is eventually connected to computer, where finally data is stored and analyzed. Users have to spell a word character by character shown to him on screen. Each column and row of matrix is highlighted successively at

a constant rate. Subject must focus attention on the letter he wants to type. When the line or the column containing the target letter is flashed, a P300 response is elicited.

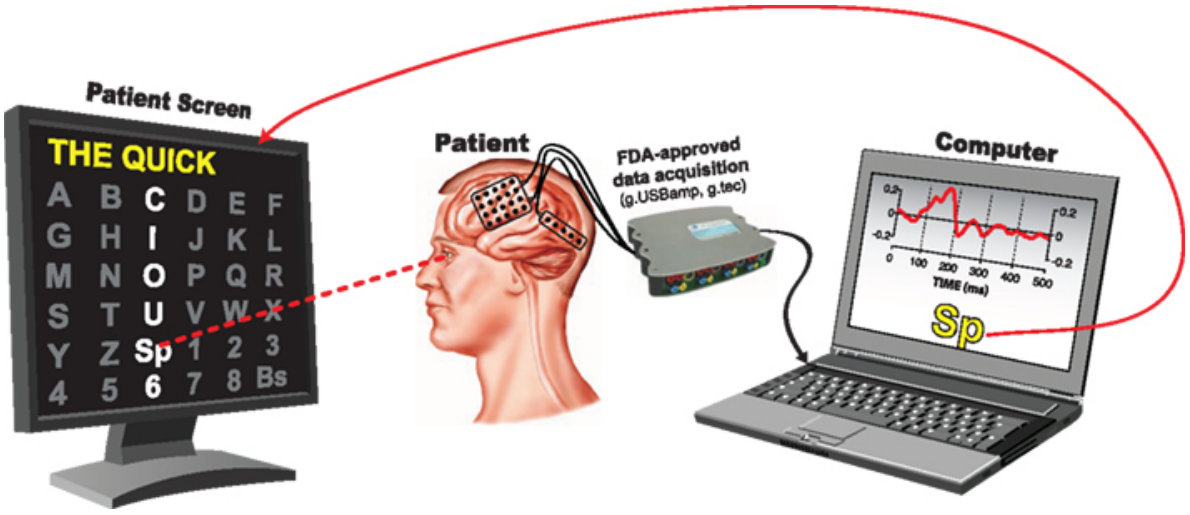

FIGURE 1.1: P300 Speller

For recording EEG, electrodes are placed in a standardized way. This creates easiness in comparing results from different experiments and studies. Figure 1.2 shows location of 64 electrodes as proposed by Pivik et al. [6]. The letters in names of electrodes F, T, C, P and O stand for frontal, temporal, central, parietal, and occipital lobes, respectively. Even numbers refer to electrode positions on the right hemisphere, whereas odd numbers refer to those on the left hemisphere. A "z" (zero) refers to an electrode placed on the mid line. For instance electrode Cz is placed at center of scalp.

1.3 Problem Statement

As humans think, brain waves are produced. These brain waves can be translated to intentions of what humans want to do. These brain waves data can be recorded using EEG. Data set collected through

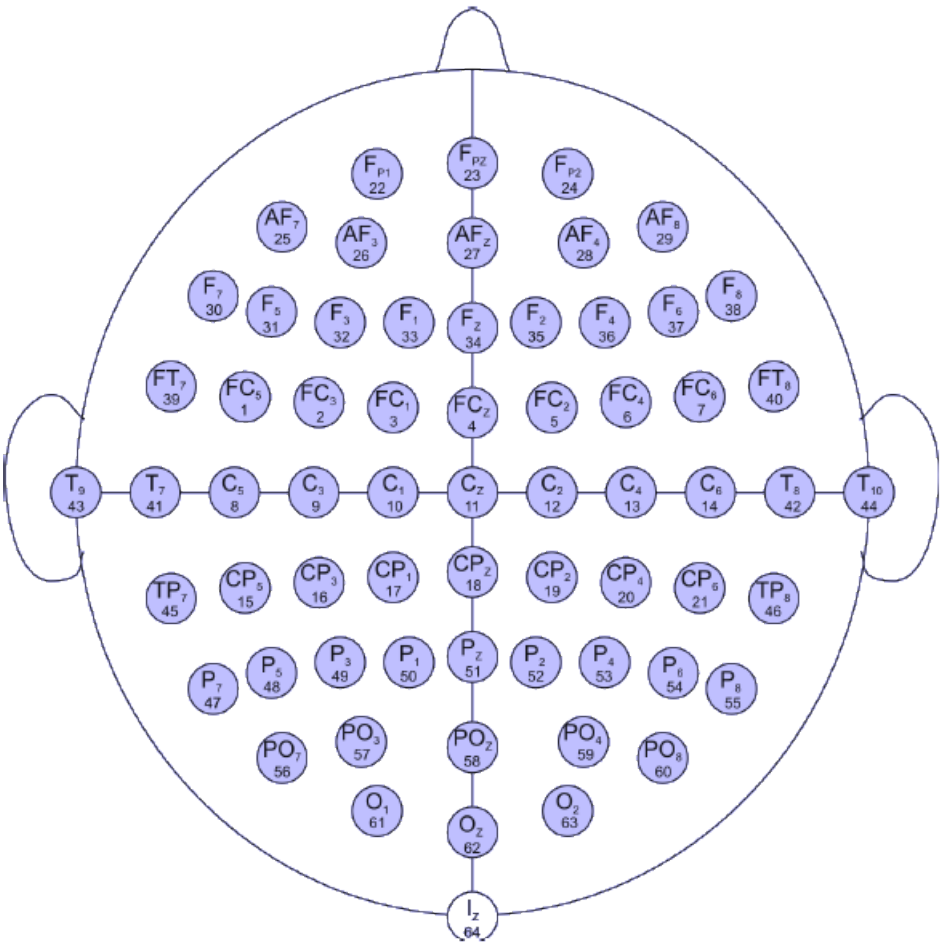

FIGURE 1.2: ELECTRODE PLACEMENT FOR A 64- CHANNEL EEG USING THE INTERNATIONAL SYSTEM [6].

EEG is very noisy and has low SNR (Signal to Noise Ratio). P300 response is hard to differentiate from recorded signal, as it contains background noise, caused by ongoing electrical activity in the brain. This creates challenges of extracting event related potential and interpreting brain activity correctly. Problem is to detect errors during spelling task, given subject's brain waves. In Figure 1.3 two trials, Target and Non target are shown by taking averaged recording from central electrode Cz after occurrence of stimulus and it is really difficult to differentiate between true P300 signal and noisy signal.

Machine Learning algorithms play vital role in designing BCI system. For instance these techniques can be used to extract most important

and relevant features from data set, as raw EEG signals are inherently noisy for various reasons. Feature extraction can be employed to remove noise, artifacts and other irrelevant activities from the raw data. Then a classifier is trained on labeled data and applied to predict unlabeled test data. Classifier tries to predict whether BCI system has identified Target letter correctly or not according to subject's intention. Our goal is to find robust and reliable feature extraction method and classifier for EEG data from P300 speller, which reduces misinterpreted classified commands from user's actual intentions and provide reliable means of communication. In order to address this, we have set two research questions:

1. Does spatial filtering used with covariance matrices for classification of P300 Speller enhance across-subject generalization and across-session generalization, having optimal accuracy?
2. Does linear classifier converge fast, so that the adaptation is efficient and robust in terms of time complexity?

1.4 Contribution

Main contribution of this thesis work is to propose robust method for preprocessing of raw EEG data and its classification in P300 speller paradigm for detection of errors. This work is based on Riemannian geometry. Covariance matrices are computed and concatenated with xDAWN filters. This intermediate results is projected into tangent space, where elastic net algorithm is used for training and testing. Our model achieved 0.79 AUC in just 37 seconds. While stat-of-the-art model by Alexandre [7] achieved AUC 0.80 in 224 seconds.

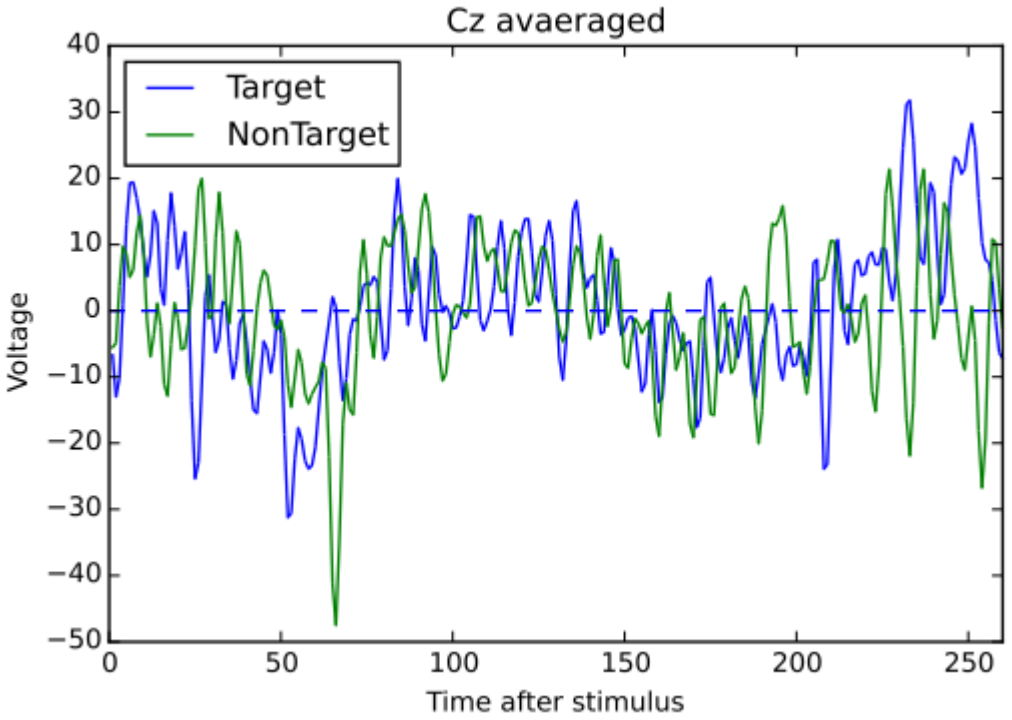

FIGURE 1.3: Averaged Cz electrode for Target and Non-Target

Performs in terms of AUC is almost similar, however our model outperforms in terms of time taken.

1.5 Organization

The thesis is organized as follows: Chapter 2 reviews briefly related work regarding preprocessing and classification techniques for P300 classification. Chapter 3 describes the dataset, also preprocessing steps and feature extraction algorithms mathematically. Chapter 4 explains the results of the experiment and includes figures that show the comparison of performances between our approach with other state-of-the-art model. Chapter 5 concludes this work by providing a more in-depth discussion of the results and opens several avenues for future work.

Chapter 2

Related Work

In this chapter we will discuss different feature extraction and classification techniques used earlier for P300 classification. Some classification methods works better on one data set while does not perform well on other data sets. Feature extraction techniques for P300 data can be categorized into two categories, one based on traditional preprocessing methods like ICA (Independent Component analysis) etc. and other based on Riemannian geometry. Here classification accuracy is not mentioned, since the classification accuracy that each method can achieve changes from one data set to another data set.

2.1 Literature Review

Krusienski, et al. [8] compared the results of different classifier algorithms for P300 classification, which shows that stepwise linear discriminant analysis (SWLDA) and support vector machines (SVMs) perform well compared to the other classifiers. SWLDA works like simple LDA, but initially no features in discrimination function, then adding single feature one by one after checking its statistical significance in terms of p-value. They used only 16 channels and restricted

features to 60. This approach is not scalable, when large number of electrodes are used.

According to Cashero [9], support vector machine (SVM) with a Gaussian kernel performs well in BCI paradigm for P300 classification. They concluded that BSS (Blind Source Separation) which includes independent component analysis (ICA), principal component analysis (PCA) and maximum noise fraction (MNF), and feature selection algorithms can contribute significant performance gains, but there is no added benefit from using both together. Feature selection is most beneficial when applied to a large number of electrodes, and BSS is most beneficial when applied to a smaller set of electrodes. Also, the results show that time-delay embedding is not beneficial for P300 classification. They used data from only 3 subjects, so it may not perform well across subjects and across sessions.

Blankertz et al [10] investigated how to classify ERP data from P300 experiment in a best way. However, they focused on feature selection and used spatial as well as spatio-temporal features, along with LDA with shrinkage as classifier. They compared their results with SWLDA (stepwise linear discriminant analysis) from Krusienski, et al., [8] and simple LDA. Results suggest their model outperforms other two. They also present an analytical method to estimate the optimal regularization parameter for LDA with shrinkage. They used 55 electrodes but only 7 time sample points used, a very small epoch window, which may have missed important temporal information for whole trial.

Farquhar et al [11] investigated interaction between different preprocessing techniques and classification methods. They used five different datasets from BCI paradigm and applied spectral, spatial

and spatio-temporal filtering along with four classification methods namely LDA, SWLDA(stepwise LDA), rLDA (regularized LDA) and rLR (regularized logistic regression). They empirically conclude that regularized classifiers perform better than their counter part. However this pipeline needs lot of parameter tuning and may not work better across sessions and needs re-training.

Congedo et al. [12] introduced MDM (Minimum Distance Mean) classifier based on Riemannian geometry. They used data from all three modalities of BCI including ERP. Data was used from 22 electrodes, which was band pass filtered between 8-30 Hz. They calculated covariance mean matrices for each class from training data and classified test data based on their minimum distances to covariance mean matrix of a particular class. They compared results with CSP+LDA (Common Spatial patterns [13] + Linear Discriminant Analysis) from Lotte et al. [14] and conclude that, MDM works better on low data and also good for across subject and across session generalization. As these covariance matrices belong to riemannian manifold, so we can not apply classical classification algorithms like LDA,SVM, etc which work in euclidean space.

Barachant et al. [15] also extended concept of MDM (Minimum Distance Mean) classifier. They proposed a technique by keeping two objectives: fast adaptation on small data and generalization across different subjects as well as across different sessions. They introduced slightly different method for preparing final trial by concatenating Pi prototyped ERP response, which help catering for temporal structural information. Three different datasets from P-300 based game Brain Invaders [16] were used. They compared results of their proposed MDM (Minimum Distance Mean) with xDAWN [17] and SWLDA [8]. Results were better than other two techniques

in context of across subjects as well as across sessions. But performance degrades when more than 64 channels are used or more time samples are used than the number of channels.

Bertrand et al. [17] proposed xDAWN algorithm for spatial filtering method for enhancing P300 evoked response. Main intuition behind this technique was that P300 response is rare and target synchronous response occupies a small spatial subspace span of the EEG signal. xDAWN algorithm works by QR factorization [18] . Data from 3 subjects using 29 electrodes was used. Results were compared by applying PCA, ICA and xDAWN spatial filtering, and Bayesian Linear Discriminant Analysis (BLDA) classifier. They conclude that xDAWN spatial filters brings significant enhancement as compared with PCA and ICA. But if too many xDAWN components are used, then performance is slightly decreased, this makes algorithm not scalable.

Yger et al. [19] studied the behavior of using different kernels for SVM on classification of covariance matrices from Motor Imagery BCI. They conclude that by applying stein or log Euclidean kernel for SVM gives equivalent results to that of CSP+LDA (Common Spatial patterns + Linear Discriminant Analysis) from Lotte et al. [14]. Data in this experiment was used from Motor imagery, but can be used to other modalities of BCI like ERP. But this method needs kernel trick for SVM and needs care ful selection of hyper-parameters.

Barachant et al. [20, 21] proposed an approach in detail for projecting Riemann manifold based covariance matrices into Euclidean space through a technique called Tangent Space mapping. This technique facilitates usage of classical classification algorithms like LDA,

SVM etc on covariance matrices, instead of Minimum Riemann Distance Mean (MRDM) [12]. Results suggest significant improvement without need of spatial filtering of electrodes. However, these both papers are applied to Motor Imagery (MI), another modality of BCI, but can be applied to classification of P300.

Perrin et al. [22] used data from 16 healthy volunteers. They used only 32 electrode's data, down sampled at 100 Hz and band pass filtered between 1-20 Hz. For feature extraction xDAWN [23] spatial filters were used along with mixture of multidimensional Gaussian model as a classifier. Study was focused on on-line error detection and correction. Their main focus was error correction, after error detection. They achieved 4% increase in spelling accuracy, provided error detection algorithm works better. They used limited number of electrodes and smaller epoch window size which may have missed important spatial as well as temporal information for each trial.

Alexandre et al. [7] presented an approach based on ad hoc filtering and covariance matrices from Riemannian geometry as features for classification algorithms. They also performed electrode selection to reduce curse of dimensionality and added other meta features like session number, feedback id, etc.Bagging model of multiple classifiers was used, which increased overall performance. They achieved first position at Kaggle[1], where AUC (Area Under Curve) was used as evaluation metric.

2.2 State-of-the-art

In literature very nice [7, 12, 15] approaches have been seen over the past few years, these systems use Riemannian based covariance

[1] https://www.kaggle.com/c/inria-bci-challenge/leaderboard

matrices as features for classification and additional meta features such like session number, feedback id, etc were included [7] to enhance the classification accuracy. Although these additional features were major contribution towards the classification problem of P300 Speller, but this inclusion has given birth complexity syndrome of the classifier, therefore, it is direly needed to propose a system which may reduce complexity along with consideration of optimal features towards time accuracy trade-off.

Chapter 3

Methodology

This chapter introduces different algorithms used for feature extraction and classification, then dataset which will be used throughout the experiment are described in detail and how they will be used.

3.1 Preprocessing

For preprocessing of raw EEG signal, EOG channel was removed. EOG (ElectroOculoGram) channel gives information about the noise introduced by eye blinking.Then EEG signals were bandpass filtered by 5^{th} order butter-worth filter between 1-40 Hz. Butter-worth filter is a type of signal filter designed to have flat response in its pass-band. Signals were epoched to take only 1.3 seconds after the occurrence of feedback event. Then following feature extraction pipeline is applied before classification.

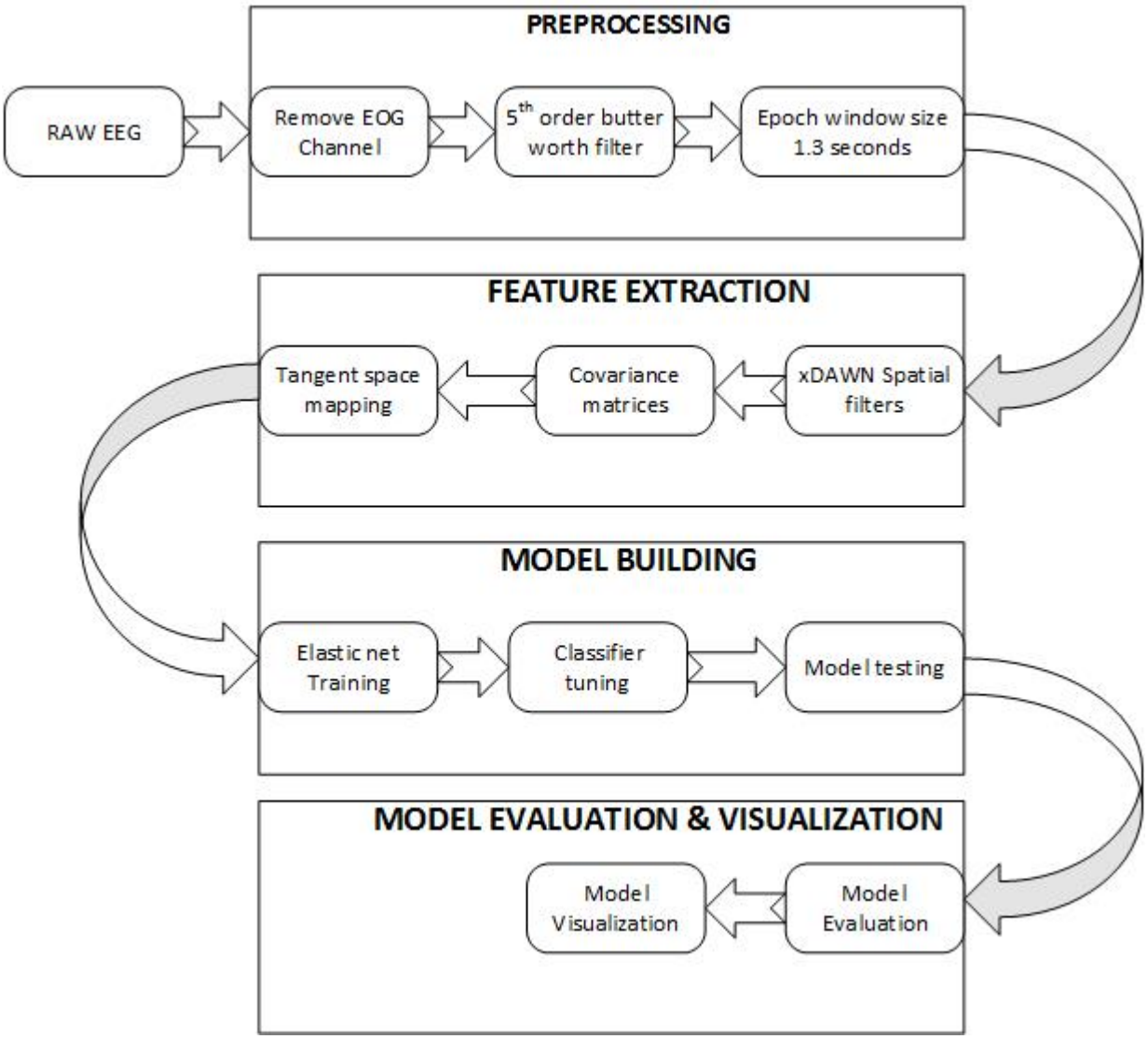

FIGURE 3.1: Details of workflow steps used

3.2 Feature Extraction and Classification

In order to better understand different approaches, that were used in this experiment, are explained in detail mathematically. Our feature extraction algorithms are described based on how they are used prior to classification algorithm. Figure 3.1 gives a big picture of all preprocessing step, feature extraction techniques and model building steps.

3.2.1 xDAWN

As raw EEG signals recorded from the human brain are inherently noisy, it contains the desired P300 evoked signals as well as other ongoing activities of the brain like eye blinking and other muscular artifacts etc. This results in very low SNR (signal to Noise Ratio) and makes classification task very difficult. There are lot of ways to improve evoked EEG signal, one of them is spatial filtering. To enhance SNR, independent component analysis (ICA) was also used [24]. Major drawback with such techniques are that, they are not designed specifically for BCI. Bertrand et al. [17] came up with an unsupervised spatial filtering technique called xDAWN. Main intuition behind this technique was that: P300 event is rare and occupies a small synchronous spatial subspace from the EEG signal. This method works by estimating synchronous responses for each electrode and then estimating spatial filters using these responses, such that the evoked P300 signals are much more distinguished form the noise.

For mathematical formulation , let us assume actual recorded EEG signal is $X \in \mathbb{R}^{N_t \times N_s}$ represents recorded EEG signals where $(i,j)^{th}$ value is $x_j(i)$, which corresponds to j^{th} electrode at time index i and let. N_s is the number of electrodes (here we have total 56 electrodes) and N_t is the number of time points. Let $a_j(t)$ represent the ERP signal for the j^{th} electrode at time index t, and let $A \in \mathbb{R}^{N_e \times N_s}$ represents the matrix of ERP signals whose $(i,j)^{th}$ value is at $a_j(i)$. Where N_s is the number of electrodes and N_e is the number of time points of the ERP (here, N_e is chosen 260 which corresponds to 1.3 second at 200 Hz [200$*$1.3 = 260]). Raw EEG signal can be described

as ERP A and Noise N:

$$X = DA + N \tag{3.1}$$

$D \in \mathbb{R}^{N_e \times N_s}$ is the Toeplitz matrix having first column $D_{i,1} = 1$, where i is the stimulus onset of the target stimulus. So DA represents synchronous response with target stimuli and N is the ongoing activity of the brain and other artifacts i.e. noise. Response of A can be estimated using least square estimation:

$$\hat{A} = arg\min_{A} |X - DA|_2^2$$

Solution is given by:

$$\hat{A} = (D^T D)^{-1} D^T X \tag{3.2}$$

For estimation of N_f spatial filters u_i:

$$XU = DAU + NU \tag{3.3}$$

Where $U \in \mathbb{R}^{N_s \times N_f}$ are the spatial filters , whose i^{th} column is u_i filters. For maximizing signal to signal plus noise ratio, U spatial filters are given by:

$$\hat{U} = arg\max_{U} \frac{Tr(U^T \hat{A}^T D^T D \hat{A} U)}{Tr(U^T X^T X U)} \tag{3.4}$$

Where $Tr(.)$ is the trace operator used for matrices. By computing QR factorization [18] of X and D , equation 3.4 can be written as:

$$\hat{V} = arg\max_{V} \frac{Tr(V^T Q_x^T Q_D Q_D^T Q_X V)}{Tr(V^T V)} \tag{3.5}$$

Where $V = R_x U$, $X = Q_x R_x$, $D = Q_D R_D$ and Q_x, Q_D are orthogonal matrices and R_x, R_D are upper triangular matrices. So matrix

$\hat{V}$ is obtained by concatenation of N_f eigen vectors and associated with N_f largest eigen values of matrix $Q_x^T Q_D Q_D^T Q_X$ [18]. These vectors can be computed using singular value decomposition (SVD) of $Q_D^T Q_X$:

$$Q_D^T D_x = \Phi \Lambda \Psi^T \tag{3.6}$$

Where$\Lambda \in R^{N_s x N_s}$is diagonal matrix of singular values stored in descending order.$\Phi \in R^{N_e x N_s}$ and $\Psi \in R^{N_s x N_s}$ are two column orthogonal matrices. These matrices can be Split into signal and noise as:

$$\Phi = \begin{bmatrix} \Phi_s & \Phi_n \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_n \end{bmatrix}$$

$$\Psi = \begin{bmatrix} \Psi_s & \Psi_n \end{bmatrix}$$

This leads to :

$$\hat{V} = \Psi_s$$

Solution of equation 3.4 is given by:

$$\hat{U} = R_x^{-1} \Psi_s \tag{3.7}$$

Finally we can rewrite equation 3.2 as:

$$\hat{A} = R_D^{-1} \Phi_s \Lambda_s \Psi_s^T R_X + R_D^{-1} \Phi_n \Lambda_n \Psi_n^T R_X \tag{3.8}$$

Now equation 3.1 can be written as:

$$X = D\acute{A}W^T + \acute{N} \tag{3.9}$$

Where

$$\acute{A} = R_D^{-1}\Phi_s\Lambda_s \tag{3.10}$$

$$W = R_x^T\Psi_s \tag{3.11}$$

Here $\acute{A}$ is synchronous response and $\acute{N}$ is the noise term. Final I dimensional evoked space is given by $(\hat{u}_i, \acute{a}_i)$ defined by equation 3.8 and equation 3.10.

3.2.2 Covariances

Spatial filtering can enhance SNR for better design of BCI, but it needs substantial amount of training data, which in case of BCI will require more repeated trials from the subject and are computationally expensive, this makes them bad choice for on-line scenario. As Congedo at al. [12] used covariance matrices for feature extraction, which was latter enhanced by Barachant et al. [15] to include temporal information. This all has taken benefits from recent theoretical advancements in field of Information geometry by Amari [25]. These theoretical foundations has made possible for its applications in different fields. Processing covariance matrices in their manifold (Manifold: a topological space that shows resemblance to euclidean space near each point.) has lot of advantages. It has been widely used for radar signal processing [26], diffusion tensor imaging [27] and image processing [28]. The information geometry is a field of information theory where the probability distributions are taken as points of a Riemannian manifold.

Let us define $P1 \in \mathbb{R}^{s\times t}$ the prototyped ERP response also known as sample covariance matrix (SCM), is obtained for each class, by

averaging all of its epoch, such as:

$$P_1 = \frac{1}{I} \sum_{i \in I} X_i \tag{3.12}$$

Where I is the indexes of trails. For each trial x_i, modified $\tilde{x_i}$ is given by combining:

$$\tilde{x_i} = \begin{bmatrix} P1 \\ X_i \end{bmatrix}$$

Final covariance matrices are computed using Sample Covariance Matrix (SCM):

$$\tilde{\sum} i = \frac{1}{N-1} \tilde{X}_i \tilde{X}_i{}^T \tag{3.13}$$

The resultant covariance matrix of each epoch is then used in next step. After concatenating spatial filters. Covariance matrices are calculated using pyriemann[1] library.

On covariance matrices we can directly apply Minimum distances to Mean (MDM) classifier [12, 15] as they all are from Riemannian fold of symmetric positive definite matrices (SPD) [29].

3.2.3 Tangent Space

As many popular and efficient algorithms can not be directly applied to covariance matrices of EEG data, because they belong to riemannian manifold. We can use SVM, but still need kernel trick as done in [21], but that is not obvious choice. Tangent space mapping is a technique that facilitates use of robust and state-of-the-art classifiers like Logistic regression , elastic net etc. in riemannian framework. Equivalent euclidean space vectors are computed using tangent space

[1] https://github.com/alexandrebarachant/pyRiemann

mapping on covariance matrices belonging to riemannian manifold. Projection of manifold is summarized in Figure 3.2.

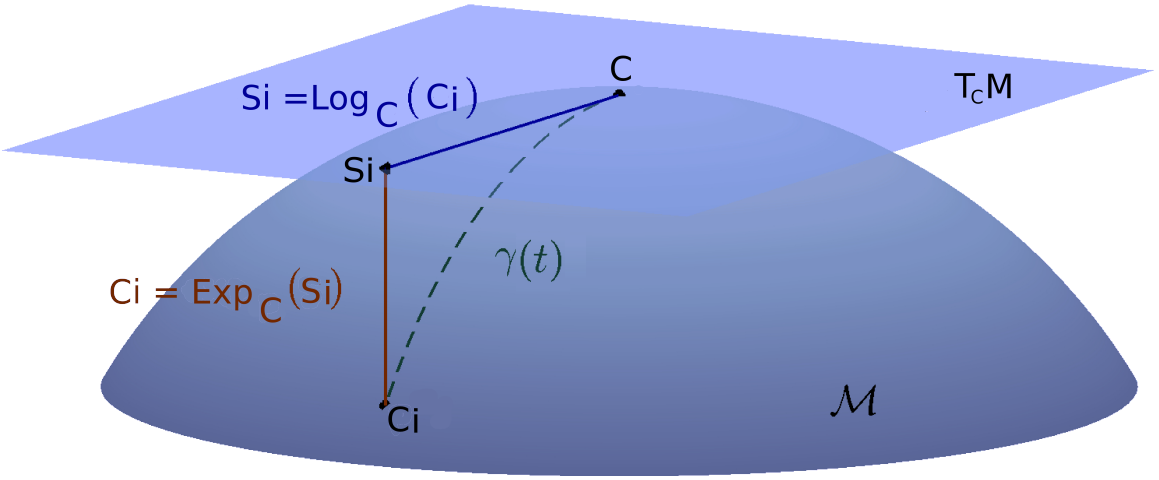

FIGURE 3.2: Manifold M and the corresponding local tangent space TcM at C. The Logarithmic map $Log_C(.)$ projects the matrix $C_i \in M$ into the tangent space. The Exponential-map $Exp_C(.)$ projects the element of the tangent space Si back to the manifold. [21]

For each point C (which is actually covariance matrix) of manifold M, there lies a tangent vector space TcM for that point in euclidean space, which is locally homomorphic to manifold, as shown in Figure 3.2. Euclidean distance computations in the tangent space is analogous to Riemannian distance computations in the manifold. Let S_1 and S_2 be two tangent vectors, their scalar product at C can be given by the equation:

$$\langle S_1, S_2 \rangle_C = Tr(S_1 C^{-1} S_2 C^{-1}) \tag{3.14}$$

3.2.3.1 Logarithmic Map

Each locally defined covariance matrix $C \in C(n)$, is projected to tangent space S_i using:

$$S_i = Log_C(C_i) = C^{\frac{1}{2}} log(C^{\frac{-1}{2}} C_i C^{\frac{-1}{2}}) C^{\frac{1}{2}} \tag{3.15}$$

On other hand, **exponential map** is used for inverse operation, i.e projection of tangent space back into the manifold using:

$$Exp_C(S_i) = C_i = C^{\frac{1}{2}} exp(C^{\frac{-1}{2}} S_i C^{\frac{-1}{2}}) C^{\frac{1}{2}} \tag{3.16}$$

3.3 Elastic Net

After projecting features into tangent space, training set is used for learning parameters using elastic net. Elastic net is linear regularized regression algorithm, which overcomes limitations of lasso and ridge regression by introducing $l1$ and $l2$ penalty [30] and also works good on numerical attributes. Our problem can be formulated in elastic net . We have $n = 5440$ observations with $p = 2211$, be the number of predictors.$Y = (y_1, y_2, y_3, ...y_n)^T$ be the class labels of training data and $X = [X_1]...[X_p]$ be model matrix where $x_j = (x_{1j}, ...x_{nj})^T, i = 1...p$ be the predictors. For any fixed non-negative λ_1 and λ_2 , the elastic net:

$$L(\lambda_1, \lambda_2, \beta) = |Y - X\beta|^2 + \lambda_2|\beta|^2 + \lambda_1|\beta|_1 \tag{3.17}$$

where

$$|\beta|^2 = \sum_{j=1}^{p} \beta_j^2 \ and \ |\beta|_1 = \sum_{j=1}^{p} \beta_j$$

Then the elastic net estimator $\hat{\beta}$ is chosen by minimizing the elastic net criterion equation 3.17:

$$\hat{\beta} = arg \min_{\beta} L(\lambda_1, \lambda_2, \beta) \tag{3.18}$$

let $\alpha = \frac{\lambda_1}{\lambda_1 + \lambda_2}$, then equation 3.17 can be rewritten as :

$$\hat{\beta} = arg \min_{\beta} |Y - X\beta|^2, \ subject\ to\ (1-\alpha)|\beta|_1 + \alpha|\beta|_2 \tag{3.19}$$

Where $(1-\alpha)|\beta|_1 + \alpha|\beta|_2$ is known as elastic penalty. If $\alpha = 1$ it becomes simple ridge and when $\alpha = 0$ it becomes simple lasso regression. Usual linear regression equation is used to predict for testing data after estimating parameters from training data.

$$\hat{y} = \beta_0 + x_1\beta_1 + ... + x_p\beta_p \tag{3.20}$$

After tangent space mapping data is used for classification purpose. Elastic net is used for training and testing. Hyper parameters of elastic net were tuned by cross validation.

3.4 Model evaluation

The area under the ROC (Receiver Operating Characteristic) curve, or simply AUC, has been widely used to measure model performance for binary classification tasks [31]. Let N_+be the number of positive instances and N_-be the number of negative instances. $x_1, \ldots, x_{N+}$ be the scores predicted by model for N_+ positives and $y_1, \ldots, y_{N-}$ be the scores predicted by model for N_- negative class. Assume both x_i and y_j has been normalized and are with in (0,1). Then AUC is given by:

$$AUC = \frac{1}{N_+ + N_-} \sum_{i=1}^{N_+} \sum_{j=1}^{N_-} I(x_i > y_j)$$

where I() is an indicator function satisfying I(true)=1 and I(false)=0.

3.5 Tools Used

In our case, for each class 5 xDAWN spatial filters are computed using scipy [32] library of python using eigen value decomposition

as described above, after preprocessed signals. Finally results is concatenated to each epoch and passed to next step for further processing.

Covariance matrices are calculated using pyriemann [2] library. Elastic net is used from scikit-learn [33] library of python.

3.6 Dataset

Data in this thesis is used from an experiment conducted by Perrin et al. [22]. This data set was also used for open data mining competition at Kaggle "BCI Challenge @ NER 2015". Subject's brain activity in this experiment was recorded with total 56 passive Ag/AgCl EEG electrodes (VSM-CTF compatible system) and 275 MEG axial gradiometers (VSM-CTF Omega 275) EEG sensors whose placement was according to the extended 10-20 system. Nose was used as a reference point for recording all the signals. The ground electrode was placed on the shoulder and impedances were kept below 10 kΩ. Position of the electrodes followed international system as shown in Figure 1.2. Recording the Event Related Potential (ERP) form the subject's brain was stimulated by visual stimuli by looking at the target letter on P300 speller matrix having 6X6 alphanumeric characters.

Data is divided into training as well as testing data. Training data contains 16 subjects and while test data has 10 subjects, who had gone through 5 different spelling sessions. In four sessions subjects have to spell twelve 5-letter words and in fifth session twenty 5-letter words. Total training trials recorded were 5440 while 3400 test trials.

[2] https://github.com/alexandrebarachant/pyRiemann

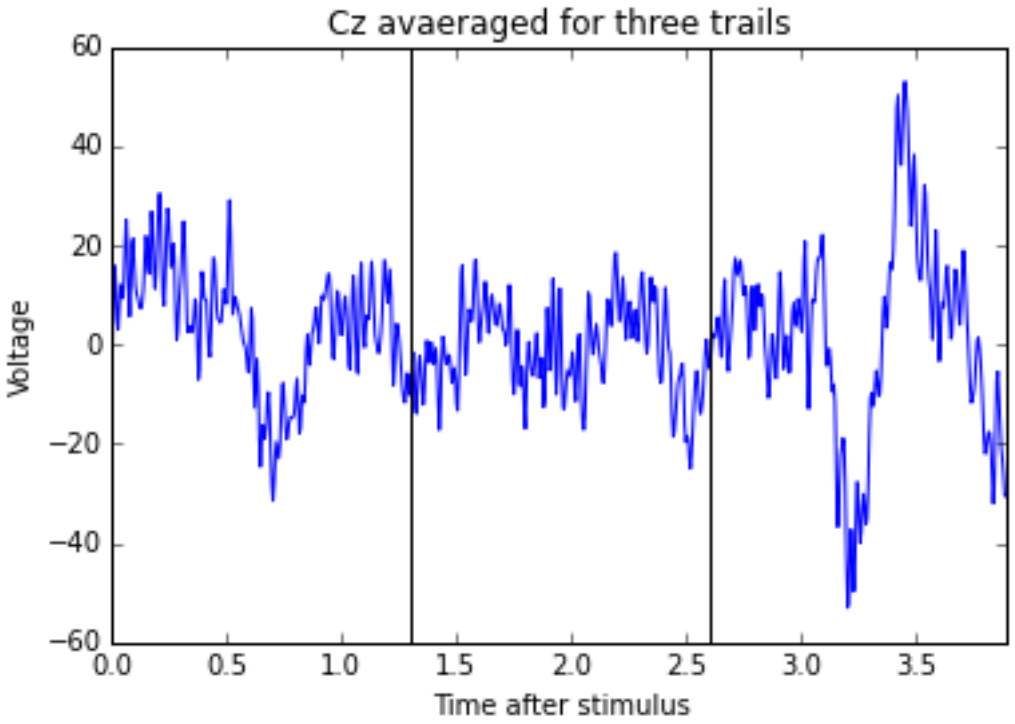

FIGURE 3.3: Three sample epoch.

In P300 BCI we have two labels of data, one is target and other is Non-target.

Three successive trials from Cz electrode are shown in Figure 3.3. Each trial is of 1.3 seconds and separated by black vertical line.

Chapter 4

Results

The results of the experiments that were described above are presented in this chapter. The observations of the performance trends are noted with the results. First the results are shown from our proposed classification pipeline and then the results are compared with most recent state-of-the-art award winning solution at Kaggle[1] by Alexandre [7].

4.1 Results Analysis

Experiments were run on all data using all electrodes as described in Section 3.2. First of all we are going to see how well our model performs, for this we have used ROC (Receiver Operating Characteristic) curve as performance measure, as it is widely used in BCI paradigm. In ROC curve AUC (Are under curve) [31, 34] determines the performance of classifier more clearer than just scalar metrics like accuracy. ROC curve along with AUC for our model is shown in Figure 4.1.

[1] https://www.kaggle.com/c/inria-bci-challenge/leaderboard

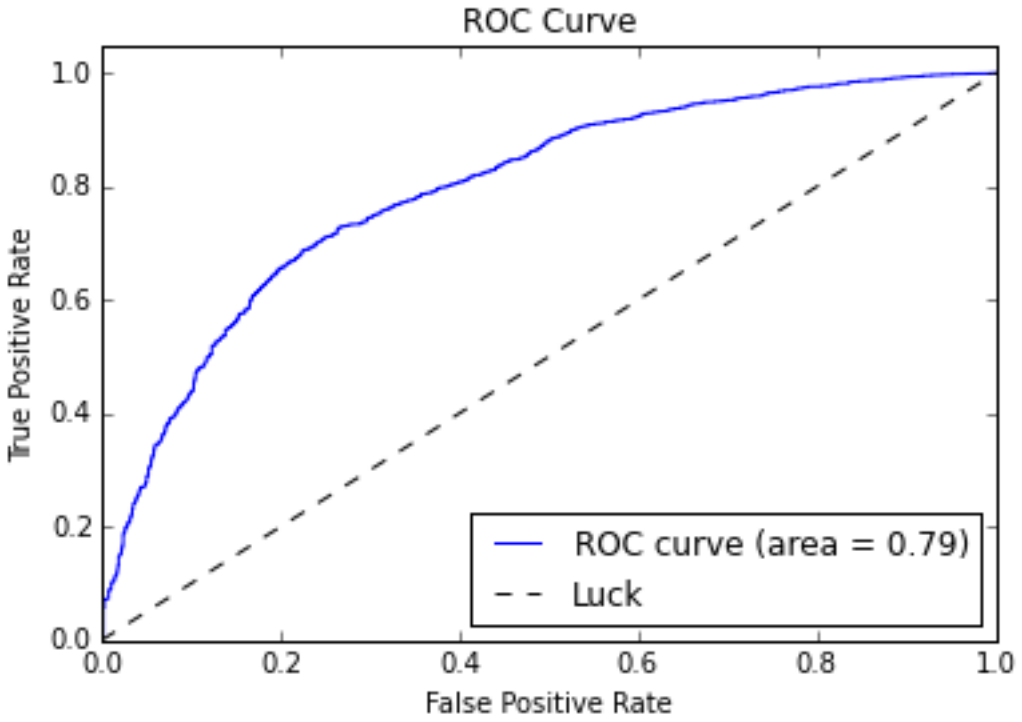

FIGURE 4.1: Area Under Curve of the model

In ROC curve, we have TPR (True Positive Rate) is plotted against FPR (False Positive Rate). More the area is covered by curve, better the model will be. Here area under curve for our model is 0.79 for whole test data, which shows better model performance.

AUC across all five sessions for all test subjects is summarized in Table 4.1. It shows good performance across subjects as classifier was

TABLE 4.1: AUC of subjects across sessions

Session→	1	2	3	4	5
Subject 1	0.79	0.80	0.81	0.66	0.56
Subject 3	0.84	0.88	0.69	0.75	0.67
Subject 4	0.80	0.94	0.91	0.90	0.80
Subject 5	0.61	0.70	0.65	0.52	0.65
Subject 8	0.91	0.90	0.90	0.86	0.75
Subject 9	0.63	0.75	0.71	0.87	0.75
Subject 10	0.90	0.96	0.90	0.93	0.96
Subject 15	0.94	0.79	0.76	0.99	0.82
Subject 19	0.64	0.58	0.63	0.65	0.69
Subject 25	0.83	0.81	0.64	0.72	0.65

trained on data from other subjects. Also good generalization across sessions as subject's mental state varies with fatigue and different position of electrodes during different sessions.

4.2 Comparison with state-of-the-art

To investigate how well our model performs, we have compared our model defined in Section 3.2 with Alexandre's model. Testing data is randomly sampled into 4 sets, each of varying size 200, 500, 800 and 1200 trials.

4.2.1 AUC Comparison

First of all performance in terms of AUC is compared and shown in Table A.1 defined in appendix section. As from table you can see neither of the model outperforms other for all these varying size samples (SS=200,500,800 and 1200) of test data. Theses both models are equivalent in terms of AUC. Same results can be verified from Figure 4.2.Where AUC of model is shown on y-axis, while x-axis shows index of the fold. In this figure green line shows AUC from Alexandre (here shortly written as alex), while our model's AUC is shown in blue color.

In Figure 4.3, a bar chart is shown for each group of varying sample size i.e 200, 500, 800 and 1200, their mean AUC is plotted along y-axis to give more detailed comparison. In blue color our model's AUC is shown while Alexander's is shown in green. As it is evident from figure that AUC gained by both the models is equivalent.

To investigate whether there is significance difference between AUC means of both models, we applied paired t-test. To compare AUC,

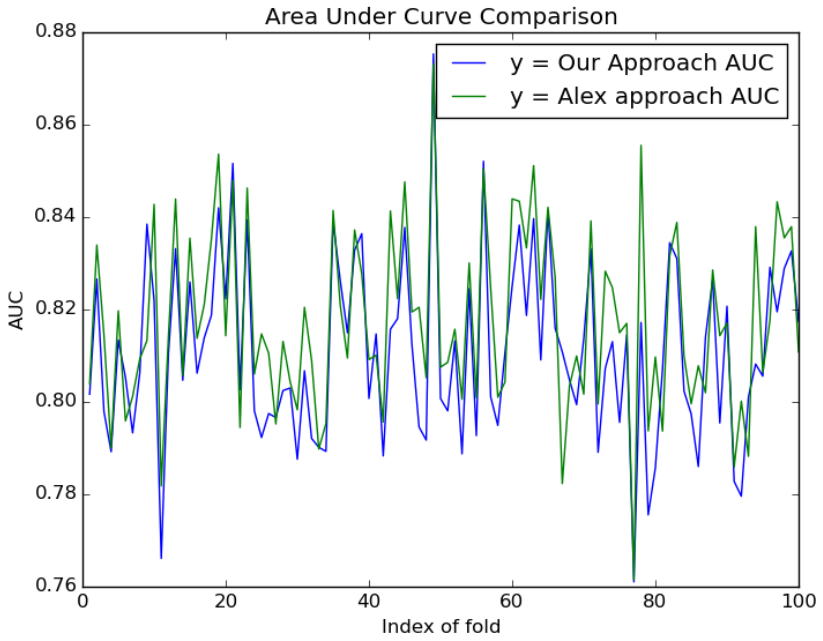

FIGURE 4.2: Comparing Area Under Curve with Alexandre Model

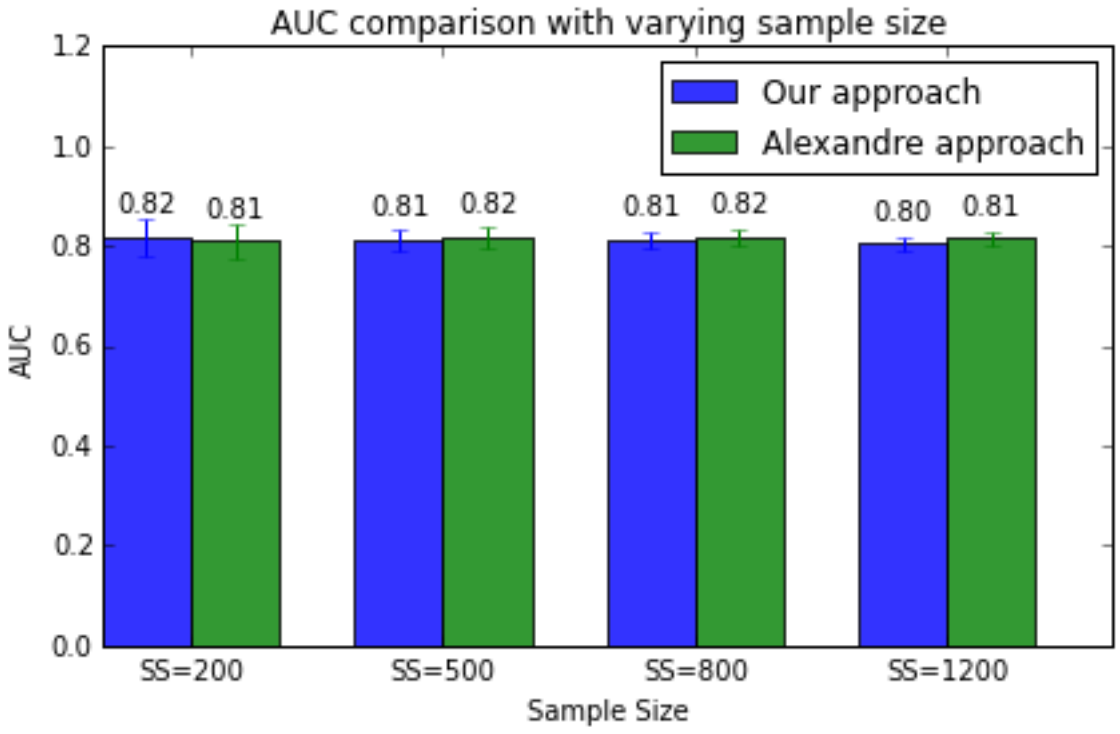

FIGURE 4.3: Comparing Area Under Curve with varying sample size

data was used from Table A.1 for sample size of 500. Test results are shown in Table 4.2. As p-value $p > 0.05$, this suggests that none of the classifier's AUC is statistically significant than other, this has even low effect size given by Cohen's-d=0.1173891.

TABLE 4.2: Comparing AUC t-test

	Test	Results
1	Paired t-test:	t(99) = 1.652, p = 0.1017, d = 0.1173891

4.2.2 Time Comparison

Now performance in terms of time taken is compared and shown in Table A.2. As from table you can see, there is lot of difference in time taken for both of the models. For sample size of 200, our model takes 34.22 seconds while Alexandre's model is taking 212.28 seconds. Our model outperforms the Alexander's model for all these varying size samples of data. Theses both models perform differently in terms of time. This result can be verified from Figure 4.4. Where time taken by model is shown on y-axis, while x-axis shows index of the fold.

In Figure 4.5, a bar chart is shown for each group of varying sample size i.e 200, 500, 800 and 1200, their mean time is plotted along y-axis to give more detailed comparison. In blue color our model's time taken is shown while Alexander's is shown in green. As it is evident from figure that time taken by our model is very much less than that of Alexander's model.

To investigate whether there is significance difference between means of time of both models, we applied paired t-test. To compare time, data was used from Table A.2 for sample size of 500. Result of test is shown in Table 4.3. As p-value $p < 0.05$, this suggests that our

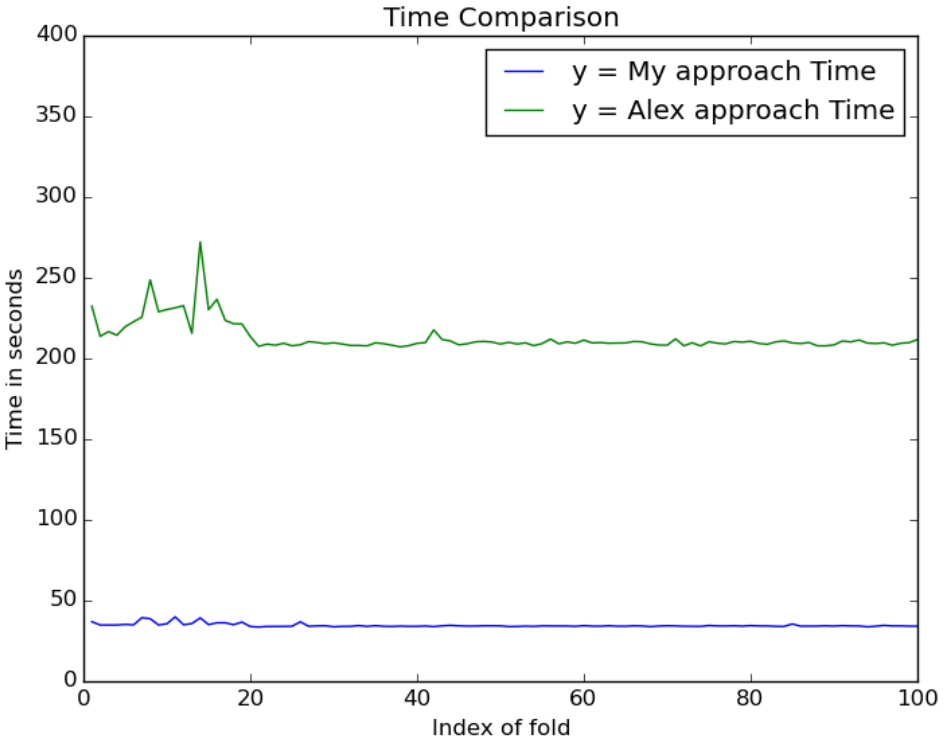

FIGURE 4.4: Comparing Time with Alexandre Model

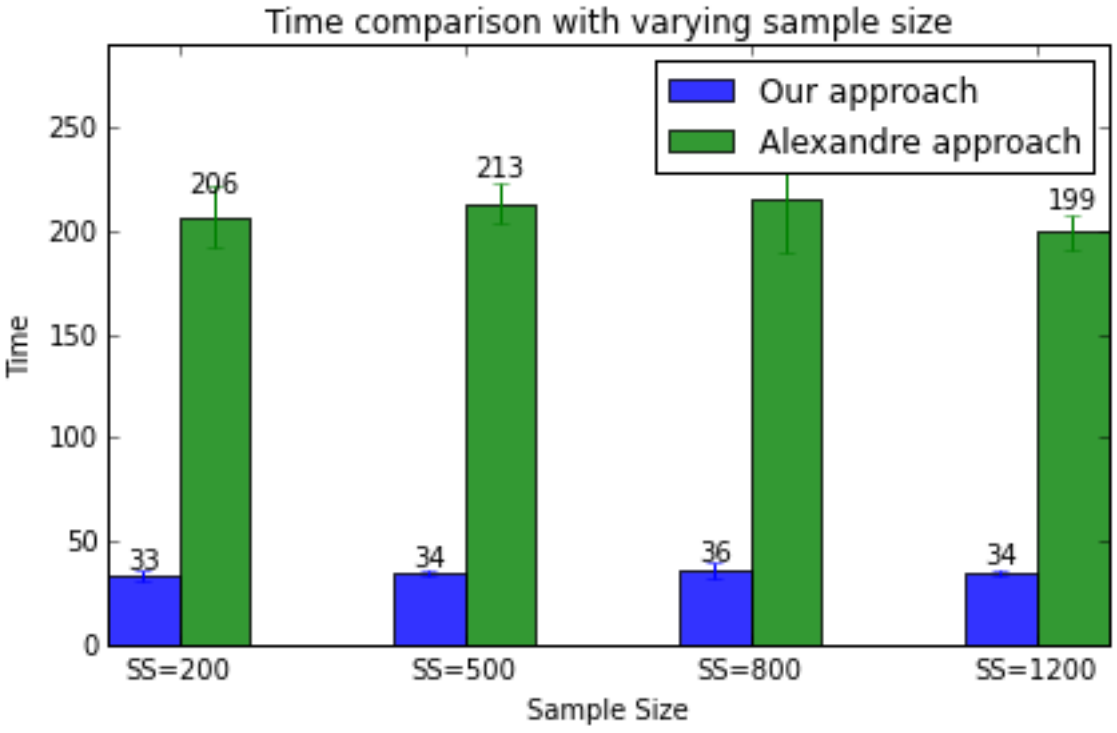

FIGURE 4.5: Comparing Time with varying sample size

classifier is statistically significant than Alexander's model, this has even high effect size given by Cohen's-d=279.1228.

TABLE 4.3: Comparing Time t-test

	Test	Results
1	Paired t-test:	t(99) = 1958.9, p = 2.2^{-16}, d = 279.1228

Chapter 5

Conclusion

5.1 Summary of Results

This thesis attempts to construct a decision model for P300 data by using tangent space mapping whereas similar approaches were applied in Motion Imagery (MI) modality of BCI. Our proposed system preprocesses the P300 data generated from brain using EEG. Raw signals were filtered through band pass filter and feature extraction was done by applying xDAWN covariances. Tangent space mapping was used before applying classification using Elastic net. Parameters for Elastic net were tuned by cross validation.

Although very nice approaches [12, 14] have been seen in literature in the recent past which offers very nice services to proposes feature extraction methods for P300 signals but most of them uses spatial filters and exploits riemannian geometry but the P300 data requires significant method to preprocess. We use xDAWN spatial filters then combining them with covariance matrices. Then intermediate results are projected from manifold belonging to riemannian framework into homologous euclidean space using tangent space mapping.

As per shown results of our proposed methodology; the system performs efficiently to classify the P300 signals in terms of AUC (Are under Curve) in real time scenarios we offer low computational and better results in most of cases. There is drastic decrease in terms of time taken to build a decision model while keeping higher performance in terms of AUC. Specially; our proposed system reflects a considerable better AUC in testing phase by supplying low amount of training data. We conclude that our proposed system gains even better AUC, decreases time drastically and it performs better during the inter-session and inter-subject variability.

5.2 Future Work

Feature extraction is one of the complex and noisy step for preprocessing of EEG data since it needs efficient dimension reduction techniques for better results in terms of AUC because during our experiment we found PCA (principal component analysis) could not be applied directly due to nonspecific design issues such as loss of useful information related to BCI features and so on. In future we are intended to improve AUC by applying more complex classifiers. As one channel selection algorithm from [35] based on backward elimination principle, was used by Alexandre, which neither improved nor deteriorated AUC, while it only increased computational complexity and overall time taken by model. Another algorithm may be used with low complexity and increased AUC.

We investigated impact of different classifiers on performance and proposed elastic net, however we did not investigate the impact of

changing feature extraction techniques etc on AUC and time complexity. They may increase AUC but again time may be increased in using these techniques.

Also we are only detecting errors, but error correction on run time can be very useful. Whenever computer misinterprets subject's intention and if error is detected, then correcting error on the fly based on given data can make BCI communication more reliable.

Appendix A

Comparing Performance

Table A.1: Comparing Area Under Curve

Our Model				Alexandre Model			
SS=200	SS=500	SS=800	SS=1200	SS=200	SS=500	SS=800	SS=1200
0.7243	0.8017	0.8299	0.8069	0.7576	0.8039	0.8395	0.8191
0.8598	0.8266	0.8077	0.8140	0.8907	0.8339	0.8106	0.8303
0.8074	0.7979	0.8486	0.7943	0.8307	0.8144	0.8392	0.7921
0.8041	0.7892	0.8023	0.8108	0.8405	0.7900	0.8148	0.8159
0.7907	0.8134	0.8099	0.8054	0.8007	0.8197	0.8327	0.8294
0.8442	0.8053	0.8068	0.7787	0.8399	0.7959	0.8032	0.7933
0.8268	0.7933	0.7750	0.8294	0.7994	0.8012	0.7990	0.8270
0.7556	0.8063	0.8068	0.7999	0.7274	0.8093	0.8050	0.8191
0.8023	0.8385	0.7935	0.8103	0.8329	0.8133	0.8130	0.8143
0.8262	0.8219	0.8315	0.7884	0.8093	0.8428	0.8362	0.8089
0.7500	0.7661	0.8161	0.8113	0.7515	0.7818	0.8250	0.8103
0.7850	0.8104	0.8172	0.8048	0.7992	0.8141	0.8274	0.8102
0.7843	0.8332	0.8360	0.8117	0.8240	0.8439	0.8392	0.8280
0.8668	0.8047	0.8036	0.8094	0.8260	0.8056	0.8086	0.8245
0.8205	0.8260	0.8122	0.7980	0.8493	0.8354	0.8215	0.8038
0.8198	0.8062	0.8265	0.8099	0.8115	0.8138	0.8318	0.8114
0.8176	0.8138	0.8159	0.8131	0.8345	0.8211	0.8141	0.8124
0.8161	0.8187	0.8134	0.7773	0.8304	0.8352	0.8081	0.7934
0.8233	0.8420	0.8216	0.8206	0.8354	0.8536	0.8276	0.8171
0.7849	0.8223	0.7892	0.8028	0.7602	0.8143	0.8013	0.8148
Mean 0.8054	0.8118	0.8131	0.8048	0.8125	0.8171	0.8198	0.8137

TABLE A.2: Comparing Time taken by both Models for varying sample size

Our Model				Alexandre Model			
SS=200	SS=500	SS=800	SS=1200	SS=200	SS=500	SS=800	SS=1200
34.22	36.92	39.82	36.40	212.28	232.44	228.15	257.78
34.48	34.89	38.09	34.86	207.08	213.74	233.46	201.07
33.50	34.89	33.66	35.08	213.65	216.66	211.73	199.49
34.20	34.89	33.57	34.76	215.88	214.43	196.04	202.54
33.90	35.25	33.53	34.92	210.20	219.69	196.28	202.04
33.74	34.95	33.53	35.80	209.93	222.78	196.19	204.06
33.71	39.39	33.56	35.07	215.26	225.60	196.10	201.06
38.09	38.76	33.46	34.66	209.30	248.71	196.17	197.02
40.30	34.90	33.54	34.71	222.09	228.97	200.66	197.47
34.27	35.63	34.83	34.68	235.35	230.32	196.36	197.48
33.85	39.91	34.75	34.74	211.86	231.42	217.03	198.43
33.45	35.02	39.09	34.69	208.35	232.67	239.19	197.70
33.25	35.79	38.26	34.82	251.57	215.62	233.66	198.07
46.29	39.28	37.56	34.67	300.09	272.19	245.85	197.13
34.44	35.12	40.39	36.17	247.31	230.19	246.35	199.45
40.43	36.25	51.85	39.06	245.98	236.71	312.25	207.15
38.96	36.26	39.70	39.38	225.71	223.59	303.85	224.23
35.19	35.07	40.48	37.98	228.15	221.56	252.48	230.21
34.06	36.69	40.66	39.00	214.12	221.39	247.24	221.43
36.53	33.92	40.59	38.27	207.28	213.66	253.46	222.28
Mean 35.84	36.18	37.54	35.98	224.57	227.61	230.12	207.80

Bibliography

[1] J d R Millán, Rüdiger Rupp, Gernot R Müller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kübler, Robert Leeb, et al. Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in neuroscience*, 4, 2010.

[2] Thorsten O Zander, Christian Kothe, Sebastian Welke, and Matthias Rötting. Utilizing secondary input from passive brain-computer interfaces for enhancing human-machine interaction. In *Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, pages 759–771. Springer, 2009.

[3] Jonathan R Wolpaw and Dennis J McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.

[4] Lawrence Ashley Farwell and Emanuel Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523, 1988.

[5] A Furdea, S Halder, DJ Krusienski, D Bross, F Nijboer, N Birbaumer, and A Kübler. An auditory oddball (p300) spelling system for brain-computer interfaces. *Psychophysiology*, 46(3): 617–625, 2009.

[6] F Sharbrough, GE Chatrian, RP Lesser, H Lüders, M Nuwer, and TW Picton. American electroencephalographic society guidelines for standard electrode position nomenclature. *J. clin. Neurophysiol*, 8(2):200–202, 1991.

[7] Alexandre Barachant, Rafał Cycon, and Cedric Gouy-Pailler. P300-speller: Géométrie riemannienne pour la détection multi-sujets de potentiels d'erreur. In *GRETSI 2015*, 2015.

[8] Dean J Krusienski, Eric W Sellers, François Cabestaing, Sabri Bayoudh, Dennis J McFarland, Theresa M Vaughan, and Jonathan R Wolpaw. A comparison of classification techniques for the p300 speller. *Journal of neural engineering*, 3(4):299, 2006.

[9] Zachary Cashero. *Comparison of EEG preprocessing methods to improve the classification of P300 trials*. PhD thesis, Colorado State University, 2011.

[10] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of erp components—a tutorial. *NeuroImage*, 56(2): 814–825, 2011.

[11] Jason Farquhar and N Jeremy Hill. Interactions between pre-processing and classification methods for event-related-potential classification. *Neuroinformatics*, 11(2):175–192, 2013.

[12] Marco Congedo, Alexandre Barachant, and Anton Andreev. A new generation of brain-computer interface based on riemannian geometry. *arXiv preprint arXiv:1310.8115*, 2013.

[13] Moritz Grosse-Wentrup and Martin Buss. Multiclass common spatial patterns and information theoretic feature extraction. *Biomedical Engineering, IEEE Transactions on*, 55(8):1991–2000, 2008.

[14] Fabien Lotte and Cuntai Guan. Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms. *Biomedical Engineering, IEEE Transactions on*, 58(2): 355–362, 2011.

[15] Alexandre Barachant and Marco Congedo. A plug&play p300 bci using information geometry. *arXiv preprint arXiv:1409.0107*, 2014.

[16] Marco Congedo, Matthieu Goyat, Nicolas Tarrin, Gelu Ionescu, Léo Varnet, Bertrand Rivet, Ronald Phlypo, Nisrine Jrad, Michael Acquadro, and Christian Jutten. " brain invaders": a prototype of an open-source p300-based video game working with the openvibe platform. In *5th International Brain-Computer Interface Conference 2011 (BCI 2011)*, pages 280–283, 2011.

[17] Bertrand Rivet, Antoine Souloumiac, Virginie Attina, and Guillaume Gibert. xdawn algorithm to enhance evoked potentials: application to brain–computer interface. *Biomedical Engineering, IEEE Transactions on*, 56(8):2035–2043, 2009.

[18] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[19] Florian Yger. A review of kernels on covariance matrices for bci applications. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–6. IEEE, 2013.

[20] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *Biomedical Engineering, IEEE Transactions on*, 59(4):920–928, 2012.

[21] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172–178, 2013. doi: 10.1016/j.neucom.2012.12.039. URL http://dx.doi.org/10.1016/j.neucom.2012.12.039.

[22] Perrin Margaux, Maby Emmanuel, Daligault Sébastien, Bertrand Olivier, and Mattout Jérémie. Objective and subjective evaluation of online error correction during p300-based spelling. *Advances in Human-Computer Interaction*, 2012:4, 2012.

[23] Hubert Cecotti, Bertrand Rivet, Marco Congedo, Christian Jutten, Olivier Bertrand, Emmanuel Maby, and Jérémie Mattout. A robust sensor-selection method for p300 brain–computer interfaces. *Journal of neural engineering*, 8(1):016001, 2011.

[24] Neng Xu, Xiaorong Gao, Bo Hong, Xiaobo Miao, Shangkai Gao, and Fusheng Yang. Bci competition 2003-data set iib: enhancing p 300 wave detection using ica-based subspace projections for bci applications. *IEEE transactions on biomedical engineering*, 51 (6):1067–1072, 2004.

[25] Shun-ichi Amari and Hiroshi Nagaoka. Methods of information geometry, volume 191 of translations of mathematical monographs. *American Mathematical Society*, page 13, 2000.

[26] F Barbaresco. Innovative tools for radar signal processing based on cartan's geometry of spd matrices & information geometry. In *Radar Conference, 2008. RADAR'08. IEEE*, pages 1–6. IEEE, 2008.

[27] P Thomas Fletcher and Sarang Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 87–98. Springer, 2004.

[28] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1713–1727, 2008.

[29] Maher Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26(3):735–747, 2005.

[30] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[31] Shaomin Wu and Peter Flach. A scored auc metric for classifier evaluation and selection. In *Second Workshop on ROC Analysis in ML, Bonn, Germany*, 2005.

[32] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. 2001–. URL http://www.scipy.org/. [Online; accessed 2016-01-07].

[33] 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*, 12: 2825–2830, 2011.

[34] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[35] Alexandre Barachant and Stephane Bonnet. Channel selection procedure using riemannian distance for bci applications. In *Neural Engineering (NER), 2011 5th International IEEE/EMBS Conference on*, pages 348–351. IEEE, 2011.