Grand Valley State University
ScholarWorks@GVSU

Masters Theses

Graduate Research and Creative Practice

12-2014

Automated Classification of EEG Signals Using Component Analysis and Support Vector Machines

Priya Balasubramanian
Grand Valley State University

Automated classification of EEG signals using component analysis and support vector machines

Priya Balasubramanian

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Engineering

Padnos College of Engineering and Computing

December, 2014

Acknowledgements

I would first like to thank my thesis advisor Dr. Samhita Rhodes for her guidance, expertise and time commitment throughout this thesis. I appreciate her patience in explaining the concepts and with the interpretation of the results. I would like to thank Dr. Paul Fishback for taking the time to be a part of all my thesis meetings and for explaining the mathematical models in great detail. I would like to thank Dr. Robert Bossemeyer for always asking the difficult questions which helped guide me in the right direction. I would also like to thank Dr. Konstantin Elisevich and Spectrum health for providing such a opportunities to work with physiological data. Thank you to Dr. Ayman Haykal for his guidance in understanding epilepsy and analysis of the epileptic data. Thanks to my parents Syamala and Balasubramanian for their support and encouragement. Lastly, I would like to thank my daughter Sneha and my husband Sanjay for their continued dedication and faith in my capabilities.

Abstract

Epileptic seizures are characterized by abnormal electrical activity occurring in the brain. EEG records the seizures demonstrating changes in signal morphology. These signal characteristics, however, differ between patients as well as between different seizures in the same patient. Epilepsy is managed with anti-epileptic medications but in some extreme cases surgery might be necessary. Non-invasive surface electrode EEG measurement gives an estimate of the seizure onset but more invasive intra-cranial electrocorticogram (ECoG) are required at times for precise localization of the epileptogenic zone.

The epileptogenic zone can be described as the cortical area targeted for resection to render the patient symptom free. Epileptologists use the “evolution” of aberrant signals for identifying epileptic seizures and the epileptogenic zone is identified by concentrating on the area contributing to the onset of seizure. This process is done by visually analyzing hours of ECoG data. The signal morphology during an epileptic seizure is not very different from abnormal discharges noticed in ECoG data thereby complicating signal analysis for the epileptologists.

This thesis aims to classify the ECoG channel data as epileptic or non-epileptic using an automated machine learning algorithm called support vector machines (SVM). The data will be decomposed into various frequency bands identified by wavelet transform and will span the range of 0-30Hz. Statistical measures will be applied to these frequency bands to identify features that will subsequently be used to train SVM. This thesis will further investigate feature reduction using multivariate analysis methods to train the SVM and compare it to the performance of classification when all the features were used to train SVM.

Results show that channel data classification using trained SVM that did not undergo feature reduction performed better with 98% sensitivity but needed more runtime than the SVM

algorithms that was trained using reduced features. For high frequency analysis of frequencies between 60-500Hz, the results show the same sensitivity yet less specificity when compared to the classification using lower frequency range of 0-30Hz.

The results seen in this thesis show that support vector machines classifiers can be trained to classify the data as epileptic or non-epileptic with good accuracy. Even though training the classifiers took almost two hours, it was still noticeably less than other machine learning algorithms such as artificial neural networks. The accuracy of this algorithm can be improved with changes to the data segment length, size of training matrix, accuracy of epileptic and non-epileptic data, and amount of data used for training.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Neurological signals

Brain activity may be measured non-invasively using electroencephalography (EEGs) or invasively using electrocorticography (ECoG). Both sets of signals have been used to understand, diagnose, and treat a number of neurological disorders and abnormalities of the brain. EEG signals are recorded by placing the electrodes on the scalp over the skull. ECoG records brain activity via slender penetrating subdural electrodes that are inserted directly in or via surface subdural electrodes that are placed over the cerebral cortex [1].

Currently clinical EEG analysis is performed visually by electroencephalographers trained to identify and locate abnormalities in EEG signals. Since multiple electrodes generate a lot of data, the visual process of EEG analysis is tedious and prone to operator bias. EEG signal processing techniques help speed up this tedious process and allow medical professionals to identify abnormalities quickly and accurately. EEG signal analysis is used in the diagnosis and treatment of various neurological diseases including but not limited to seizure detection. Hence automation of signal analysis saves valuable time. The challenges towards automation of signal analysis techniques particularly for seizure detection include the following:

- Seizure activity presents with a variety of signal characteristics that differ between patients as well as between different seizures in the same patient.
- Abnormalities noticed during epileptic seizures may also be seen during some non-epileptic activities like subconscious mental activity or stress, thereby complicating the classification process.

For simple classification, a seizure is considered to be epileptic when it results from abnormal brain activity and the seizure is non-epileptic when it is a result of subconscious mental activity that are more psychological in nature such as psychogenic non-epileptic seizures. The resemblance between these seizures exists yet varies considerably so to derive a single set of features that can classify the seizure as epileptic or non-epileptic is difficult. The psychogenic non-epileptic seizures are, however, characterized by absence of electrographic activity. Automation presents significant challenges but is still coveted by neurologists, epileptologists and neurophysiologists.

1.2. Focus: Analysis and Classification of Epileptic Seizures

EEG is used for the following purposes in epilepsy studies:

- To clinically diagnose epilepsy
- To classify epileptic seizures
- To identify the epileptogenic zone for pre-surgical patients
- To confirm the absence of epileptic seizures

Since the recorded EEG requires gigabytes of data storage space for a single patient, automated signal analysis, classification, and prediction are being explored. Research suggests that in order to devise a reliable prediction algorithm for epilepsy that will provide insight into the neurophysiologic state just prior to an epileptic seizure, it is important to test and train an automated learning system which performs EEG signal analysis and classification. Even though EEG signals are used for multiple other purposes, the focus of this thesis will be automated detection of epileptic seizures (Figure 1.1).

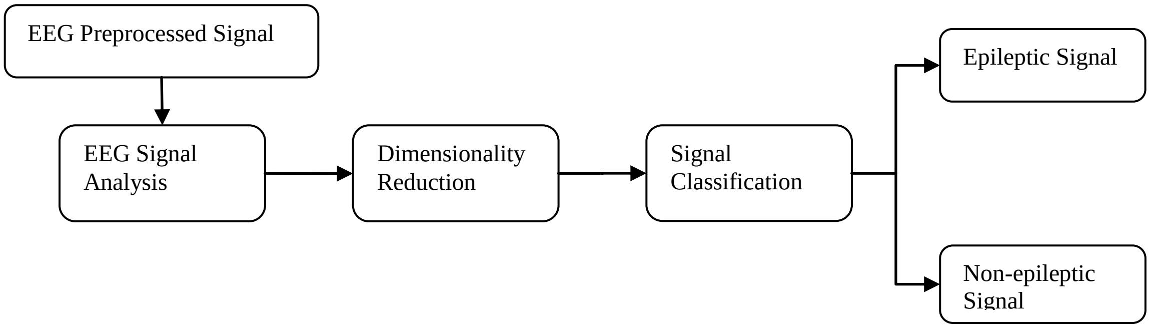


Figure 1.1 EEG Signal Classification

Subasi and Gursoy[2] performed EEG signal classification using multivariate analysis and support vector machines to compare the performance of the classification processes in an attempt to identify the optimal process. According to the authors, the heterogeneity of epilepsy mandated the requirement of configuring intelligent devices to each patient's neurophysiology prior to clinical operation. The authors used epileptic data from scalp EEG recordings of seizure activity from patients diagnosed with petit mal epilepsy and non-epileptic data from the scalp EEG recordings of healthy volunteers with no history of epilepsy. The proposed method showed promising results identifying the multivariate analysis methods for dimensionality reduction to be superior to training SVM without dimensionality reduction. Even though the authors used scalp recording for their proposed method, this paper forms the basis for this thesis in the automated classification of the ECoG signals.

The automated epilepsy detection system would include four steps; EEG signal preprocessing, EEG signal analysis using discrete wavelet transform, dimensionality reduction using multivariate analysis methods, and signal classification using support vector machines[3]. EEG signal preprocessing prepares the raw data for easy analysis. In this stage, the signal is filtered and normalized to remove noise due to artifacts. Time-frequency analysis of EEG signals using discrete wavelet transform allows the data to be classified based on the wavelet coefficients. Features are extracted from wavelet coefficients by computing mean and variances of different frequency bands to create a feature matrix. The feature matrix will serve as inputs in creating the training matrix for the support vector machines to classify the signal as epileptic or non-epileptic. Support vector machines (SVM) algorithms are learning algorithms that distinguish epileptic and non-epileptic signals. In addition, dimensionality reduction via multivariate analysis is performed on the feature matrix. The transformed and reduced feature training matrix will also be used as inputs to SVM. Results from the SVM classified data for the correct classification of epileptic rhythms will determine whether or not to use the training matrix comprised of entire feature matrix or one with reduced dimensionality.

1.3. Summary

Epilepsy is one of the most common neurological diseases and patients suffer from epileptic seizures that may be unpredictable and recurrent [4]. Scalp or intracranial EEGs are used clinically to diagnose, differentiate and classify epileptic seizures because epilepsy manifests aberrant EEG signal changes. These signals are called epileptogenic discharges and may appear in the form of spikes, poly-spikes or spike and waves. Ictal (i.e. state during epileptic seizure) EEG recordings are more reliable in diagnosing epilepsy than interictal (i.e. state between two epileptic seizures) recordings but they are expensive and difficult to obtain in patients with infrequent epileptic seizures.

Whether the data gathered is EEG data from non-invasive scalp electrodes or the more invasive ECoG data from intra-cranial electrodes, it currently must be interpreted by medical professionals. As the data usually contains interictal and ictal data, it can be quite cumbersome when analyzed visually. The purpose of computer aided EEG classification is not only to save time and effort for medical professionals but also to train the automated system to predict occurrences of epileptic seizures.

2. Literature Review

2.1. Electroencephalography and Electrocorticography

Neurons communicate by transmitting and receiving electrical impulses or signals. An EEG is a recording involving a set of electrodes or sensors for monitoring these electrical impulses in brain or the “brain waves” and was first recorded in human in 1924 by Hans Berger. The sensors or electrodes used to record the brain’s electrical activity are placed at strategic positions on top of the scalp (EEG) or in direct contact with the exposed brain cells (ECoG), depending on the type of desired neurophysiologic information.

EEG is in the form of waveforms of different frequencies and amplitudes measured across time. Since EEG recorded at the scalp is a spatial average of a large area of cortical neuronal activity, the scalp acts as a volume conductor and subsequently the EEG signals have low spatial or temporal resolution and poor signal - to - noise ratio when compared to ECoG. However, EEG is non-invasive and less expensive than the ECoG recordings, which provide signals that have less susceptibility to artifacts when compared to the standard EEG [5].

Despite the relatively poor signal quality in EEG signals, they are routinely used to study neuropathophysiology. Using advanced signal processing techniques for signal analysis allows researchers to develop automated tools that aid medical professionals in interpreting the EEG signals.

2.2. Epilepsy and EEG

Neurological diseases can be interpreted as a chemical or electrical imbalance in the brain that causes impaired brain function. Degenerative diseases, neurogenetic diseases, and convulsive disorders are but some of the examples of neurological diseases [6]. Some seizures are characterized by uncontrollable and rapid shaking of an individual's body due to an irregular or atypical electrical conductivity or connection in the brain. Epilepsy is the most common form of convulsive or seizure disorder. Even though seizures are an inherent part of epilepsy, not all seizures are due to epilepsy. While epilepsy is incurable, it can be controlled with medications or in some cases it is self limiting. When patients, do not respond to medications then surgery is an option.

EEG is most commonly used to diagnose epilepsy because epileptic seizures result from abnormal electrical activity in the brain. Seizure activity causes the EEG signal to deviate from its normal morphology. EEG signals demonstrate changes in the form of a decrease in signal frequency and an increase in signal amplitude. Also, multiple recording sites on the brain start showing an ordered pattern during seizure activity that is not prevalent during normal brain function.

The discharges in an epileptic brain can be divided into four different categories; interictal, pre-ictal, ictal and post-ictal. Pre-ictal state occurs right before the start of an epileptic seizure. Ictal state is defined as the epileptic seizure during which the functioning of the brain is impaired. Interictal state occurs between two consecutive epileptic seizures and can be characterized as normal or abnormal brain activity. During the abnormal interictal state, interictal epileptiform discharges (IEDs) or interictal slow activity can be observed. During the post-ictal state the brain is recovering from an epileptic seizure[7]. The dynamics of pre-ictal state are most complex and during this stage there is a reduction in the connectivity of neurons in the epileptogenic zone[8]. The primary objective of this thesis is to differentiate between the ictal and interictal discharges. The morphology of abnormal IEDs is similar to the ictal discharges but the "evolution" of transient changes is required to classify the discharge as ictal. The "evolution" of discharges can be described as the change in frequency, change in field (spreading to other parts of the brain), change in morphology, or change in amplitude. The neurologists or epileptologists visually scan the pre-ictal, ictal and interictal states to detect epileptic discharges but an automated signal classification algorithm will allow reproducible and objective analysis of EEG data.

2.3. Signal analysis

2.3.1. Time – Scale Domain: Discrete Wavelet Transform

The signals from any complex system can be analyzed by using both linear and non-linear tools. The use of linear tools requires the underlying assumption that the signal being analyzed is linear. This is demonstrably not the case in complex neurophysiologic systems. Hence, their use may result in the loss of information from non-linearities. On the other hand, non-linear tools are computationally intensive. The main analysis of EEG signals can be classified into time domain, frequency domain, time-frequency domain, and time-scale methods.

A signal can be constructed as a linear combination of “basis functions”. In time domain methods, the basis function is a function that isolates elements in time. In frequency domain methods, the sinusoidal basis function isolates the frequency components of the signal because sinusoids are good at isolating components of different frequencies. Time domain and frequency domain representations contain the same information but differ in the features that are accentuated in each domain.

As the time domain method fails to provide frequency content information and the frequency domain method provides temporal information only with the help of windowing, the time-frequency and the time-scale methods resolve the temporal and frequency content for non-stationary signals.

The signal analysis methods involve the remapping of the signal so more information about it can be extracted. The Fourier transform (FT) provides the frequency information by comparing the signal to a whole family of sine or cosine functions at harmonically related frequencies[9]. The advantage of using sinusoidal functions is that they contain energy at only one specific frequency which then leads to easy conversion into the frequency domain. FT uses the computationally attractive algorithm called "Fast Fourier Transform (FFT)", to allow a clear visualization of the periodicities of the signal that helps in understanding the underlying physical phenomena. FT based spectral analysis of the EEG signal data is the most commonly used quantitative method for analysis because FT allows the separation and study of different EEG rhythms when several rhythms occur simultaneously[10]. Despite the usefulness of FT in analyzing EEG signal data, there are some disadvantages to this method. FT requires or assumes the signal to be stationary but the EEG signal data is highly non-stationary and since FT is based on comparing the signal with sinusoids that extend through the whole time domain, there is a clear lack of information about the time evolution of the frequencies.

Short term Fourier transform (STFT) or spectrogram is a time-frequency analysis method which involves segmenting the signal into short time windows and performing FT on each segment. STFT has been successfully applied in a number of biomedical applications such as ECG analysis for arrhythmias, classifications of the lung sounds, and biomedical image analysis for tumor detection. STFT uses classical Fourier transforms while reducing the disadvantages of FT by assuming stationarity of the signal over shorter time segments. STFT can be used to analyze signals with high frequency components where frequency resolution is not critical. The selection of optimal window length for data segments and the time-frequency tradeoff due to the shortening the data length is the main disadvantage of spectrogram.

Time-scale analysis or “Wavelet analysis” utilizes expanded and compressed wavelets as basis functions to provide a combination of temporal and frequency information. Wavelets use varying window size, wide for slow frequencies and narrow for fast frequencies, leading to an optimal time-frequency resolution in all frequency ranges. Wavelets do not require the signals to be stationary as the windows are adapted to different transients of EEG data that are correlated to the wavelet coefficients of different scales.

For wavelet transforms, a wavelet is defined as a small wave with finite duration and energy and upon correlation with EEG signals generates wavelet coefficients.

$$\boldsymbol{W(a,b) = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)dt} \qquad \textbf{(2.1)}$$

where b acts to translates the function across x(t) and a is the scaled value of the wavelet function ψ. For a>1, the wavelet ψ, is stretched along the time axis and for a<1 the wavelet is contracted in time. The normalizing factor $\frac{1}{\sqrt{|a|}}$ ensures that the energy is the same for all values of a[9]. When a = 1 and b=0, the wavelet is in its natural form and is known as the mother wavelet[3]. The wavelet coefficients W(a,b), describe the correlation between the signal and the wavelet at various translations(b) and scales(a) and the coefficients must be added together to reconstruct the original signal. Like FT, if the wavelet function is appropriately chosen then the original signal can be reconstructed using the wavelet coefficients. Just as in STFT, the time-frequency tradeoff exists in the wavelet transformation as decreasing the wavelet time range (decreasing a) provides better time resolution but reduces the frequency resolution and increasing the wavelet time range provides better frequency resolution but poor time resolution.

The two types of wavelet transforms are Discrete Wavelet (DWT) and Continuous Wavelet Transform (CWT)[9]. The CWT (Equation 2.1), shows that ‘a’ is variable and changes during the analysis and it is often easier to analyze or recognize patterns. The CWT is highly redundant and provides an oversampling of the signal by generating more coefficients than is necessary to describe the signal[9]. The DWT (Equation 2.2), however, restricts the variation in translation and scale to powers of 2.

$$\boldsymbol{x(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k,l) 2^{-\frac{k}{2}} \psi\left(2^{-kt} - l\right)} \tag{2.2}$$

where a = 2^{k} and b = 2^{kl} and d(k,l) is a sampling of W(a,b) at discrete point k.

DWT for signal and image processing applications is described in terms of filter banks. In DWT, the EEG signal is first filtered using low-pass filter to obtain coarse coefficients and then the detailed coefficients are obtained by passing through a set of high pass filters [2]. These groups of filters are used to divide up the signal into the spectral components called sub-band coding. This method is known as the multi-resolution decomposition of the EEG signal and the main parameter of the wavelet is to choose the number of levels of decomposition of the signal where these levels are based on the dominant frequency components of the signal. The extracted wavelet coefficients provide a compact representation of the energy distribution of the EEG signal in time as well as frequency.

Both the time-frequency method such as STFT, and the time-scale methods such as, Wavelet transforms, map the non-stationary EEG signal into a two-dimensional space of time and frequency but with a time-frequency tradeoff. STFT uses a single window for the transformation while wavelet transforms uses multiple window durations that allow for varying frequency resolutions from coarse to fine. The varying size of the wavelets allows different features of the signal to be extracted.

Wavelet analysis allows the signal to be expressed as wavelet coefficients obtained by shifting and dilating a single function called the “mother wavelet”. Therefore, the signal can be reconstructed by summing the wavelet coefficients. Most of the energy of the wavelets is restricted to finite time intervals and when compared to STFT, the wavelet transforms provide good frequency resolution for lower frequencies (long time windows) and good time resolution for high frequencies (short time windows).

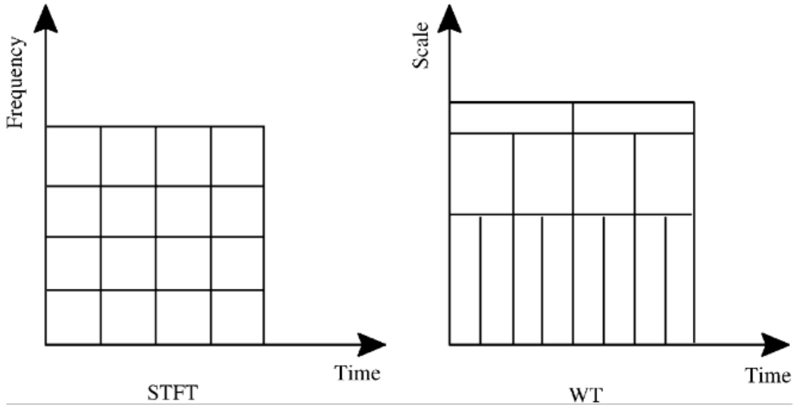


Figure 2.1 Window regions of STFT and Wavelet Transforms (WT)

In other words, STFT provides uniform time resolution across all frequencies whereas discrete wavelet analysis provides high time resolution and low frequency resolution for high frequencies while providing high frequency resolution and low time resolution for low frequencies[11]. Figure 2.1 shows the representation of window regions of STFT and wavelet transforms analysis[12,13].

2.4. Component analysis

2.4.1. Principal Component Analysis

Principal component analysis (PCA) is a form of signal analysis that identifies the principal components of multivariate data and uses these components to reduce the dimension of the data. Principal components contain statistically significant information about the data and can be defined as the variance in that data [14]. In other words, PCA identifies the components that contribute the most to the variance in the data as these components are most important to recreating the data.

Principal components are ranked in order if their decreasing contribution to the variance in the data with first principal component containing information contributing maximally to the variance in the data and subsequent principal components ranked in order. The primary purpose of PCA is to find a new set of axes so that when the data is projected on these axes, the projected points have maximum variation in a way that the projected data points are widely spread out [15]. For a given data set $x^{(i)}$ where i = 1… m, each $x^{(i)}$ is an n dimensional dataset. So when this data is reduced to a k-dimensional data where k<n, the reduction in dimension is done to reduce noise, visualize higher dimensional data into reduced number of dimensions, and to compress high dimensional data to save time and computational complexities.

To perform PCA, the eigenvectors and eigenvalues of the covariance matrix of the normalized data are obtained. The eigenvalues and their associated eigenvectors are ranked in order of magnitude so to reduce n-dimensional data; eigenvectors associated with the first k eigenvalues are chosen which form the new principal axes for dataset $x^{(i)}$.

The covariance matrix **S** can be reduced to a diagonal matrix **D** by pre and post multiplication with an orthonormal matrix **U**[9]**.** The diagonal elements of **D** make up the variances of the new data and form the eigenvalues of the covariance matrix **S**. The columns of **U** constitute the eigenvectors for the corresponding eigenvalues. These eigenvalues determine the percentage of the total variance that any given principal component represents. To simplify this approach, the singular value decomposition (SVD) works directly with the data matrix **X** that is decomposed into **D** and the principal components matrix **U**. Using SVD, the eigenvalues describe the variance accounted for by the associated principal components that are ordered by size and can be meaningful in identifying the number of principal components that are really significant. These principal components can then be used to reduce the data set as those contributing the least to the variance in the data can be eliminated.

EEG data is considered to be a large dataset and the purpose of reducing the dimension, while allowing minimal information loss as most of the data is in the lower dimensional space, is to use it as input to machine learning systems such as support vector machines.

2.4.2. Independent Component Analysis

Independent component analysis (ICA) is similar to PCA in identifying new dimensional space for representing data, but differs from PCA by treating the data as coherent groups [16]. PCA tries to identify new dimensional space for the data by finding the major axis of variation, whereas ICA identifies new dimensional space by finding the independent components of variation of data. The main computational difference between PCA and ICA is that while PCA uses variance which is only a second order statistic, ICA uses higher order statistics for computation of independent variables[9].

For a given data set of observations $x^{(i)}$ for i = 1… N

$$\boldsymbol{x^{(i)}} = \boldsymbol{As^{(i)}} \tag{2.3}$$

where A is the “mixing matrix” with sources $s^{(i)}$ that generates the data $x^{(i)}$. Here s is composed of all the source signals. The model assumes that the mixed signals are the product of instantaneous linear combinations of the independent sources.

For an “unmixing matrix”$W = A^{-1}$, the main goal is to identify W so that the original sources $s^{(i)}$ can be generated from Equation 2.4.

$$\boldsymbol{s^{(i)}} = \boldsymbol{Wx^{(i)}} \tag{2.4}$$

where W = $\begin{bmatrix} - & {w_1}^T & - \\ - & \dots & - \\ - & {w_n}^T & - \end{bmatrix}$.

It is important to note that ICA algorithms have some inherent ambiguities; the ICA algorithm cannot identify the order of the independent components or the original sources and ICA algorithms cannot differentiate between the signs of the original sources [17]. These ambiguities however do not affect most of the applications for which ICA is used.

When the original sources are considered non-Gaussian and independent, the joint density of the sources is given by

$$\boldsymbol{p(s)} = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{p_s}\,(\boldsymbol{s_i}) \qquad \textbf{(2.5)}$$

where p(s) is the density of the original source s. For recorded, zero-meaned data $x^{(i)}$, ICA uses a monotonic function (any non-Gaussian function such as sigmoid function) to identify the parameters of W using the learning rate of the training algorithm and after the algorithm converges, $s^{(i)} = Wx^{(i)}$ is computed to recover the original sources of the dataset.

To identify the independent sources in a signal using the mixing matrix, ICA has only two requirements; the sources variables ‘s’ should be truly independent and that the dataset is non-Gaussian. Since real signals satisfy both these conditions, one of the applications of ICA is the processing of various types of brain data such as EEG. As each electrode measures electrical potential generated as combinations of the underlying components of brain activity, and since only these mixtures of the components can be observed, ICA is useful in identifying the independent components in the signals in order to uncover meaningful information. In other words, since EEG data consists of recording from multiple locations on the brain and the data recorded is comprised of the mixed neural activity of the brain, ICA is used to obtain the independent components to help observe the original components of the brain activity.

2.5. Signal Classification

2.5.1. Support Vector Machines

A Learning System is the process of training a computer to perform tasks that it has not been explicitly programmed to do due to the lack of a suitable mathematical model. Learning systems use existing examples or training set data to find patterns and build classification models for problem solving. Artificial Neural Network (ANN) and Support Vector Machines (SVM) are two most common supervised learning algorithms that are used to train classifiers to separate EEG data into epileptic and non-epileptic signals [18]. While the performance of SVM and ANN with respect to linear data is quite similar, the difference is observed in non-linear data [19]. ANN uses multiple layers and various activation functions on non-linear data whereas SVM uses the kernel function as a key to separate the non-linear data. ANN uses the gradient descent algorithm to converge to local minima that leads to over-fitting of data whereas SVM converges to a global minima while providing a simple geometric interpretation and reducing errors due to over-fitting.

To understand SVM and to see how the learning system employs the computational learning theory to classify the data, it is important to gain knowledge of the margins that act as the optimum marginal classifiers and kernels that allow SVM to be used efficiently with the high dimensional data. A classifier is considered to be linear when the separating hyperplane is a decision boundary that separates the positive and negative classes of the data with a clear margin. The classifier for a linear classification is

$$\boldsymbol{h_{w,b}(x) = y(w^T x + b)} \qquad \textbf{(2.6)}$$

where y∈{-1,1}, w∈R^n, x∈R^n and b is a real number.

The linear classifier is the simplest form of SVM and is usually used for data that can be separated using a separating hyperplane and forms the basis for more complex and non-linear classifiers in SVM. If the data are not linearly separable and the points overlap then the linear classifiers are limited to decision boundaries that are straight lines. In more complex datasets, the linear boundary is attainable if the data is transformed to a higher dimensional space.

In Figure 2.2, the solid line represents the maximal margin separating the hyperplane and there are two negative and one positive dataset point that lie on the dotted lines that are parallel to the hyperplane. These points are called "support vectors" and in an SVM algorithm there are very few support vectors while the rest of the dataset points are non-support vectors.

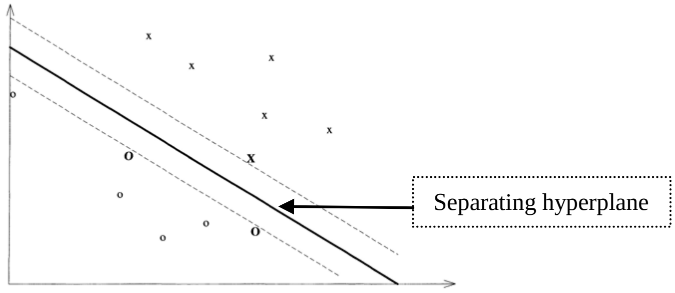


Figure 2.2 A maximal margin hyperplane with its support vectors highlighted [20]

When data vectors are mapped onto higher dimension space defined by the function ϕ(x) then the kernel function K$(x^{(i)}, x^{(j)})$ is the inner product of the feature vectors $\phi^{(i)} and\ \phi^{(j)}$ because calculating ϕ(x) is computationally intensive being a high dimensional vector. The idea of SVM is to replace the inner products in the algorithm with kernel functions that can be computed very efficiently which allows computations in high dimensional feature space. The kernel function is defined as K$(x^{(i)}, x^{(j)}) \equiv\ \phi(x^{(i)})^T \phi(x^{(j)})$. The kernel function types include:

- Linear : K$(x^{(i)}, x^{(j)}) \equiv\ (x^{(i)})^T (x^{(j)})$
- Radial Basis Function: K$\left(x^{(i)}, x^{(j)}\right) = \exp^{(-\gamma\left\|x^{(i)}, x^{(j)}\right\|^2)}$, γ>0
- Polynomial: K$(x^{(i)}, x^{(j)}) \equiv\ (\gamma(x^{(i)})^T (x^{(j)}) + r)^d$, γ>0
- Sigmoid: K$(x^{(i)}, x^{(j)}) \equiv tanh(\gamma(x^{(i)})^T (x^{(j)}) + r)$

Here, γ, r and d are kernel parameters[21].

While considering the EEG data which can be described as non-linear dataset as it originates from a non-linear system, the SVM algorithm takes the input data and projects them to a higher dimensional space. The kernel function defines the characteristics of the input data in this high dimensional space. Once the data that could not be separated linearly is projected to a higher dimension, it is easier for SVM to obtain a separating plane between the two classes of the data. The radial basis function is preferred as the kernel handles non-linear data with less numerical difficulties. RBF kernel uses γ parameter as kernel parameter for SVM and σ as the penalty parameter. The penalty parameter allows the SVM to misclassify some of the parameters in order to obtain an optimal separating plane. The identification of these parameters is paramount for training of the classifier to achieve high accuracy[21].

3. Specific aims

The purpose of this thesis is to classify ECoG signals as epileptic and non-epileptic using an automated machine learning algorithm. In this thesis, the epileptic signals will concentrate on ictal discharges and the non-epileptic signals are made up of normal and abnormal interictal discharges. To achieve this, the algorithm will:

1. Extract and normalize the ECoG data channel of interest using EEGLab.
2. Decompose the signal using wavelet decomposition.
3. Extract statistical features to create a feature set.
4. Use Principal Component analysis and Independent Component analysis to reduce the number of features of the feature set to decrease runtime.
5. Use the reduced feature set to train the Support Vector Machine.
6. Compare the performance of the SVM trained on original and reduced feature set to separate epileptic from non-epileptic signals on a test data set.

4. Methodology

4.1. Data Collection: Clinical ECoG data

The ECoG data used in this thesis comprised of ECoG collected from a single patient. Subject #2 was suffering from focal epilepsy and the data was collected as part of pre-surgical preparations prior to removing the epileptogenic focus. The epileptogenic focus was defined by the epileptologist as the site in the brain that comprised of the seizure onset zone. The resection of this zone eliminated the clinical symptoms due to epileptic seizures for the patient. The data were obtained with permission from Spectrum Health's Epilepsy Monitoring Unit (EMU) located in Grand Rapids, Michigan. During this comprehensive data collection, 72 intracranial electrodes were implanted on the cerebrum of the patient and ECOG data with a sampling frequency of 1000 Hz was collected for twenty four hours a day over two weeks. This data was recorded as 2 hour blocks in a file format called European Data Format or .edf. The anatomical locations of each electrode, labeled as channels throughout this thesis, and the electrode grids can be found in Appendix A. The data were then annotated by an expert epileptologist who marked the start and stop times of each seizure for the patient along with the channels, and therefore anatomical locations, that exhibited marked ictal discharge. The durations and channels varied with each of the fifteen seizures that Subject #2 experienced.

The data were used for algorithm development and for creation of the training matrices for all models and the test data set. EEGLab was used to import the data into MATLAB from which the 20 second data segments were extracted. Of the fifteen events, 12 were used for training the SVM classifiers and the remaining were part of the test data set. The training matrices and test data set comprised of 20 second segments from epileptic as well as non-epileptic data. Epileptic data were selected only from the channels and times that been noted as demonstrating ictal discharge by the epileptologist. Non-epileptic data were selected from files that did not contain any ictal activity noted by the epileptologist in his report. The purpose of this data selection was to obtain a training matrix that contained only ictal activity assigned as epileptic and absence of ictal activity assigned as non-epileptic.

4.2. Algorithm

Four models of classification, as shown in Figure 4.1 were created for identifying the most suitable combination of dimensionality reduction technique paired with SVM that gave the highest sensitivity and specificity in classifying epileptic and non-epileptic data. Each 20 second data segment in the training matrices and test data was normalized and decomposed into wavelets from which features were extracted. The steps for decomposing the signal and extracting features have been described in detailed in sections 4.4 and 4.5 respectively. Next, the features were reduced using PCA or ICA. Finally, the SVM algorithm was implemented with the training matrix created using the original set of features (Figure 4.1, classifier IV), the reduced set of features via PCA (Figure 4.1, classifier I), ICA (Figure 4.1, classifier II) or PCA+ICA (Figure 4.1, classifier III). The SVM classifier training using PCA and ICA were similar to the paper by authors Subasi and Gursoy on which this thesis is based[2]. The third model to combine the principal components and independent components to create a training set was introduced in this thesis to observe the accuracy of the classifier. Since, principal components represent the variation in data and the independent components identify the independent sources contributing to the variance in the data, the third model was created.

Once the SVM was trained and the classifiers obtained for each model, the test data set was classified. The sensitivity and specificity of each classification technique or model was computed using equations 4.1 and 4.2. The sensitivity (Equation 4.1) of the model is the measure of its ability to correctly identify the epileptic channels, and the specificity (Equation 4.2) of the model is the measure of its ability to correctly identify non-epileptic channels. The classification of the channels and data performed by the epileptologist are the gold standard for the calculation of sensitivity and specificity of each model.

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Negatives}} \boldsymbol{x100} \quad \textbf{(4.1)}$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{True\ Negatives}}{\boldsymbol{True\ Negatives + False\ Positives}} \boldsymbol{x100} \quad \textbf{(4.2)}$$

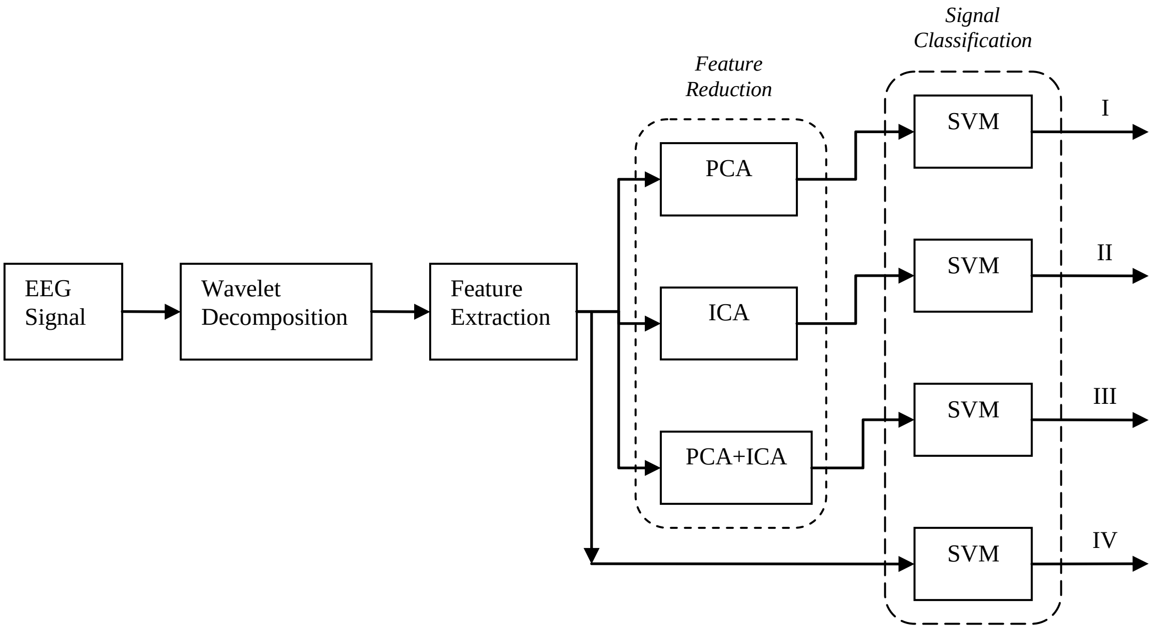


Figure 4.1 Functional diagram

4.3. Software Overview

All software development was done in MATLAB R2014a (Mathworks, Natick, MA) for performing the following steps:

1. Reading the clinical ECoG data for analysis
2. Performing wavelet decomposition on the signals
3. Reducing features using PCA and ICA algorithms

4. Implementing machine learning using SVM algorithm on the training matrix
5. Obtaining performance measures of sensitivity and specificity for test data set.

EEGLAB, an open source MATLAB compatible package, was used to import the patient data for analysis in step 1. EEGLAB is an open source environment for electrophysiological signal processing and was developed by the Swartz Center for Computational Neuroscience (SCCN) and is distributed under the GNU General Public License[22]. The Wavelet Toolbox is part of the MATLAB 2014a which allows analysis and synthesis of signals and images using wavelet techniques and was used for performing step 2. The PCA is a function of the Statistics Toolbox of MATLAB 2014a which allows the Principal Component Analysis of raw data. For dimensionality reduction using the ICA algorithm, the algorithm for real-valued signal developed by Jean-Francois Cardoso was used[23]. SVM is also a function available in the Statistics Toolbox of MATLAB 2014a and was used for classifying data into two classes.

4.4. Analysis using Discrete Wavelet Transform

The wavelet technique applied to the EEG signals reveals features related to the transient nature of the signal in both the time and frequency content. The DWT analyzes the signal at various frequency bands by decomposing the signal into coarse and fine information. This decomposition into frequency bands is achieved by using filter banks that divide a signal into various spectral components called sub-bands.

The signals are divided into high pass (Hi_D) and low pass (Lo_D) spectral characteristics where the high pass filter is like applying a wavelet to the original signal and the low pass filter is like applying a scaling or smoothing function. Figure 4.2 shows the sub-band decomposition of the signal using DWT in which each stage is made up of high pass and low pass filters. The first filter Hi_D is the discrete mother wavelet, high pass in nature and the second filter Lo_D is its mirror version, low pass in nature[12,13]. The down sampled outputs of the filters are the detailed decomposition called D1 and approximate decomposition called A1. The approximation signal A1 is further decomposed using the high pass and low pass filter pair.

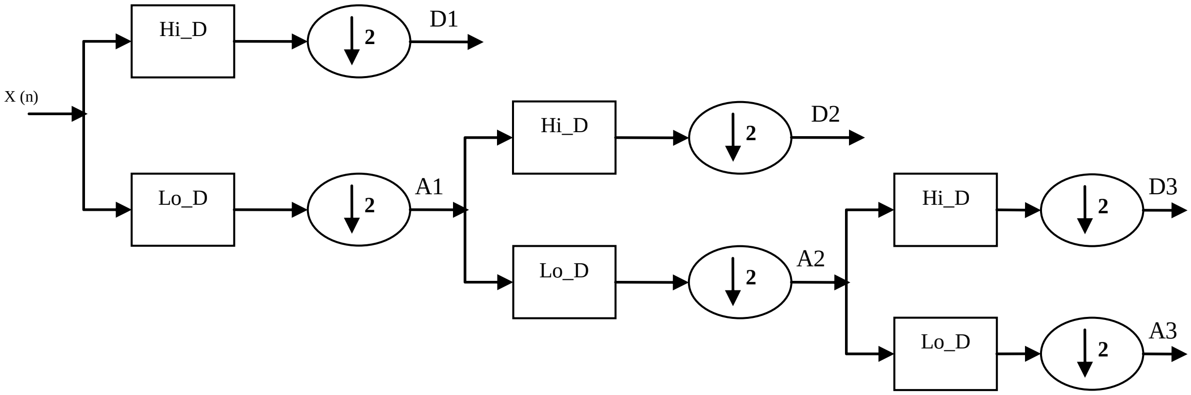


Figure 4.2 Sub-band decomposition of DWT

Table 4.1 Ranges of frequency bands for the ECOG signal with a sampling frequency of 1000 Hz

Decomposed Signal	Frequency Range (Hz)
D1	250-500
D2	125-250
D3	62.5-125
D4	31.25 – 62.5
D5	15.625 – 31.25
D6	7.8125 – 15.625
D7	3.9062 – 7.8125
A7	0 – 3.9062

Table 4.1 shows the number of decomposition levels chosen for the ECoG signal based on the dominant frequency components of the signal. The sampling frequency of the ECoG signals used for analysis was 1000Hz. The levels shown in Table 4.1 were chosen such that each

frequency sub-band retains frequencies necessary for classification of the signal. The ECoG signals were decomposed into D1-D7 detailed coefficients and final approximation coefficient A7.

The signal was decomposed using the functions from the wavelet toolbox in MATLAB. Daubechies filter ("db6") was used for obtaining the detailed and approximation coefficients. Several other filters such as Morlet filter, or Haar filter are used for wavelet transform of the physiological signals. The Daubechies filter was used in this thesis to follow the paper on which this thesis is based. This method of wavelet decomposition was applied to epileptic (Set E - Appendix C) and non-epileptic (Set NE - Appendix C) data of clinical ECoG data of Subject 2.

4.5. Feature Matrix

After the wavelet coefficients were reconstructed, the detailed coefficients D5, D6, D7 and approximate coefficient A7 were used to generate the "Feature matrix". Based on Table 4.1, these coefficients constitute the range of frequency bands from 0 – 31.25Hz. As the ECoG signal analyzed in this thesis only looks at the frequency spectrum below 30Hz, the coefficients D1-D4 were not included in the analysis allowing the feature matrix to be reduced.

Feature matrix is the representation of the signal using statistics over the set of wavelet coefficients D5-D7 and A7. The fifteen statistical features representing the time-frequency distribution of the ECoG signal for each sub band (as described in Table 4.1) are:

1. Mean of the absolute values of the wavelet coefficients: Four features representing the frequency distribution of the signal.
2. Average power of the wavelet coefficients: Four features representing the frequency distribution of the signal.

3. Standard deviation of the wavelet coefficients: Four features representing the amount of change in the frequency distribution
4. Ratio of the mean values between adjacent sub bands: Three features representing the relative change in the frequency distribution

These statistical features were chosen in this thesis to emulate the feature extraction method explained in the paper written by Subasi and Gursoy[2]. The authors chose these statistical measures because promising results were observed using these statistical measures with classification of lung sounds[12]. Each 20 second data segment, sampled at 1000Hz, was further divided into subsegments of 0.5seconds or 500 samples and an overlap of 0.25 seconds or 250 samples between adjacent subsegments was used to calculate the statistical features for the feature matrix[11]. So each 20 second data segment resulted in a feature matrix with 79 rows, one for each 0.5 second subsegment and 15 columns for each of the 15 statistical features.

To include the high frequency components observed in the ECoG signals during an epileptic event, another feature matrix was created to include coefficients D1-D3. This feature matrix included high frequency bands between 60Hz and 500Hz for performing analysis. The steps involved in creating the feature matrix as explained in this section were followed to obtain the high frequency feature matrix as well.

4.6. Generating Model

To evaluate the efficacy of SVM for differentiating epileptic from non-epileptic signals, four training models were implemented; each model amending the feature matrix used to train the SVM classifiers. To evaluate the efficiency of each model in classification of the signal, the sensitivity and specificity of each model was compared.

4.6.1. Model 1: Principal Components

The feature matrix created from the wavelet coefficients had 79 time points for a total of 15 statistical features. The efficacy of training SVM using data where the feature matrix had been reduced using PCA was explored in this model. PCA was used to generate principal components that are orthogonal to each other ensuring that no redundant information exists. The *pca*() function in MATLAB was used to identify the principal components that explained the total variance in the original feature matrix. The principal component analysis constructed independent new variables that are linear combinations of the original variables.

As the pairwise correlation between the features showed substantial difference in variance of different columns, the inverse variance of the data was used as weights for computing the principal components. The *pca*() function computed the coefficients of the principal components and transformed them so they are orthonormal. The *score* matrix generated by the function contained the coordinates of the new principal axes and organized them in ascending order from the first principal component coordinates to the last principal component coordinates. The number of columns in the *score* matrix had equal number of columns as the feature matrix because *pca*() generates the same number of principal components as the number of categories in the data.

As each principal component corresponds to the percentage variance explained by that component with respect to the original data, the scree plot provided a visual representation of the percentage of variance explained for each principal component[24].

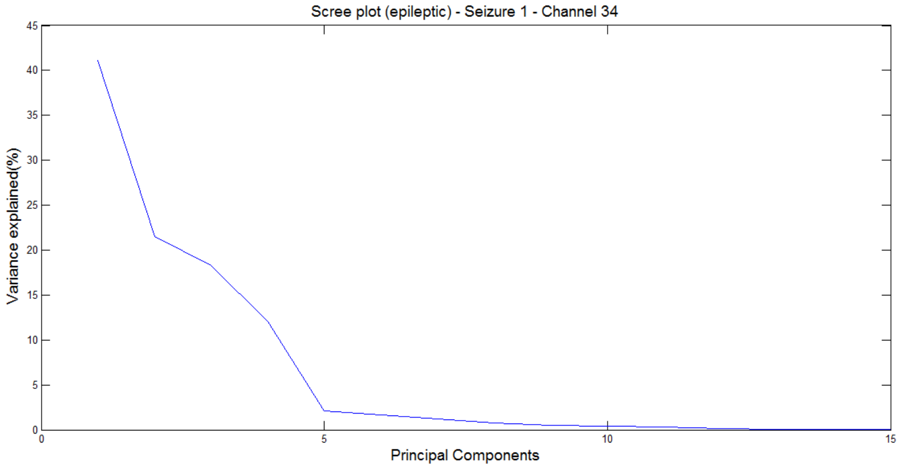


Figure 4.3 Scree plot of an epileptic seizure

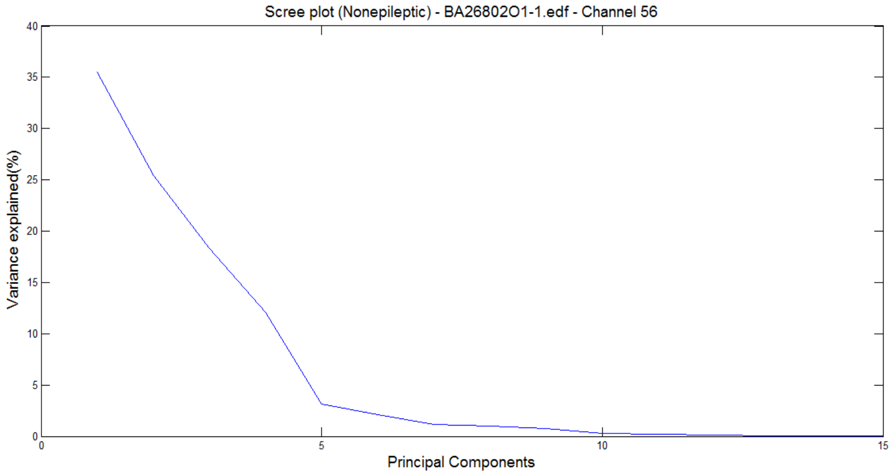


Figure 4.4 Scree plot of non-epileptic data

Based on scree plots of multiple epileptic and non-epileptic channel data, three principal components explained 80% or more variance in the original data and were taken to create the training matrix to train SVM classifier I.

4.6.2. Model 2: Independent Components

In the second model number of features in the feature matrix was reduced using ICA. To identify the independent components an algorithm based on joint diagonalization of cumulant matrices called JadeR was used[25].

The first step of this statistics based algorithm, created by Cardoso and Souloumiac, to use with real data, involved whitening of the feature matrix[23]. This was done to obtain a new mixing matrix that is orthogonal with values that are uncorrelated and with variances equal to unity. The next step involved estimation of cumulant matrices and to find rotation matrix such that the cumulant matrices are diagonal before estimating the independent components. This step used joint diagonalization method as the optimization technique to obtain the objective function that relates to variable independence[23]. Scree plots such as shown in Figure 4.3 were used to identify that the optimum number of independent components needed is three, so the first three independent components of the feature matrix computed using the JadeR were obtained to create the training matrix to train SVM classifier II.

4.6.3. Model 3: Principal + Independent Components

This model is a combination of models 1 and 2 where three principal components and three independent components are used together to create the training matrix to train the SVM classifier III.

4.6.4. Model 4: Feature Matrix

For this model, the entire feature matrix was used as the training matrix to train the SVM classifier IV. It should be noted that as only 15 features are present in the Feature matrix, training an SVM classifier using all of the features would likely provide better sensitivity and specificity.

4.6.5. Obtaining SVM Classifiers

SVM algorithm classifies the two classes (epileptic and non-epileptic) by finding an optimal hyperplane with the largest margin that separates the data points of the classes. The support vectors are the data points closest to this hyperplane and on the margins of the border separating the classes. For non-separable data, a softer margin is identified that separates many if not all points.

The basic premise of SVM is to produce a classifier (based on the training matrix) which predicts the classification of the test data set given only the test data attributes. Training an SVM classifier was achieved in three distinct steps; first step was to train the machine classifier, the second step was to classify the data using the classifier and the third step was to tune the classifier for optimal classification. *fitcsvm*() is a MATLAB function that is available from the statistical toolbox for training SVM. The training matrix containing the epileptic and non epileptic data along with the class matrix containing the classification (class = -1 for epileptic and class = +1 for non-epileptic) was used to train the SVM. For all the aforementioned models, the data matrix changed based on the number of columns used for creating the training matrix.

The radial basis function (RBF) kernel which is a Gaussian function was used for training SVM classifiers. This function non-linearly maps the data points onto a higher dimensional space where it becomes close to linearly separable under the change of variables. The resulting SVM trained classifier contains the optimized parameters that helped classify the test data set. *predict*() function from MATLAB along with the classifier was used to classify the test data set, where each row corresponds to a new time point. The test data set contains all 79 time points for the 20 second data segment for each channel included in the test data. SVM classifier then classifies each time point for the channel as epileptic or non-epileptic. In order to identify whether the entire channel can be classified as epileptic or non-epileptic, the preponderance of classification was used. For the channel to be classified as epileptic, 51% of the 79 time points are required to have a classification as epileptic (class = -1) and for the channel to be classified as non-epileptic, 51% of the 79 time points are required to have a classification as non-epileptic or (class = +1).

In order to tune the classifier for optimal performance, it was necessary to identify the best parameters for σ and γ where σ is the penalty parameter and γ is the kernel parameter. This was accomplished using a ten-fold cross-validation to identify the parameters where multiple σ and γ values are used to compute the best cross validation accuracy. In ten-fold cross-validation, the training data was partitioned into ten subsets. The classifier was trained using nine of the subsets of which the tenth one was used as a test set to obtain a score that corresponds to the percentage of data that was classified correctly. This process was repeated ten times with each subset as a test set exactly once. The ten test scores were then averaged and the classifier with the highest test score was chosen[12]. Cross-validation was used for comparing all the SVM classifiers possible to identify the best classifier. After identifying the kernel parameters, the SVM classifier was trained and the test data was classified. The process of obtaining classifiers described above was repeated for each model classifier with its corresponding training matrix and test data set. The classifiers obtained were classifier I for model 1, classifier II for model 2, classifier III for model 3, and classifier IV for model 4.

5. Results

Figure 5.1 shows the ECoG data of ictal discharge seen in epileptic data and Figure 5.2 shows ECoG data from the non-epileptic files that shows no ictal activity.

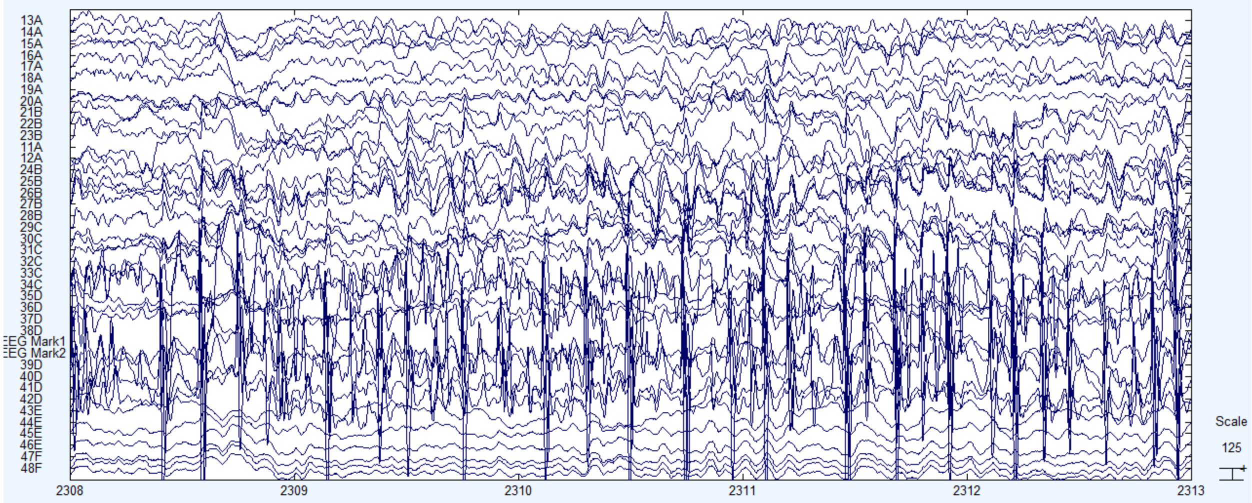


Figure 5.1 Epileptic ECoG data

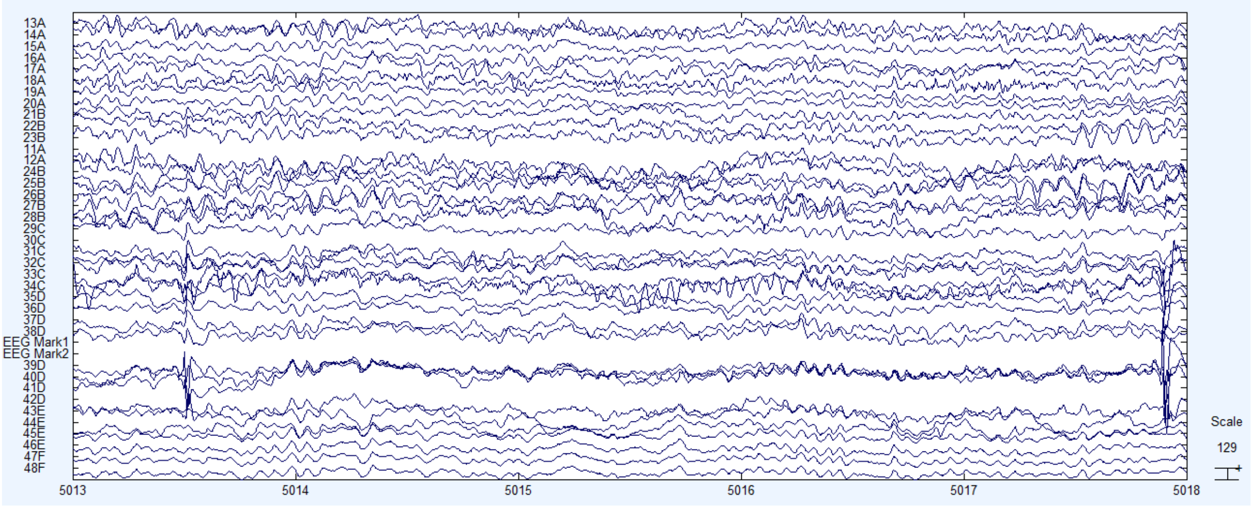


Figure 5.2 Non-epileptic ECoG data

As explained in section 4, an algorithm was written to convert 20 second data segment for a channel into a feature matrix containing 79 time points and 15 features. Figure 5.4 shows the wavelet decomposition of the epileptic data during seizure #2 on Channel 34 (Figure 5.3) and Figure 5.6 shows the decomposition of the non-epileptic data for the same channel (Figure 5.5).

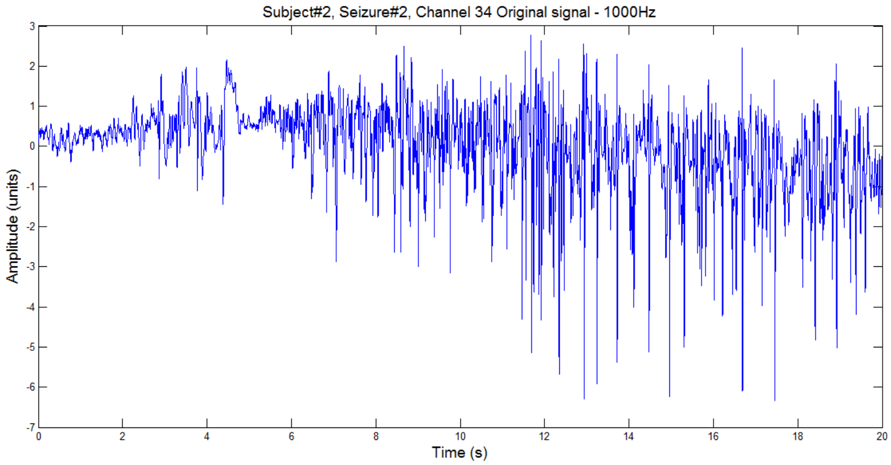


Figure 5.3 Epileptic data in Channel 34

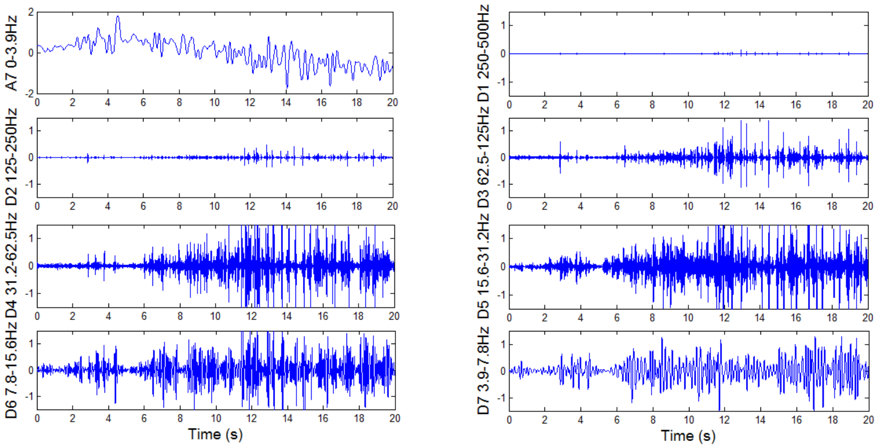


Figure 5.4 Wavelet decomposition of epileptic data on Channel 34 of Subject #2

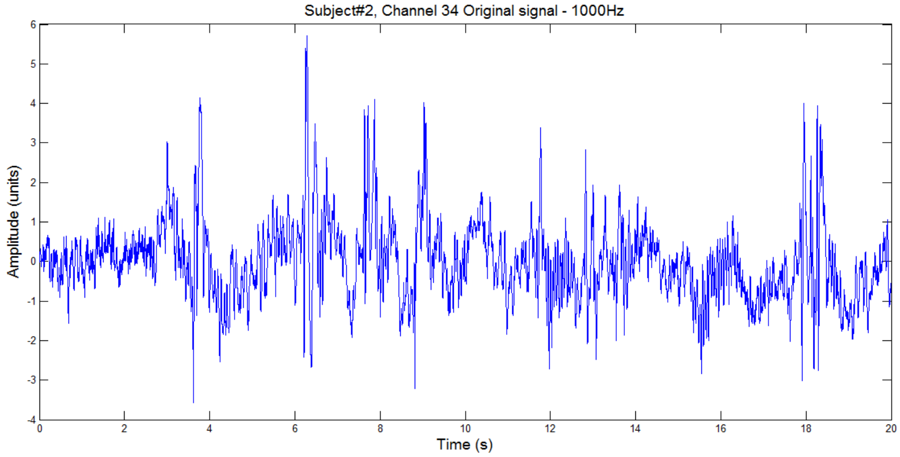


Figure 5.5 Non-epileptic data in Channel 34

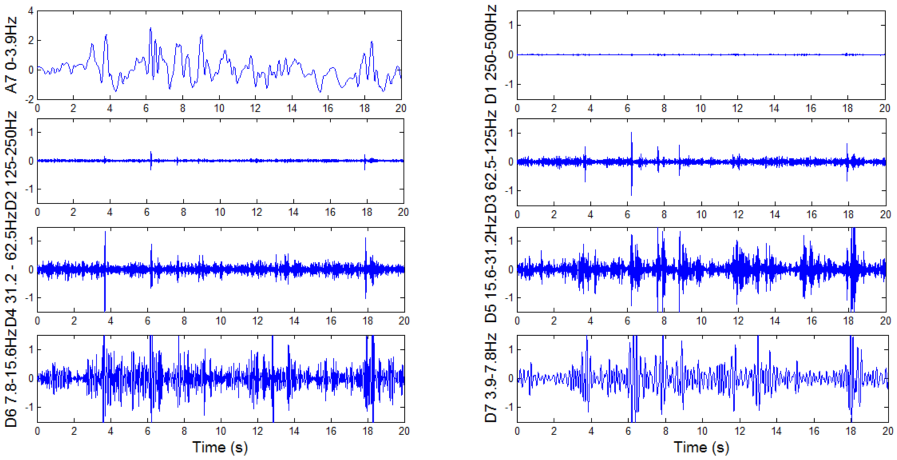


Figure 5.6 Wavelet decomposition of non-epileptic data on Channel 34 of Subject #2

The extracted coefficients provided a representation of energy distribution of the ECoG signal in time and frequency. Each coefficient was reconstructed using *wrcoe*f() function from the wavelet toolbox. The feature matrix obtained from these coefficients had 79 time points and 15 features as explained in section 4.5. Figure 5.7shows each channel being classified as epileptic. Notice that out of 79 time points preponderance (greater than 51%) shows that the channel is epileptic as the SVM classifier classifies epileptic data as -1 and non-epileptic data as +1. Figure 5.8 shows the channel as being non-epileptic. Again the preponderance of time points is shown as non-epileptic which is +1.

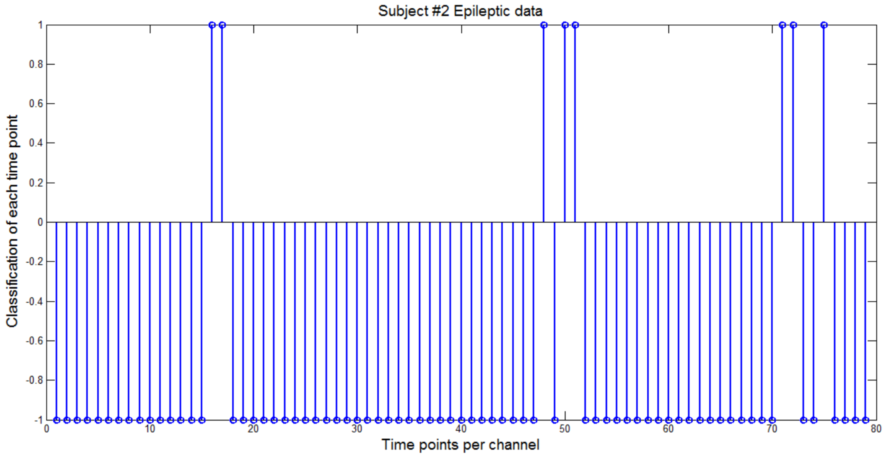


Figure 5.7 Preponderance of time points showing the channel as epileptic

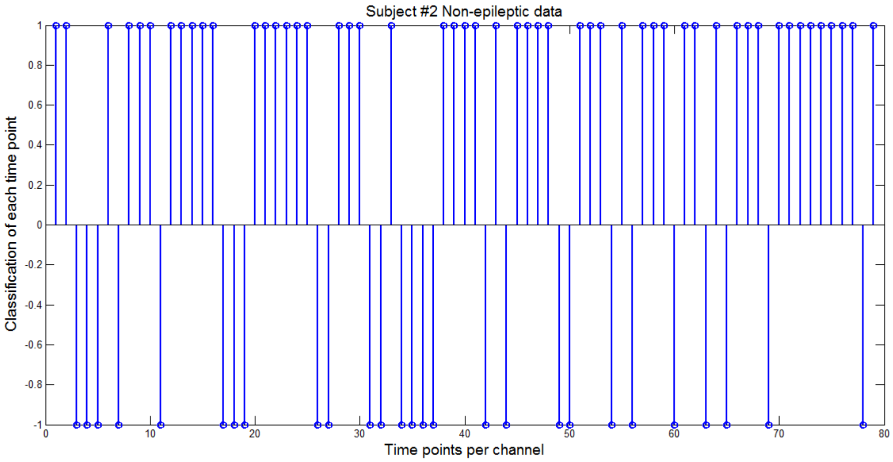


Figure 5.8 Preponderance of time points showing the channel as non-epileptic

Using this algorithm, a training matrix for each model was created using data from 12 seizures experienced by Subject #2. The training matrix contained epileptic data from a total of 253 channels with 20000 time points and non-epileptic data from 190 channels with 15000 time points. The feature matrix for each of the channels was computed individually using the algorithm and all the feature matrices were combined to create the training matrix. It should be noted that the training matrix contains more epileptic channels than non-epileptic channels; the reason was to increase the sensitivity of the SVM classifiers in classifying the epileptic from non-epileptic data.

Table 5.1 Class distribution of the channels in the training and data sets

Class	Training Matrix	Test Data set
Epileptic	253 channels	50 channels
Non-epileptic	190 channels	50 channels

The test data set was created using the epileptic and non-epileptic data of Subject #2 not used in the training matrix. This test data set was used with each model to identify the models' sensitivity and specificity. Table 5.1 shows the number of epileptic and non-epileptic channels chosen for the training matrix and test data set.

5.1.1. Testing using Model 1

For this model only three principal components were chosen to train the SVM classifier I and the number of channels used to train and test were as shown in Table 5.1. While the model was able to correctly identify all the epileptic channels, it also identified all the non-epileptic channels as epileptic. This allowed the sensitivity of the model to be 100% but the specificity of the model was zero. Table 5.2 shows the time the model took for training classifier I and classifying the test data set using classifier I. Figure 5.9 shows the classification of a seizure performed by classifier I and due to low specificity, almost all channels have been marked as epileptic. The channels of interest for this seizure were 34,39-41 only.

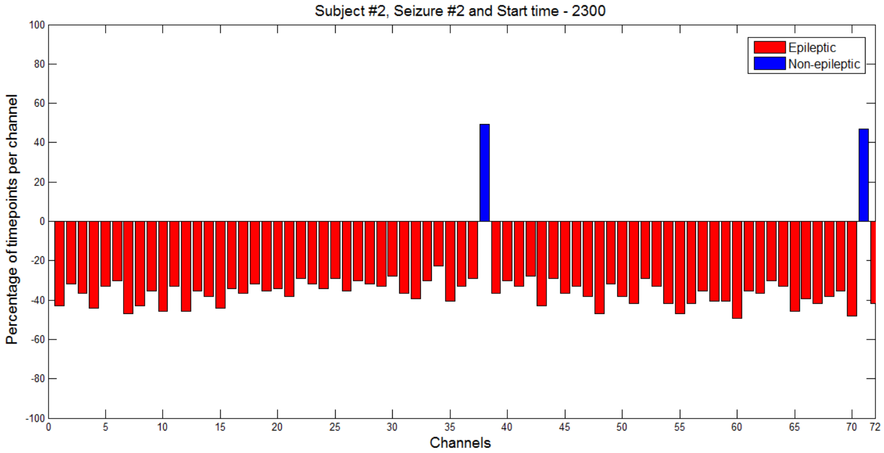


Figure 5.9 Channel classification performed by classifier I

5.1.2. Testing using Model 2

In this model, three independent components were used to train and test the SVM for performing the classification. This model behaved in the same way as model 1 where it correctly identified all the epileptic channels, identified by the epileptologist, correctly but it also showed zero specificity. Table 5.2 shows time the model took for training classifier II and classifying the test data set using classifier II. Figure 5.10 shows the classification of the same seizure performed by classifier II and as the low specificity exhibited with the test data set, only 13 channels have been marked as non-epileptic.

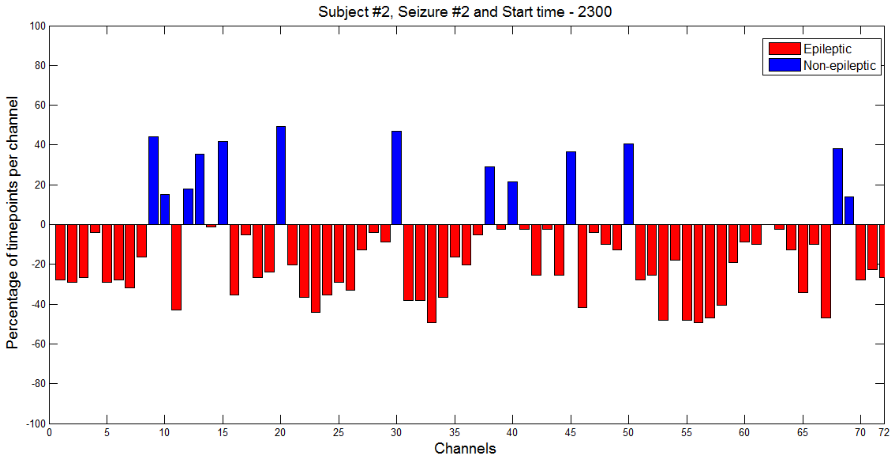


Figure 5.10 Channel classification performed by classifier II

5.1.3. Testing using Model 3

The training matrix for this model was created with the first three principal components using PCA and the first three independent components using ICA. This model showed results that were better than both model one and model two. While the sensitivity of the model was lower compared to the previous models, the specificity improved. Table 5.2 shows the sensitivity and specificity values of the model along with the time the model took to train the SVM classifier III.

This model was used to classify the channels of epileptic data during a seizure and non-epileptic data where no ictal activity was observed by the epileptologist. While the model identified the channels chosen by the epileptologist as the channels of interest, several other channels were chosen as well (Figure 5.11). Similarly in the non-epileptic data, classifier III identified several channels that were shown as epileptic (Figure 5.12).

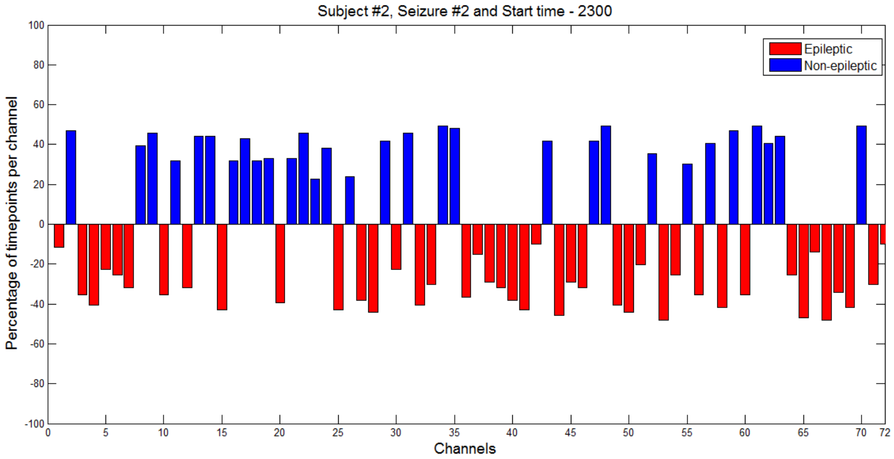


Figure 5.11 Channel classification performed by classifier III

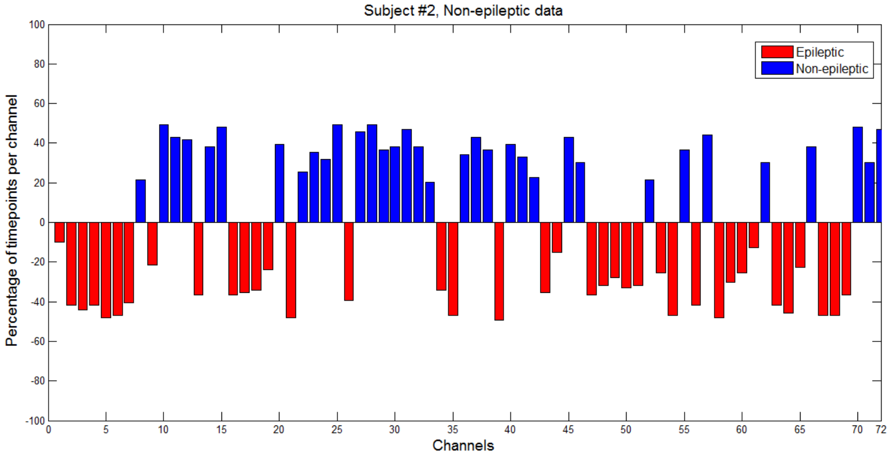


Figure 5.12 Channel classifying non-epileptic data using classifier III

5.1.4. Testing using Model 4

For this model no dimensionality reduction methods were used to reduce the feature matrix and the training matrix comprised of all fifteen features. The sensitivity of the model was 98% and the specificity was 80% and both these values were observed to be the best among all the four models. The time taken for training classifier IV for this model was considerably more than the time taken by other models for training their classifiers (Table 5.2).

Figure 5.13 shows the classification of seizure #2 performed by classifier IV. The classifier correctly identifies the channels chosen by the epileptologist while pick four additional channels as epileptic. Figure 5.14 shows the classification of a non-epileptic data. Please note that this is the same data used for classification by model 3.

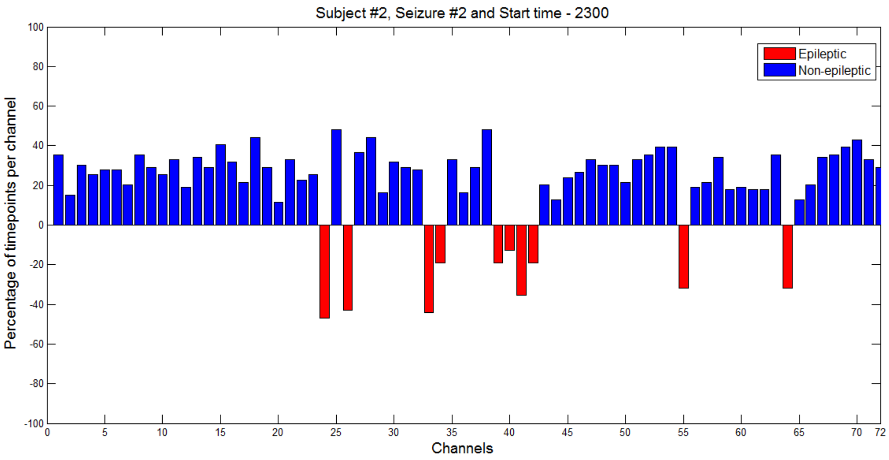


Figure 5.13 Channels classified using classifier IV (0-30Hz)

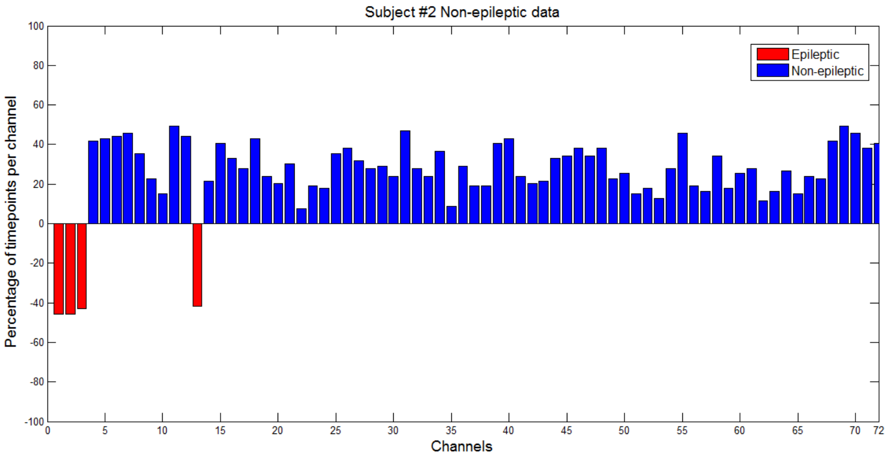


Figure 5.14 Channel classifications of non-epileptic data using classifier IV (0-30Hz)

Table 5.2 Sensitivity and specificity of the four models

	Sensitivity	**Specificity**	**Time to train classifier**
Model 1	100%	0%	17 minutes
Model 2	100%	0%	18 minutes
Model 3	80%	46%	44 minutes
Model 4	98%	80%	70 minutes

In order to observe the classification of data using just the high frequency components observed in the ECoG signals, the second feature matrix created using the wavelet coefficients D1-D3 was used to train and test classifier IV.

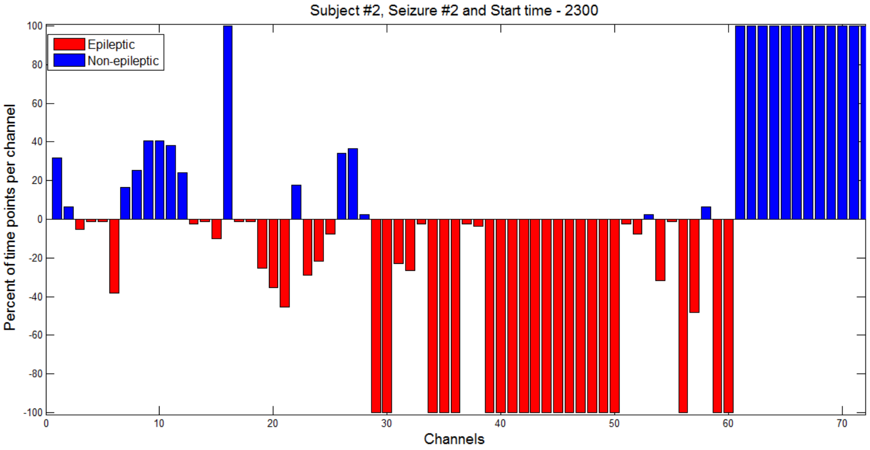


Figure 5.15 Channels classified using the high frequency classifier IV (60-500Hz)

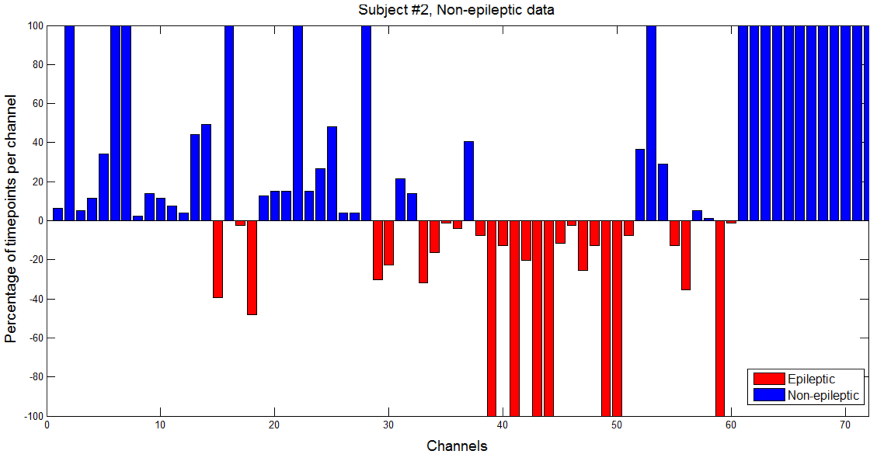


Figure 5.16 Channel classifications of non-epileptic data using high frequency classifier IV (60-500Hz)

Figure 5.15 shows the classification of all the channels for the epileptic data from Subject # 2. Figure 5.16 shows the classifications of all the channels from non-epileptic files of Subject # 2. Please note that the training data used for the training the high frequency classifier IV was created using the wavelet coefficients D1-D3 which was different from the training data created using coefficients D5-D7 and A7.

6. Discussion

The test data set created using 50 epileptic channel and 50 non-epileptic channel data was able to provide the sensitivity and specificity of each model studied in this thesis. The use of component analysis methods to reduce dimensionality of the feature matrix used to train the SVM classifier to differentiate between the epileptic and non-epileptic signals were explored. The objective of this thesis was the classification of data along with identifying the optimal technique that achieved the differentiation with more accuracy.

After selecting the data length of extraction for analysis as twenty seconds, the wavelet transform was used to find the wavelet coefficients of the signal segment. To obtain good time and frequency resolutions, the signal was decomposed into frequency sub bands as shown in Table 4.1. Primarily the frequency spectrum of interest for ECoG signals was in the ranges 0-30Hz, the detailed coefficients D5-D7 and approximate coefficients A7 were used to create the feature matrix using statistical features. These features are mean of the absolute values, average power, standard deviation, and ratio of the mean in order to obtain the frequency distribution along with change in the frequency distribution in each sub band. For creating the feature matrix, the signal was divided into 0.5 second subsegments with an overlap of 0.25 seconds and stationarity was assumed for the subsegment. Each channel of data was represented in the feature matrix as a total of 79 time points with a total of 15 features.

The feature matrix was then used with the PCA algorithm to obtain principal components and ICA algorithms to obtain the independent components for creating the training matrix for models 1, 2 and 3. The number of principal and independent components used for analysis was identified from the scree plot of multiple epileptic and non-epileptic channel data. The inflection point was observed to be near the 4th and 5th principal component (Figure 4.3). After visually analyzing several scree plots for both epileptic and non-epileptic data, it was observed that the first three principal components explained more than 80% - 90% variation in the signal. Hence, only three principal and independent components were chosen for creating the training matrices. The other reason for choosing only three principal components was to reduce the training time of the SVM classifiers.

In their work, Subasi and Gursoy suggested that using PCA and ICA to reduce the feature dimension resulted in faster and more accurate classification. They also suggested that using the feature matrix as a whole to train the SVM classifiers lead to longer runtime as well as reduced accuracy. The authors used EEG data recorded using scalp electrodes and their epileptic data contained only seizure activity and their non epileptic data was taken from healthy individuals with no epileptic activity. In this thesis, the data used for analysis is ECoG from patients who were diagnosed with epilepsy. The epileptic data used contained seizure activity as noted by the epileptologist. As it is impossible to obtain ECoG data from healthy individuals, the non-epileptic data came from files that the epileptologist found no epileptic activity.

Significant differences exist in data as well as methods used for creating the feature matrix between the algorithm of Subasi and Gursoy and the one used in this thesis. The data segment along with the subsegment chosen for wavelet transform is different in both the studies. The authors also chose the frequency bands during feature extraction to only include frequencies between 0 -21.7Hz. In this thesis the initial analysis and calculation of sensitivity and specificity of all models were calculated using the frequencies between 0-30Hz. After obtaining the best classifier among the four models tested, a high frequency analysis was performed using the high frequency range between 60-500Hz.

In order to account for the changes between the two studies, it was necessary to evaluate using the entire feature matrix to train SVM classifiers and, therefore, model 4 was included for analysis. Just as the training data, the test data for each model was created using the feature matrix of the epileptic and non-epileptic data excluded from the training matrix. Hence, SVM classified each of the 79 time point as epileptic or non-epileptic. In order to classify the channel as epileptic or non-epileptic, 51% preponderance of classifications was calculated for all the 79 time points of that channel.

Results of the analysis using 0-30Hz showed that out of the four models, model 4 and classifier IV was best at classifying the test data set with a sensitivity of 98% and specificity of 80%. But model 4 also had the longest runtime when compared to the other three models. Model 3 fared better than models 1 and 2 in terms of sensitivity and specificity.

All the four model classifiers were used to classify all 72 channels of seizure #2 for a 20 second time segment. Results show that classifier I and II had subpar performance and classifier III also classified a lot of non-epileptic channels as epileptic. Classifier IV performed optimally by choosing all the channels that the epileptologist highlighted. The other channels chosen with the epileptic and non-epileptic data could have been for the following reasons:

1. The channels chosen showed subclinical epileptic discharge that lasted for a very short duration and did not cause any clinical symptom.

2. The algorithm misclassified it as epileptic even though no epileptic discharge was present.

3. The epileptic discharge was just starting at that time and would have evolved or dissipated in the next 20 seconds.

As there are high frequency components present in the EEG data during ictal activity, the analysis using all 72 channels of Seizure # 2 for a 20 second time segment was carried out using the training matrix created with the high frequency wavelet coefficients D1-D3. Results show that while the channels highlighted by the epileptologist were picked by the classifier as well several other channels were classified as epileptic. This was also observed when the high frequency classifier was used to classify the channels from the non-epileptic data. The classifications done by classifier IV and the high frequency classifier IV for the same data files were visually analyzed. It was observed that while all the channels highlighted as epileptic were picked by both the classifiers, the high frequency classifier has an increased number of false positives as several other channels were picked as epileptic.

To further evaluate the algorithm and the classifications performed by the classifiers, it is imperative to analyze each and every misclassification by analyzing the raw EEG data with the experts.

7. Future Work

The results in this thesis suggest that using the entire feature set to train the SVM allows for higher sensitivity and specificity but accuracy of SVM classification is highly dependent on the training data. The possibility of using PCA and ICA to reduce the dimensions of the data were explored in this thesis but not all combinations of the number of principal components and independent components to be used, were explored. To perform an exhaustive comparison between various techniques of component analysis and training an SVM classifier, it is imperative to try all combinations and compare the results. ICA is used with scalp EEG data with more efficiency as the data recorded has influence from independent sources of neural activity captured by the electrode. Hence, this algorithm should be tested using the scalp EEG data as well.

The sensitivity and specificity of each model was determined based on the comparison between the channels that the SVM classified as epileptic and the channels that the epileptologist classified as epileptic. The efficacy of the models and the algorithm used in this thesis should be verified by the epileptologist. Such verifications were not possible with the data used in this thesis due to time constraints.

As SVM trains and performs better with the amount and quality of data used, it is possible to fine tune the ways that SVM is trained. While some information, such as the length of data used for analysis, window size for obtaining the wavelet coefficients and the number of statistical features applied to the chosen coefficients, were empirically chosen for optimal performance based on the present knowledge of the data as well as the recommendation of the paper written by Subasi and Gursoy, there is certainly a need to analyze other ways to increase the efficiency of SVM in performing classifications.

The availability of additional patient ECoG data provides promising avenues for training and testing the models using data from these patients. Since the time taken to train the SVM classifier is considerably smaller than other machine learning options such as artificial neural networks (ANNs) and since the possibility of over-fitting the model is much lower, this additional data can be used to help increase the robustness of the machine algorithm.

8. Conclusion

This thesis contributes to the detection of epilepsy by providing an automated classification method that allows the data to be sorted as epileptic or non-epileptic. While several commercially available software packages exist for assisting medical professionals in making this distinction, a black box that can intuitively classify the available data and learn to improve its performance, is more useful. This black box or support vector machine can be used to classify the data as either epileptic or non-epileptic based on currently available data. With clinical ECoG single patient data for training and testing, the sensitivity and specificity of the SVM classifier was 98% and 80%, respectively, for the model that used the feature matrix to train SVM.

SVM runtime for training the classifiers is much faster when compared to other machine learning algorithms and the testing time was observed to be less than five seconds for classifying all 72 channels of a twenty second data segment. This allows the algorithm to be used for obtaining real time classification of data while recording. While, identification of epileptogenic zone will require the expert opinion of an epileptologist, the machine learning algorithm can be used to assist in signal analysis and classification.

9. Bibliography

1. Sanei, S. & Chambers, J. A. *EEG Signal Processing*. (Wiley, 2008).
2. Subasi, A. & Ismail Gursoy, M. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Syst. Appl.* **37,** 8659–8666 (2010).
3. Varsavsky, A., Mareels, I. & Cook, M. *Epileptic seizures and the EEG measurement, models, detection and prediction*. (CRC Press, 2011). at <http://dx.doi.org/10.1201/b10459>
4. *Introduction to epilepsy*. (Cambridge University Press, 2012).
5. Hill, N. J. *et al.* Recording human electrocorticographic (ECoG) signals for neuroscientific research and real-time functional cortical mapping. *J. Vis. Exp. JoVE* (2012). doi:10.3791/3993
6. Brain Basics: Know Your Brain: National Institute of Neurological Disorders and Stroke (NINDS). at <http://www.ninds.nih.gov/disorders/brain_basics/know_your_brain.htm>
7. Miller, J. W. & Goodkin, H. P. *Epilepsy*. (2014). at <http://dx.doi.org/10.1002/9781118456989>
8. Acharya, U. R., Vinitha Sree, S., Swapna, G., Martis, R. J. & Suri, J. S. Automated EEG analysis of epilepsy: A review. *Knowl.-Based Syst.* **45,** 147–165 (2013).
9. Semmlow, J. L. *Biosignal and Medical Image Processing*. (CRC Press, 2004).
10. Quiroga, R. Quantitative analysis of EEG signals: Time-frequency methods and Chaos theory. (1998). at <http://scholar.googleusercontent.com/scholar?q=cache:C8qm8cFl0pwJ:scholar.google.com/+quantitative+analysis+of+EEG+signals&hl=en&as_sdt=0,23&as_vis=1>

11. Tzanetakis, G., Essl, G. & Cook, P. Audio analysis using the discrete wavelet transform. in *Proc. Conf. in Acoustics and Music Theory Applications* (2001). at <http://masters.donntu.edu.ua/2010/fknt/prylepskyi/library/tzanetakis.html>
12. Kandaswamy, A., Kumar, C. S., Ramanathan, R. P., Jayaraman, S. & Malmurugan, N. Neural classification of lung sounds using wavelet coefficients. *Comput. Biol. Med.* **34,** 523–537 (2004).
13. Subasi, A. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Syst. Appl.* **32,** 1084–1093 (2007).
14. Principal Component Analysis 4 Dummies: Eigenvectors, Eigenvalues and Dimension Reduction. *georgemdallas* at <http://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>
15. Ng, A. CS229 Lecture notes: Principal Component Analysis. *CS229 Lect. Notes* **1,** (2000).
16. Ng, A. CS229 Lecture notes: Independent Component Analysis. *CS229 Lect. Notes* **1,** (2000).
17. Hyvärinen, A. & Oja, E. Independent component analysis: algorithms and applications. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **13,** 411–430 (2000).
18. Fielding, A. H. *Cluster and Classification Techniques for the Biosciences.* (Cambridge University Press, 2006). at <http://public.eblib.com/EBLPublic/PublicView.do?ptiID=281729>
19. Ren, J. ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowl.-Based Syst.* **26,** 144–153 (2012).
20. Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. (Cambridge University Press, 2000).

21. Hsu, C.-W., Chang, C.-C., Lin, C.-J. & others. *A practical guide to support vector classification*. (2003). at <https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf>

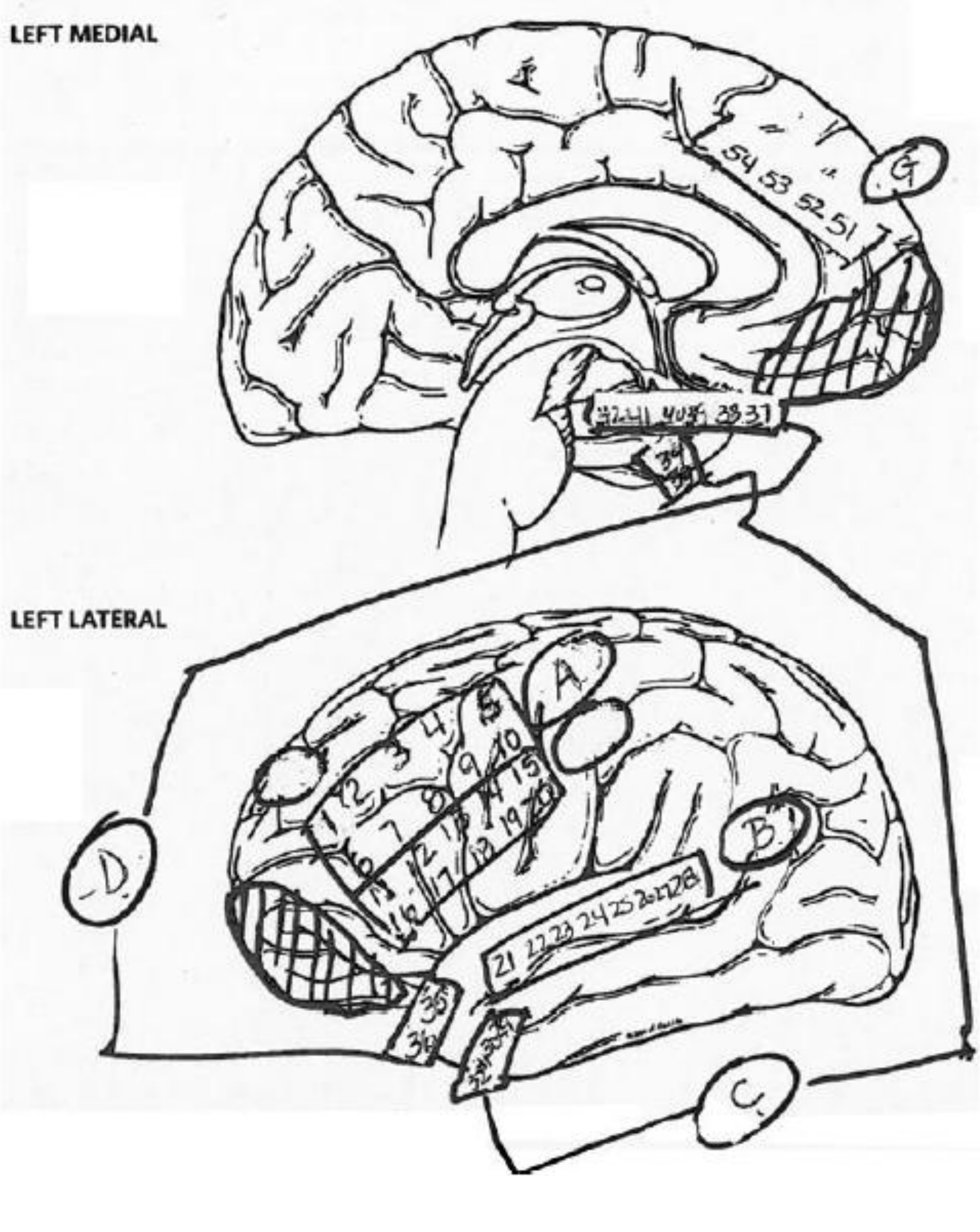
22. EEGLAB - Open Source Matlab Toolbox for Electrophysiological Research. at <http://sccn.ucsd.edu/eeglab/>

23. Cardoso, J.-F. High-Order Contrasts for Independent Component Analysis. *Neural Comput.* **11,** 157–192 (1999).

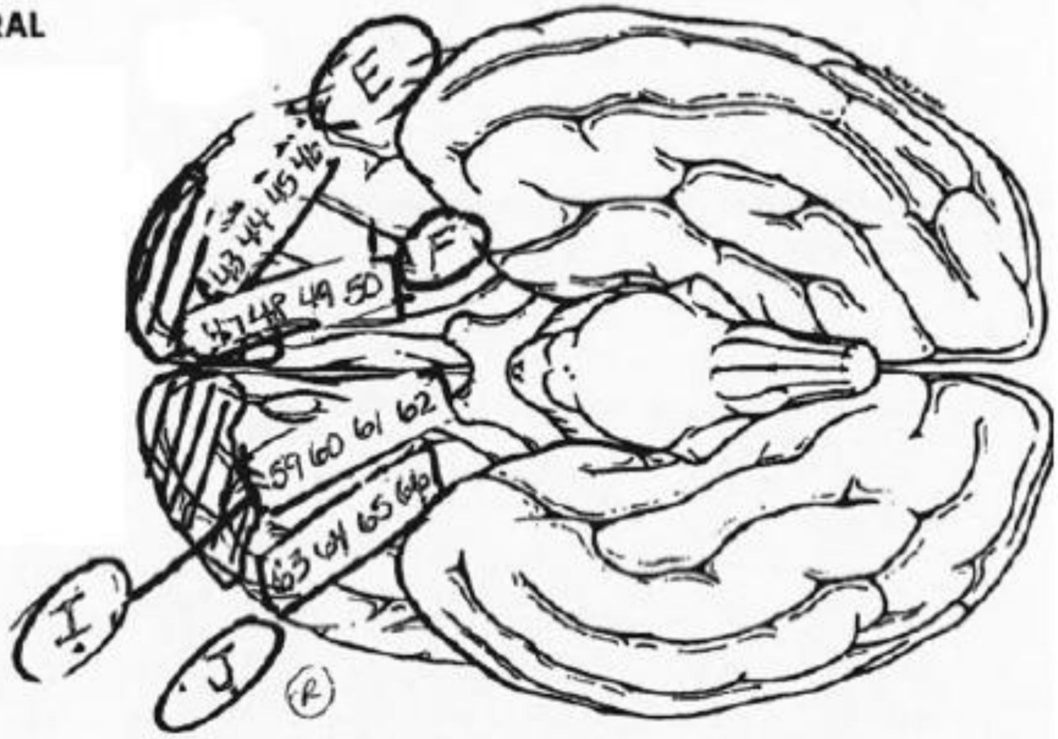
24. Feature Transformation - MATLAB & Simulink. at <http://www.mathworks.com/help/stats/feature-transformation.html#f75476>

25. Blind source separation and Independent component analysis. at <http://perso.telecom-paristech.fr/~cardoso/guidesepsou.html>

10.Appendix A: Electrode Locations Subject #2



SUBTEMPORAL



SUPERIOR

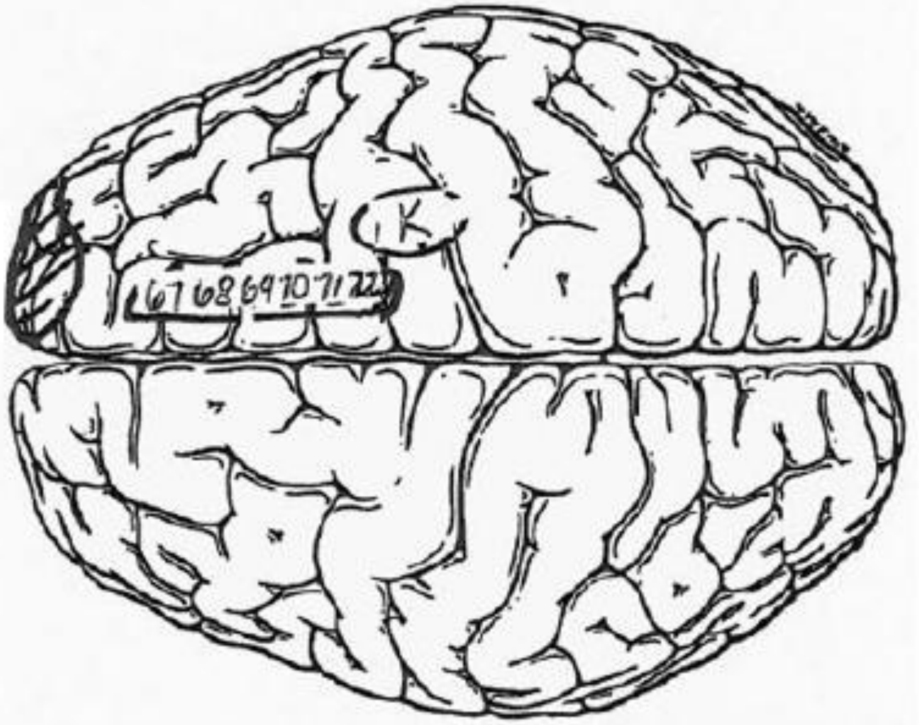


Table 10.1 Electrode Grids

Grid	Electrode Number
A	1-20
B	21-28
C	29-34
D	35-42
E	43-46
F	47-50
G	51-54
H	55-58
I	59-62
J	63-66
K	67-72

11.Appendix B: List of epileptic and non-epileptic files used for analysis

Table 11.1 List of epileptic and non-epileptic files

Non-epileptic files (Set NE)	Epileptic files (Set E)
BA26802O_1-1.edf	BA26802N_1-1.edf
BA26802S_1-1.edf	BA26802P_1-1.edf
BA26802U_1-1.edf	BA26802Q_1-1.edf
BA26802V_1-1.edf	BA26802R_1-1.edf
BA26802W_1-1.edf	BA26802T_1-1.edf
BA26802Y_1-1.edf	BA26802X_1-1.edf
BA26802Z_1-1.edf	BA26802X_1-1.edf
BA26803A_1-1.edf	BA26802X_1-1.edf
BA26803B_1-1.edf	BA268030_1-1.edf
BA26803D_1-1.edf	BA268031_1-1.edf
BA26803E_1-1.edf	BA26803C_1-1.edf
BA26803F_1-1.edf	BA268049_1-1.edf
BA26803G_1-1.edf	BA26804K_1-1.edf
BA26803H_1-1.edf	BA26804L_1-1.edf
BA26804A_1-1.edf	BA26804M_1-1.edf

12.Appendix C: Matlab Code : Epileptic

```
%###########################################################################
%This program performs the following actions in a sequence:
%   - Add path to the files containing the data from the excel
%   - Get information of the start time and stop time for each seizure
%   - Read the .edf file of the patient
%   - extract 20 second data for epileptic channels
%
%Some initial steps for loading the EEG data for analysis has been adapted
%from James Gurisko's thesis.
%
% Priya Balasubramanian
% Created on 07/23/2014
% Last updated on 11/24/2014
%###########################################################################
clear all;
clc;
close all;

%--------------------------------------------------------------------------
% User entered parameters
%
% The information entered below will change for each patient and each
% seizure. The sampling frequency as well as the epileptic and nonepileptic
% channels being considered will change based on the patient and seizure as
% well. The length of the data remains as 20 seconds for each dataset being
% analyzed.

Filename = 'DataNotesP.xlsx'; %File containing the patient data
Directory = 'C:/eegData/SH-EEG/'; % Directory containing the patient data
SeizureNumber =2; % Seizure number being analyzed
Nextset = 0;% to collect the next 20 seconds worth of data
DataLength = 0; % Length of the data in seconds
Fs = 1000 ; % sampling frequency in Hz
E1begin = 1;
E1end = 74; % Range of first set of epileptic channels
% E1begin = 4;
% E1end = 6;
% E2begin = 33;
% E2end = 46;
%E2begin = 41;
%E2end = 74; % Range of second set of epileptic channels

%--------------------------------------------------------------------------
```

```
% Add path to files
%
addpath('C:/eegData/Priya','C:/eegData/MATLAB/WOSSPA_Mathworks_v2',...
    'C:/eegData/Priya/eeglab10.2.2.4b');
% Use addpath to add the necessary folders

%--------------------------------------------------------------------------
%Read information of the seizure from the excel file
%
% Read the start times for each seizure
Hour = xlsread(Filename,1,'O:O');
Minute = xlsread(Filename,1,'P:P');
Second = xlsread(Filename,1,'Q:Q');

% Calculate the absolute start time for each seizure
StartSeizure = ((((Hour.*60)+Minute).*60)+Second);

%Read the file name corresponding to each seizure from the excel file
[Num,FileEDF,Raw] = xlsread(Filename,'I:I');
size(FileEDF)
clear Num;
clear Raw;

%Create a single string for importing data
%strcat combines the directory with the .edf filename to create a single
%string for importing patient data

FilenameCombined = cell(16,1);

for i = 1:16,
    FilenameCombined(i,1) = strcat(Directory,FileEDF(i+1,1));
end

%--------------------------------------------------------------------------
% Read the .edf file of the patient and open EEGlab for analysis
%
% Open EEGlab
eeglab;
FileTemp = FilenameCombined(SeizureNumber);
FileSeizure = FileTemp{1};

EEG = pop_biosig(FileSeizure,'importevent','off','blockepoch','off');

EEG.setname = 'CurrentSet';
EEG = eeg_checkset(EEG);
eeglab redraw;
```

```
%-------------------------------------------------------------------------
% Extract 20 second data for analysis
%
% The 20 second data to be analyzed is 20 seconds during the seizure.
% This value will be extracted from epileptic channels. A total of 20
% seconds for 20 channels will be extracted. The electrode label of the
% channels will be extracted as well for reference.

y = EEG.data;
y = double(y);

%Prior = 10;
%During = 10;
During = 20;

StartData = StartSeizure(SeizureNumber);
%StartData1 = StartData -2000;
% 10 seconds prior to the start of the seizure
%Start = StartData - Prior;
Start = StartData+Nextset;
% 10 seconds after the start of the seizure
%Stop = StartData + During;
Stop = StartData+Nextset + During;

% For 20 seconds of data for first set of epileptic channels

x(1:length(E1begin:E1end),:)= y(E1begin:E1end,Fs*Start:Fs*Stop-1);



% Creating channel label vectors for first set of epileptic channels

xlabels_temp = char(EEG.chanlocs.labels);xlables = xlabels_temp;

xlabels(1:length(E1begin:E1end),:) = xlabels_temp(E1begin:E1end,:);

xlabels = cellstr(xlabels);

% Normalize mean and standard deviation
x2 = zscore(x,0,2);

% % For 20 seconds of data on second set of epileptic channels
%
% z(1:length(E2begin:E2end),:)= y(E2begin:E2end,Fs*Start:Fs*Stop-1);
%
%
```

```
% % Creating channel label vectors for second set of epileptic channels
%
% zlabels_temp = char(EEG.chanlocs.labels);zlables = zlabels_temp;
%
% zlabels(1:length(E2begin:E2end),:) = zlabels_temp(E2begin:E2end,:);
%
% zlabels = cellstr(zlabels);
%
% % Normalize mean and standard deviation
% z2 = zscore(z,0,2);
%

% %--------------------------------------------------------------------------
% % Till this step the data is extracted and normalized.
% % The next step is to visualize the 20 second data.
% %Signal = x2(1,:);
% %Signal = z2(2,:);
% taxis = 0:1/Fs:20-1/Fs; % Time for the axis on graph
% figure(2);
% plot(taxis,x2(34,:));
% title('Subject#2, Seizure#1, Channel 34 Original signal - 1000Hz');
% xlabel('Time (s)');
% ylabel('Amplitude (units)');
%
% % plot(z2(1,:));
% % title('Original signal - 1000Hz');
% Signal = x2(34,:);
%
% [ A7,D1,D2,D3,D4,D5,D6,D7 ] = Wavelet_Decomposition( Signal,7);
%
% figure(3);
% subplot(4,2,1);
% plot(taxis,A7);
% xlabel('Time (s)');
% ylabel('A7');
% %title('Epileptic - Approximation A7');
% subplot(4,2,2);
% plot(taxis,D1),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D1');
% %title('Epileptic - Detailed D1');
% subplot(4,2,3);
% plot(taxis,D2),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D2');
% %title('Epileptic - Detailed D2');
% subplot(4,2,4);
% plot(taxis,D3),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D3');
% %title('Epileptic - Detailed D3');
% subplot(4,2,5);
% plot(taxis,D4),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D4');
% %title('Epileptic - Detailed D4');
% subplot(4,2,6);
% plot(taxis,D5),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D5');
% %title('Epileptic - Detailed D5');
% subplot(4,2,7);
% plot(taxis,D6),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D6');
% %title('Epileptic - Detailed D6');
% subplot(4,2,8);
% plot(taxis,D7),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D7');
% %title('Epileptic - Detailed D7');
% %---------------------------------------------------------------------------
%
% % Generating the PCA and ICA final matrix for all the signal
%
% % First set of epileptic channel signals
% Signal0 = x2(1,:); % Channel 4
% Signal1 = x2(2,:); % Channel 5
% Signal2 = x2(3,:); % Channel 6
% %
% % % Second set of epileptic channel signals
% Signal3 = z2(1,:); % Channel 33
% %Signal4 = z2(2,:);  % Channel 34
% % %Signal5 = z2(3,:); % Channel 35
% % %Signal6 =z2(4,:);  % Channel 36
% % %Signal7 = z2(5,:); % Channel 37
% %Signal8 = z2(6,:); % Channel 38
% %Signal9 = z2(9,:); % Channel 39
% %Signal10 = z2(10,:); % Channel 40
% %Signal11 = z2(11,:);  % Channel 41
% %Signal12 = z2(12,:); % Channel 42
% % %Signal13 = z2(13,:): % Channel 43
% Signal14 = z2(14,:); % Channel 44
% %
% [FS0,P0,IC0,exp0] = Combination(Signal0);
% [FS1,P1,IC1,exp1] = Combination(Signal1);
% [FS2,P2,IC2,exp2] = Combination(Signal2);
% [FS3,P3,IC3,exp3] = Combination(Signal3);
% %[FS4,P4,IC4,exp4] = Combination(Signal4);
% % %P5 = Combination(Signal5);
% % %P6 = Combination(Signal6);
% % %P7 = Combination(Signal7);
% %[FS8,P8,IC8,exp8] = Combination(Signal8);
% %[FS9,P9,IC9,exp9] = Combination(Signal9);
% %[FS10,P10,IC10,exp10] = Combination(Signal10);
% %[FS11,P11,IC11,exp11] = Combination(Signal11);
% %[FS12,P12,IC12,exp12] = Combination(Signal12);
% % %P13 = Combination(Signal13);
% [FS14,P14,IC14,exp14] = Combination(Signal14);
% %------------------------------------------------------------------------
% % % Printing the scree plot for the above channels
% % figure(2);
% % plot(exp3);
% % title('Scree plot (epileptic) - Seizure 12 - Channel 33');
% % xlabel('Principal Component');
% % ylabel('Variance explained(%)');
% % figure(3);
% % plot(exp4);
% % title('Scree plot (epileptic) - Seizure 2 - Channel 34');
% % xlabel('Principal Component');
% % ylabel('Variance explained(%)');
% % figure(4);
% % plot(exp11);
% % title('Scree plot (epileptic) - Seizure 12 - Channel 41');
% % xlabel('Principal Component');
% % ylabel('Variance explained(%)');
% % figure(5);
% % plot(exp12);
% % title('Scree plot (epileptic) - Seizure 12 - Channel 42');
% % xlabel('Principal Component');
% % ylabel('Variance explained(%)');
%
% % exp3 = exp3';
% % exp4 = exp4';
% % exp11 = exp11';
% %------------------------------------------------------------------------
% % Creating the training matrix for the epileptic data with first three rows
% % as principal components 1,2 and 3 and the next three rows as independent
% % components 1,2 and 3. Echan contains the channel numbers for the data
% i1 = 1;
% i2 = 79;
% i = 4;
% Etrain(i1:i2,2:20) = FS0;
% Etrain(i1:i2,21:23) = P0;
% Etrain(i1:i2,24:26) = IC0;
% Etrain(i1:i2,1) = i;
%
% i1 = i1+79;
% i2 = i2+79;
% i = 5;
% Etrain(i1:i2,2:20) = FS1;
% Etrain(i1:i2,21:23) = P1;
% Etrain(i1:i2,24:26) = IC1;
% Etrain(i1:i2,1) = i;
%
% i1 = i1+79;
% i2 = i2+79;
% i = 6;
% Etrain(i1:i2,2:20) = FS2;
% Etrain(i1:i2,21:23) = P2;
% Etrain(i1:i2,24:26) = IC2;
% Etrain(i1:i2,1) = i;
%
% i1 = i1+79;
% i2 = i2+79;
% i = 33;
% Etrain(i1:i2,2:20) = FS3;
% Etrain(i1:i2,21:23) = P3;
% Etrain(i1:i2,24:26) = IC3;
% Etrain(i1:i2,1) = i;
%
% i1 = i1+79;
% i2 = i2+79;
% i = 44;
% Etrain(i1:i2,2:20) = FS14;
% Etrain(i1:i2,21:23) = P14;
% Etrain(i1:i2,24:26) = IC14;
% Etrain(i1:i2,1) = i;
%
% % i1 = i1+79;
% % i2 = i2+79;
% % i = 44;
% % Etrain(i1:i2,2:20) = FS14;
% % Etrain(i1:i2,21:23) = P14;
% % Etrain(i1:i2,24:26) = IC14;
% % Etrain(i1:i2,1) = i;
% %------------------------------------------------------------------------
% Creating the training matrix for all the channels for one seizure. The
% training matrix has first column as the channel number. For feature
% matrix the columns are 15 features. and for the PCA+ICA model the columns
% are 3 pcs and 3 Ics.


i1 = 1;
for i = 1:38;
    Signal = x2(i,:);
    i2 = i1+78;
    [FS,P,IC] = Combination(Signal);
    Etrain(i1:i2,2:20) = FS;
    Etrain(i1:i2,21:23) = P;
    Etrain(i1:i2,24:26) = IC;
    Etrain(i1:i2,1) = i;
    i1=i1+79;
end
for i = 41:74;
    Signal1 = x2(i,:);
    i2 = i1+78;
    [FS,P,IC] = Combination(Signal1);
    Etrain(i1:i2,2:20) = FS;
    Etrain(i1:i2,1) = i-2;
    Etrain(i1:i2,21:23) = P;
    Etrain(i1:i2,24:26) = IC;
    i1=i1+79;
end

TestNew = Etrain;
%------------------------------------------------------------------------
% Using the Etrain matrix as TestNew matrix with the SVM function

Edata = xlsread('EpilepticTrain.xlsx');
NEdata = xlsread('NEpilepticTrain.xlsx');
% TestTrain = xlsread('TestTrainFMData.xlsx');
% TestNew = xlsread('TestFMData.xlsx');

% Creating the Training matrix for epileptic and non epileptic data
data1 = Edata(4001:24000,2:20);
data2 = NEdata(4001:19000,2:20);
data3 =[data1;data2]; % data set containing the training set data
class = ones(35000,1); %NE = 1 and E = -1
class(1:20000) = -1;% data set containing the classifier for training set
```

```
% Creating the matrix for picking optimal values for the SVM

datac1 = Edata(24001:25500,2:20);
datac2 = NEdata(24001:25500,2:20);
datac3 =[datac1;datac2]; % data set containing the training set data
classc = ones(3000,1); %NE = 1 and E = -1
classc(1:1500) = -1;% data set containing the classifier for training set

%-------------------------------------------------------------------------
% SVM training and cross validation
tic
%Picking optimal values for the rbf_sigma and boxconstraint
[d1,d2] = size(datac3);
c = cvpartition(d1,'KFold',10);
minfn = @(z)kfoldLoss(fitcsvm(datac3,classc,'CVPartition',c,...
    'KernelFunction','rbf','Boxconstraint',exp(z(2)),...
    'KernelScale',exp(z(1))));
opts = optimset('TolX',5e-4,'TolFun',5e-4);
[searchmin,fval] = fminsearch(minfn,randn(2,1),opts);
z = exp(searchmin);
%z = [1.2992;1.3311];

%-------------------------------------------------------------------------

% Using fitcsvm and values from cross validation to train the SVM model
SVMModel = fitcsvm(data3,class,'KernelFunction','rbf',...
    'KernelScale',z(1),'BoxConstraint',z(2));
CVSVMModel = crossval(SVMModel);
misclass = kfoldLoss(CVSVMModel);
misclass
toc
%-------------------------------------------------------------------------
% Using SVMModel to test the test data and generate the matrix

tic
% Classification of the TestNew data which is the test matrix created using
% the data from subject #2 and the data was not used for training the SVM
OL = 0;
WL = 79;
SignalBlock = (length(TestNew)/79);
SignalEnd = WL-1;

i1 = 1;
for i = 1:SignalBlock
    i2 = i1+SignalEnd;
    %Testing the data using SVMStruct
    Test(:,:) = TestNew(i1:i2,2:20);
    [label,score] = predict(SVMModel,Test);
    Test1(i1:i2,1:20) = TestNew(i1:i2,1:20);
    Test1(i1:i2,21) = label;

    i1 = (i2-OL)+1;
end

toc


j1 = 1;
for j = 1:72
    j2 = j1+78;
    temp = sum(Test1(j1:j2,21));
    if(temp<=0)
        x = (temp+79)/2;
        epi(j,1) = -((x/79)*100);
        %epi(j,2) = ((79-x)/79)*100;
        %temp = (temp/79)*100;
       lab = -1;
       %epi(j,1) = temp;
    else
        x = (temp+79)/2;
        lab = 1;
        %nonepi(j,1) = ((-x)/79)*100;
        nonepi(j,1) = ((79-x)/79)*100;
        %temp = (temp/79)*100;
        %nonepi(j,1) = temp;
    end

%     epi(j,1) = -(x/79)*100;
%     nonepi(j,1) = ((79-x)/79)*100;
    %results(j,1) = j;
    preresults(j,1) = temp;
    results(j,1) = lab;
    j1 = j1+79;
end

figure(4);
bar(preresults,'b');
title('Subject #2, Seizure #2 and Start time - 2300');
xlabel('Channels');
ylabel('epileptic vs. non-epileptic(units)');
xlim([0 72]);
```

```
ylim([-80 80]);
figure(5);
bar(epi,'r');
title('Subject #2, Seizure #2 and Start time - 2300');
xlabel('Channels');
ylabel('epileptic vs. non-epileptic(units)');
xlim([0 72]);
ylim([-100 100]);
hold on
bar(nonepi,'b');
hold off
```

13.Appendix D: Matlab Code: Non-epileptic

```
%########################################################################
%This program performs the following actions in a sequence:
%   - Add path to the files containing the data from the excel
%   - Get information of the start time and stop time for each seizure
%   - Read the .edf file of the patient
%   - extract 20 second data for for all the channels from the
%   non-epileptic edf files.
%
%Some initial steps for loading the EEG data for analysis has been adapted
%from James Gurisko's thesis.
%
% Priya Balasubramanian
% Created on 09/25/2014
% Last updated on
%########################################################################
clear all;
clc;
close all;

%-------------------------------------------------------------------------
% User entered parameters
%
% The information entered below will change for each patient and each
% seizure. The sampling frequency as well as the epileptic and nonepileptic
% channels being considered will change based on the patient and seizure as
% well. The length of the data remains as 20 seconds for each dataset being
% analyzed.

Filename = 'DataNotesP.xlsx'; %File containing the patient data
Directory = 'C:/eegData/SH-EEG/'; % Directory containing the patient data
%SeizureNumber = 1; % Seizure number being analyzed
NEFileNumber = 3; % File number of the nonepileptic file being analyzed
DataLength = 20; % Length of the data in seconds
Fs = 1000 ; % sampling frequency in Hz
NEbegin = 1;
NEend = 76;% Range of non-epileptic channels
%-------------------------------------------------------------------------
% Add path to files
%
addpath('C:/eegData/Priya','C:/eegData/MATLAB/WOSSPA_Mathworks_v2',...
   'C:/eegData/Priya/eeglab10.2.2.4b');
% Use addpath to add the necessary folders
```

```
%--------------------------------------------------------------------------
%Read information of the seizure from the excel file

% Read the start time for each non-epileptic file
StartNE = 5000;

%Read the file name corresponding to each nonseizure from the excel file
[Num1,NEFileEDF,Raw1] = xlsread(Filename,'G:G');
size(NEFileEDF)
clear Num1;
clear Raw1;

%Create a single string for importing data
%strcat combines the directory with the .edf filename to create a single
%string for importing patient data

NEFilenameCombined = cell(16,1);

for i = 1:16,
    NEFilenameCombined(i,1) = strcat(Directory,NEFileEDF(i+1,1));
end

%--------------------------------------------------------------------------
% Read the .edf file of the patient and open EEGlab for analysis
%
% Open EEGlab
eeglab;

% Read the nonepileptic files
NEFileTemp = NEFilenameCombined(NEFileNumber);
NEFileSeizure = NEFileTemp{1};

EEG = pop_biosig(NEFileSeizure,'importevent','off','blockepoch','off');

EEG.setname = 'CurrentSetNE';
EEG = eeg_checkset(EEG);
eeglab redraw;



%--------------------------------------------------------------------------
% Extract 20 second data for analysis
%
% The 20 second data to be analyzed is 10 seconds prior to the start of a
% seizure and 10 seconds during the seizure. This value will be extracted
% from the ten nonepileptic channels and 10 epileptic channels for
% analysis. A total of 20 seconds for 20 channels will be extracted. The
% electrode label of the channels will be extracted as well for reference.

NEy = EEG.data;
NEy = double(NEy);

During = 20;

NEStartData = StartNE;

% 20 seconds worth of data of non-epileptic
NEStart = NEStartData;
NEStop = NEStart+During;

% For 20 seconds of data on nonepileptic channels

NEx(1:length(NEbegin:NEend),:)= NEy(NEbegin:NEend,Fs*NEStart:Fs*NEStop-1);

% Creating channel label vectors for epileptic channels

% NExlabels_temp = char(EEG.chanlocs.labels);NExlables = NExlabels_temp;
%
% NExlabels(1:length(NEbegin:NEend),:) = NExlabels_temp(NEbegin:NEend,:);
%
% NExlabels = cellstr(NExlabels);

% Normalize mean and standard deviation
NEx2 = zscore(NEx,0,2);

% % Signal = NEx2(34,:);
% %
% taxis = 0:1/Fs:20-1/Fs; % Time for the axis on graph
% figure(2);
% plot(taxis,NEx2(34,:));
% title('Subject#2, Channel 34 Original signal - 1000Hz');
% xlabel('Time (s)');
% ylabel('Amplitude (units)');
% Signal = NEx2(34,:);
%
% [ A7,D1,D2,D3,D4,D5,D6,D7 ] = Wavelet_Decomposition( Signal,7);
%
% figure(3);
% subplot(4,2,1);
% plot(taxis,A7);
% xlabel('Time (s)');
% ylabel('A7');
% %title('Non-Epileptic - Approximation A7');
% subplot(4,2,2);
% plot(taxis,D1),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D1');
% %title('Non-Epileptic - Detailed D1');
% subplot(4,2,3);
% plot(taxis,D2),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D2');
% %title('Non-Epileptic - Detailed D2');
% subplot(4,2,4);
% plot(taxis,D3),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D3');
% %title('Non-Epileptic - Detailed D3');
% subplot(4,2,5);
% plot(taxis,D4),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D4');
% %title('Non-Epileptic - Detailed D4');
% subplot(4,2,6);
% plot(taxis,D5),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D5');
% %title('Non-Epileptic - Detailed D5');
% subplot(4,2,7);
% plot(taxis,D6),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D6');
% %title('Non-Epileptic - Detailed D6');
% subplot(4,2,8);
% plot(taxis,D7),ylim([-1.5 1.5]);
% xlabel('Time (s)');
% ylabel('D7');
% %title('Non-Epileptic - Detailed D7');


% [pca_final1,ica_final1,explained1] = Combination(Signal);
% explained1 = explained1';
% Signal = NEx2(34,:);
% [pca_final2,ica_final2,explained2] = Combination(Signal);
% explained2 = explained2';
% Signal = NEx2(41,:);
% [pca_final3,ica_final3,explained3] = Combination(Signal);
% explained3 = explained3';
% Signal = NEx2(42,:);
% [pca_final4,ica_final4,explained4] = Combination(Signal);
% explained4 = explained4';
%
% % figure(2);
% % plot(explained);
% % title('Scree plot (Nonepileptic) - BA26802O_1-1.edf - Channel 56');
% % xlabel('Principal Component');
% % ylabel('Variance explained(%)');
%
% %------------------------------------------------------------------------
% % % Till this step the data is extracted and normalized.
%
%------------------------------------------------------------------------
% Creating the training matrix for all the channels for one seizure. The
% training matrix has first column as the channel number. For feature
% matrix the columns are 15 features. and for the PCA+ICA model the columns
% are 3 pcs and 3 Ics.


i1 = 1;
for i = 1:38;
    Signal = NEx2(i,:);
    i2 = i1+78;
    [FS,P,IC] = Combination(Signal);
    NEtrain(i1:i2,2:20) = FS;
    NEtrain(i1:i2,21:23) = P;
    NEtrain(i1:i2,24:26) = IC;
    NEtrain(i1:i2,1) = i;
    i1=i1+79;
end
    %
%i1 = 1;
for i = 41:74;
    Signal = NEx2(i,:);
    i2 = i1+78;
    [FS,P,IC] = Combination(Signal);
    NEtrain(i1:i2,2:20) = FS;
    NEtrain(i1:i2,21:23) = P;
    NEtrain(i1:i2,24:26) = IC;
    NEtrain(i1:i2,1) = i-2;
    i1=i1+79;
end

TestNew = NEtrain;
%------------------------------------------------------------------------
% Using the NEtrain matrix as TestNew matrix with the SVM function
```

```
Edata = xlsread('EpilepticTrain.xlsx');
NEdata = xlsread('NEpilepticTrain.xlsx');
% TestTrain = xlsread('TestTrainFMData.xlsx');
% TestNew = xlsread('TestFMData.xlsx');

% Creating the Training matrix for epileptic and non epileptic data
data1 = Edata(4001:24000,2:20);
data2 = NEdata(4001:19000,2:20);
data3 =[data1;data2]; % data set containing the training set data
class = ones(35000,1); %NE = 1 and E = -1
class(1:20000) = -1;% data set containing the classifier for training set

% Creating the matrix for picking optimal values for the SVM

datac1 = Edata(24001:25500,2:20);
datac2 = NEdata(24001:25500,2:20);
datac3 =[datac1;datac2]; % data set containing the training set data
classc = ones(3000,1); %NE = 1 and E = -1
classc(1:1500) = -1;% data set containing the classifier for training set

%-------------------------------------------------------------------------
% SVM training and cross validation
tic
%Picking optimal values for the rbf_sigma and boxconstraint
[d1,d2] = size(datac3);
c = cvpartition(d1,'KFold',10);
minfn = @(z)kfoldLoss(fitcsvm(datac3,classc,'CVPartition',c,...
    'KernelFunction','rbf','Boxconstraint',exp(z(2)),...
    'KernelScale',exp(z(1))));
opts = optimset('TolX',5e-4,'TolFun',5e-4);
[searchmin,fval] = fminsearch(minfn,randn(2,1),opts);
z = exp(searchmin);
%z = [1.2992;1.3311];

%-------------------------------------------------------------------------

% Using fitcsvm and values from cross validation to train the SVM model
SVMModel = fitcsvm(data3,class,'KernelFunction','rbf',...
    'KernelScale',z(1),'BoxConstraint',z(2));
CVSVMModel = crossval(SVMModel);
misclass = kfoldLoss(CVSVMModel);
misclass
toc
%-------------------------------------------------------------------------
% Using SVMModel to test the test data and generate the matrix
```

```
tic
% Classification of the TestNew data which is the test matrix created using
% the data from subject #2 and the data was not used for training the SVM
OL = 0;
WL = 79;
SignalBlock = (length(TestNew)/79);
SignalEnd = WL-1;

i1 = 1;
for i = 1:SignalBlock
    i2 = i1+SignalEnd;
    %Testing the data using SVMStruct
    Test(:,:) = TestNew(i1:i2,2:20);
    [label,score] = predict(SVMModel,Test);
    Test1(i1:i2,1:20) = TestNew(i1:i2,1:20);
    Test1(i1:i2,21) = label;

    i1 = (i2-OL)+1;
end

toc



j1 = 1;
for j = 1:72
    j2 = j1+78;
    temp = sum(Test1(j1:j2,21));
    if(temp<=0)
        x = (temp+79)/2;
        epi(j,1) = -((x/79)*100);
        %epi(j,2) = ((79-x)/79)*100;
        %temp = (temp/79)*100;
       lab = -1;
       %epi(j,1) = temp;
    else
        x = (temp+79)/2;
        lab = 1;
        %nonepi(j,1) = ((-x)/79)*100;
        nonepi(j,1) = ((79-x)/79)*100;
        %temp = (temp/79)*100;
        %nonepi(j,1) = temp;
    end

%     epi(j,1) = -(x/79)*100;
%     nonepi(j,1) = ((79-x)/79)*100;
```

```
    %results(j,1) = j;
    preresults(j,1) = temp;
    results(j,1) = lab;
    j1 = j1+79;
end

figure(4);
bar(preresults,'b');
title('Subject #2, Seizure #1 and Start time - 6750');
xlabel('Channels');
ylabel('epileptic vs. non-epileptic(units)');
xlim([0 72]);
ylim([-80 80]);
figure(5);
bar(epi,'r');
title('Subject #2, Seizure #1 and Start time - 6750');
xlabel('Channels');
ylabel('epileptic vs. non-epileptic(units)');
xlim([0 72]);
ylim([-100 100]);
hold on
bar(nonepi,'b');
hold off
```

14.Appendix E: Matlab Code: Functions

```
function [ FeatureSignal,PCAFinal,ICAFinal,explained ] = Combination( Signal )
%UNTITLED7 Summary of this function goes here
%   Detailed explanation goes here
%-----------------------------------------------------------------------
% Decimate the signal for ease of computation and to get the right
% frequency range using the Decimating_Frequency function
% Function inputs are df = decimating factor,os = original signal. Here
% the df = 5

% Decimating channels used by NE
% y2 = Decimating_Frequency(x2,5);
% [m,n] = size(y2);
%figure(3);
% plot(y2(1,:));
% title('Decimated signal - 200Hz');
%Signal = y2(1,:);

% Decimating channels used by E
% Y2 = Decimating_Frequency(z2,5);
% [m1,n1] = size(Y2);
%figure(3);
%plot(Y2(3,:));
% title('Decimated signal - 200Hz');
%Signal = Y2(3,:);

%-----------------------------------------------------------------------
% Performing the wavelet decomposition
[A7,D1,D2,D3,D4,D5,D6,D7] = Wavelet_Decomposition(Signal,7);

% figure(2);
% subplot(4,2,1);
% plot(A7);
% title('Epileptic - Approximation A7');
% subplot(4,2,2);
% plot(D1),axis([ 0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D1');
% subplot(4,2,3);
% plot(D2),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D2');
% subplot(4,2,4);
% plot(D3),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D3');
% subplot(4,2,5);
% plot(D4),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D4');
% subplot(4,2,6);
% plot(D5),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D5');
% subplot(4,2,7);
% plot(D6),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D6');
% subplot(4,2,8);
% plot(D7),axis([0 20000 -1.5 1.5]);
% title('Epileptic - Detailed D7');

%-------------------------------------------------------------------------
% Extracting the statistical fetures from D5, D6, D7 and A7 subbands

% There will be a total of 15 features dimensions;
% Meanx4 subbands+averagex4 subbands+ SDx4 subbands+ ratiox4 subbands =
% 4+4+4+3 = 15 features dimensions
% Extracting the statistical features

overlap = 250; % 0.25 second overlap
WindowLength = 500; % 0.5 second window length
SignalBlock = (length(D5)/overlap)-1;



%Statistical features for wavelet subband D5
[MeanD4,AvgD4,SDD4] = Stat_Features(D4,overlap,WindowLength);
%Statistical features for wavelet subband D5
[MeanD5,AvgD5,SDD5] = Stat_Features(D5,overlap,WindowLength);
%Statistical features for wavelet subband D6
[MeanD6,AvgD6,SDD6] = Stat_Features(D6,overlap,WindowLength);
%Statistical features for wavelet subband D7
[MeanD7,AvgD7,SDD7] = Stat_Features(D7,overlap,WindowLength);
%Statistical features for wavelet subband A7
[MeanA7,AvgA7,SDA7] = Stat_Features(A7,overlap,WindowLength);

% Calculating the ratios R1,R2 and R3
RatioR1 = rdivide(MeanD4,MeanD5);
RatioR2 = rdivide(MeanD5,MeanD6);
RatioR3 = rdivide(MeanD6,MeanD7);
RatioR4 = rdivide(MeanD7,MeanA7);

% Creating a 79x15 feature matrix
% Rows of the matrix corresponds to observations or timepoints
% Columns of the matrix corresponds to variables or features
```

```
FeatureSignal = zeros(SignalBlock,19); % Preallocating for the feature matrix

% Extracts the feature lable from the dataexcel
% Labels =MeanD5,MeanD6,MeanD7,MeanA7,AvgD7,AvgD6,AvgD7,AvgA7,SDD5,SDD6,
% SDD7,SDA7,Ratio1,Ratio2,Ratio3

%[num1,FeatureLabel,raw1] = xlsread(Filename,3,'B:B');
%clear num1;
%clear raw1;

FeatureSignal(:,1) = MeanD4;
FeatureSignal(:,2) = MeanD5;
FeatureSignal(:,3) = MeanD6;
FeatureSignal(:,4) = MeanD7;
FeatureSignal(:,5) = MeanA7;
FeatureSignal(:,6) = AvgD4;
FeatureSignal(:,7) = AvgD5;
FeatureSignal(:,8) = AvgD6;
FeatureSignal(:,9) = AvgD7;
FeatureSignal(:,10) = AvgA7;
FeatureSignal(:,11) = SDD4;
FeatureSignal(:,12) = SDD5;
FeatureSignal(:,13) = SDD6;
FeatureSignal(:,14) = SDD7;
FeatureSignal(:,15) = SDA7;
FeatureSignal(:,16) = RatioR1;
FeatureSignal(:,17) = RatioR2;
FeatureSignal(:,18) = RatioR3;
FeatureSignal(:,19) = RatioR4;

%------------------------------------------------------------------------
% Performing PCA on the FeatureSignal

[PCAFinal, explained] = Pca_Analysis(FeatureSignal);

% Performing ICA on the FeatureSignal and presenting the ICs

NumberIC = 3; % number of Independant components we are interested in
FeatureICA = FeatureSignal';
% Cumputing the ICA using the jadeR function
W = jadeR(FeatureICA,NumberIC);

ICAFinal = (W*FeatureICA)';



end
```

```
function [ Mean,Avg,SD ] = Stat_Features( SB,OL,WL )
%Extracts the statistical features of a subband
%   1. Mean of the absolute values of the coefficients in each sub band
% eg. If there are 20000 samples so 20s worth of data, the windown length
% for the statistical analysis will be 0.5s each with an overlap of 0.25
% seconds and that will give a total of 79 time points or sample points
% for each subband.
% 2. Average power of the wavelet coefficients in each sub band
% 3. Standard deviation of the coefficients in each sub band

SignalBlock = (length(SB)/OL)-1;
SignalEnd = WL - 1;

% Preallocating matrix for mean, avg and std

Mean = zeros(1,SignalBlock);
Avg = zeros(1,SignalBlock);
SD = zeros(1,SignalBlock);

i1 = 1;
for i = 1:SignalBlock
    i2 = i1+SignalEnd;
    %Statistical features for the wavelet subband
    Mean(i) = mean(abs(SB(i1:i2)));
    Avg(i) = mean((SB(i1:i2).^2));
    SD(i) = std((SB(i1:i2).^2));

    i1 = (i2-OL)+1;
end


end
```

```
function [ A7,D1,D2,D3,D4,D5,D6,D7 ] = Wavelet_Decomposition(S,L)
%UNTITLED3 Wavelet decomposition of the original signal
% C is the vector formed by concatenating approximation and detail
% coeeficients at each level. L is the vector that gives the length of each
% component. Courtsey:
% http://www.mathworks.com/help/wavelet/ug/one-dimensional-
% discrete-wavelet-analysis.html#f4-997029

% S = signal, L = level of the decomposition.
```

```
[C,L] = wavedec(S,L,'db6');

cA7 = appcoef(C,L,'db6',7); % Extracts approximation coefficents

% Extracts detailed coefficients

[cD1,cD2,cD3,cD4,cD5,cD6,cD7] = detcoef(C,L,[1,2,3,4,5,6,7]);



% Reconstructs the signal component corresponsing to each of the six
% wavelet coefficient sequences

A7 = wrcoef('a',C,L,'db6',7);
D1 = wrcoef('d',C,L,'db6',1);
D2 = wrcoef('d',C,L,'db6',2);
D3 = wrcoef('d',C,L,'db6',3);
D4 = wrcoef('d',C,L,'db6',4);
D5 = wrcoef('d',C,L,'db6',5);
D6 = wrcoef('d',C,L,'db6',6);
D7 = wrcoef('d',C,L,'db6',7);
end
```

```
function [ pca_final, explained] = Pca_Analysis( Feature_Signal )
%Pca_Analysis perform the PCA analysis on the Feature matrix
%   This function uses the buitl in matlab function to determine the
%   covariance matrix that has eigenvectors as its rows and the PCAs as its
%   column. The t-squared value generated by the function is used to
%   identify the 8 most extreme points and based on the scree plot the
%   first 4 PCAs are used to generate the output matrix called pcafinal
%   which is an 8x4 matrix.
% Input arguments include the Feature signal which is a matrix of rows with
% observations or time points and columns with 15 features.

% % Boxplot to look at the distribution of the feature data
% figure(5);
% boxplot(Feature_Signal,'orientation','horizontal','labels',FeatureLabel);
% title('Distributions of features');

% Checking the correlation between the features/variables

Correlation = corr(Feature_Signal,Feature_Signal);

% Performing PCA by using the inverse variances of the data as weights
```

```
[wcoeff,score,latent,tsquared,explained] = pca(Feature_Signal,...
    'VariableWeights','variance');

c3 = wcoeff(:,1:3);% The first three principal component coefficients

% Transforming the coefficients so that they are orthonormal
coefforth = (diag(std(Feature_Signal)))\wcoeff;

% "Score" contains the coordinates of the original data in the new
% coordinate system defined by the principal components. The score matrix
% is the same size as the input data matrix.

csscores = zscore(Feature_Signal)*coefforth;

% figure(6);
% %subplot(2,1,1);
% plot(score(:,1),score(:,2),'+');
% xlabel('1st Principal Component');
% ylabel('2nd Principal Component');

% figure(7);
% plot(explained);
% title('Scree plot');
% xlabel('Principal Component');
% ylabel('Variance explained(%)');

% Using the Hotellings T-squared statistic to analytically find the most
% extreme points in the data

[st2, index] = sort(tsquared, 'descend');
extreme = index(1:8,:);
pca_final(:,:) = score(:,1:3);

% Visualizing both the orthonormal principal component coefficients for
% each variable and the principal component scores for each observation
% in a single plot.
% All 15 Features are represented in this bi-plot by a vector, and the
% direction and length of the vector indicate how each variable
% contributes to the two principal components in the plot.

% figure(8);
% biplot(coefforth(:,1:2),'scores',score(:,1:2),'varlabels',FeatureLabel);
% axis([-.26 0.6 -.51 .51]);
end
```