8-2016

Identification and Use of PSD-Derived Features for the Contextual Detection and Classification of EEG Epileptiform Transients

Sharan Rajendran
Clemson University, sharanr1993@gmail.com

Identification and Use of PSD-derived Features for the Contextual Detection and Classification of EEG Epileptiform Transients

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Electrical Engineering

---

by
Sharan Rajendran
August 2016

Accepted by:
Dr. Robert J. Schalkoff, Committee Chair
Dr. Adam Hoover
Dr. Carl W. Baum

Abstract

Epilepsy is a chronic disorder, which is characterized by seizures. For diagnosis, trained neurologists go over the patient's EEG (Electroencephalograph) records looking for epileptic transients. This is a tedious and long process.

The objective of this thesis is to automate the procedure by developing a detector that would pick out epileptic transients containing the "Abnormal Epileptiform Paroxysmal" (AEP) type. The process was split into detection of potential AEPs and the classification of the detected segments. The detection of potential AEPs (called Yellow Boxing) passed boxed segments of the EEG signal to be classified as to segments that contain paroxysmal activity or not.

For yellow boxing potential AEPs, a neural network was trained to determine if the signal contained in a sliding window was to be yellow boxed or not. If yellow boxed, the yellow box was then classified using a neural network trained to handle the classification problem. The networks were trained based on yellow boxes (potential AEPs) marked by trained neurologists.

The resulting performance of the networks was studied using sensitivity, specificity and precision as parameters. The overall performance of the detector was verified with respect to expert marked AEPs. An additional parameter, based on the detected AEP length, was also introduced for detection to overcome the drawbacks found in using specificity.

Acknowledgments

I would like to convey my thanks to a number of people who supported me. I would like to extend my sincere gratitude to my thesis advisor, Dr. Robert J. Schalkoff, for his continuous support and guidance of my research. I would thank Dr. Jonathan Halford for providing us with data necessary for this project. I would also like to thank Dr. Adam Hoover and Dr. Carl W. Baum for being part of the commitee. Finally, I would like to thank my research team members for their constant encouragement and continuous support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Description

Epilepsy or "seizure disorders", stated by the Epilepsy Foundation, is the fourth most common neurological disorder in the world . When the nerve cells in the brain start firing impulses that are abnormal, an electrical surge in the brain called a seizure, occurs. A pattern of repeated and unpredictable seizure activity is called as epilepsy [1]. Although seizure activity affects different parts of the human body, the abnormal electrical activity in the human brain is considered as one of the main symptoms of epileptic seizure.

Neurons in the human nervous system communicate by using electrical impulses that are transferred from one neuron to another. Since the 1930s, electroencephalography (EEG) has been used in order to record this electrical activity for study of brain function and diagnosis of brain disorders. This procedure records the electrical impulses from electrodes placed on the scalp. Routine electroencephalography is the most common clinical procedure used for diagnosis of epilepsy.

Trained physicians go over these recordings looking for epileptiform transients (ET) that point to the onset of epilepsy in patients. Epileptiform transients (ETs) are brief bursts of activity or transients usually lasting less than one second which occur intermittently throughout the day and night in patients with epilepsy [2]. These bursts of activities last for 20 to 70 milliseconds. ETs generally appear in the form of spikes (20-70 ms duration) or sharp waves (70-200ms duration) [3]. The process of detecting these epileptic transients is a time consuming process and although physicians are trained extensively, there is a 23% of chance of misdiagnosis [4].

Research of automated detection of ETs have been conducted taking into consideration various characteristics of the brain signals. The main challenge faced has been the creation of a training data set consisting of ETs and non-ETs. This difficulty arises due to the absence of a proper definition of what constitutes an ET. Also, there tends to be a lot of similarity between ETs and other spikes in the EEG that may occur due to normal involuntary behavior such as eye blinks and jaw movements.

1.2 Previous Work

Past research work into the analysis of epileptic transients in EEGs have revolved around differentiating between epileptic transients and normal EEG activity. Most of the research has been conducted on a common EEG database. This database contains EEG segments of fixed length of 23.6s that is acquired from different patients. The data set is divided into five classes. These five classes, each contain EEG recordings that are classified as healthy and relaxed with their eyes open, healthy and relaxed with their eyes closed, hippocampal recordings, seizure free recordings and recordings taken during seizure activity [5]. Research revolving around classificiation of these signals containing epileptic activity have shown promising results.

The investigation of power spectral density (PSD) as a classification feature has been studied while combining them with other features like wavelet transforms. PSDs have been predominantly used as an important part of the frequency domain analysis of EEG signals along with correlation. An investigation of the feature set containing spectral band powers along with other features have been conducted [6] and the ability of classifiers to learn from these datasets have been recorded. The classifiers used are Support Vector Machines (SVMs), Fischer Discriminant Analysis, Binary Decision trees, Naive Bayes, Nearest Mean and Quadratic classifiers. The performance of these classifiers using frequency domain related features and without those features have been studied in the same.

PSDs have also been used as part of a Brain Computer Interface (BCI) [7]. Research was conducted regarding the classification of EEG signals on the basis of motor actions. It made use of signal PSD that was computed as a frequency response of an autoregressive model of the signal. Another method based on a time varying autoregressive model was presented [8], where the paramters of the model were estimated by means of a Kalman filter.

Other PSD related analysis of EEGs include a study of the variance of the signal PSD as a classification feature [9], the average energy of the power spectrum [10], the spectral entropy [11] and the relative power of all five sub bands [12]. The spectral components of EEG signals have also been studied as part of an EEG-based emotion recognition system [13].

An Adaptive Neuro-Fuzzy Inference system [14] has also been used for classifying EEG signals. This made use of the neural nets adaptive abilities and a fuzzy confidence system to classify the EEG signals into five predefined classes, with each set containing EEG signals with common characteristics, such as area of recording and if or not it is an epileptic spike.

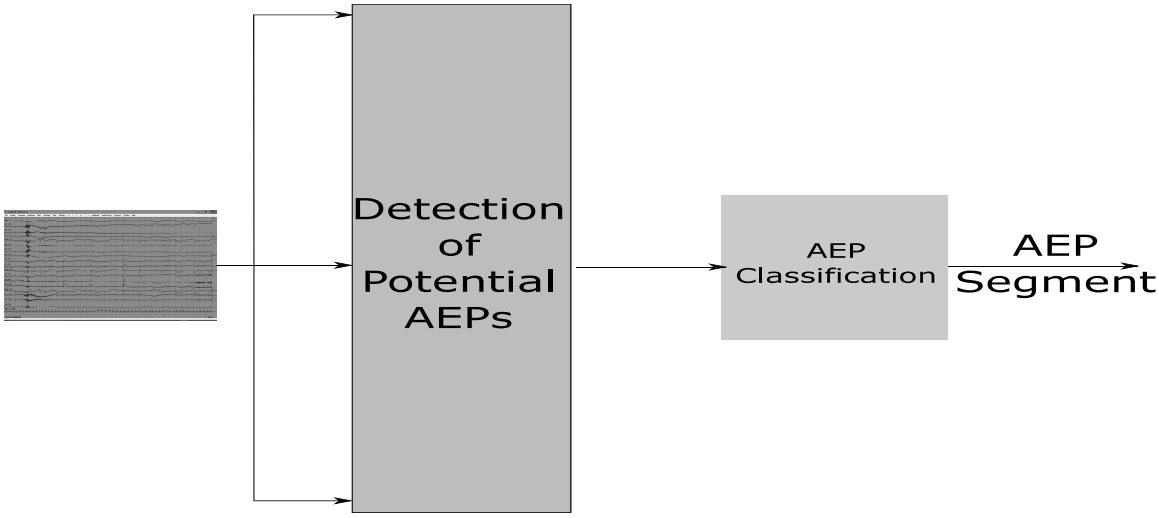


Figure 1.1: Block Diagram Representation of AEP Detection.

1.3 Approach

The main objective of this research is to automate the annotation of Epileptic transients that contain "abnormal epileptiform" paroxysmal type (AEP). The abnormal epileptiform needs to be differentiated from normal EEG activity and EEG artifacts. The approach is detailed in the following paragraphs.

In order to detect AEPs in EEG, the process is split into two parts. The first portion involves annotation of potential AEPs. This method is called "Yellow Boxing" as physicians usually mark suspect AEPs in yellow colored boxes. The second part involves classification of the obtained

Yellow-Boxed segments in to AEPs or non-AEPs. Features are extracted and then used to determine the classification output using pre-trained neural networks.

Detection and classification are done individually, so that detection can concentrate on yellow boxing suspect AEP's of variable lengths, that can be later classified as AEPs or non-AEPs. Thus, taking in to account the varying range of AEPs, it is better to handle classification as a seperate problem.

Features extracted and used are entirely based on the power spectral density of the signals, so as to exploit the sudden power surges in brain's electrical activity during these seizures.

Chapter 2

Background Information

2.1 Electronencephalography and Epileptiform Transients

The electroencephalogram (EEG) is defined as the electrical activity of an alternating type recorded from the scalp surface after being picked up by metal electrodes and conductive media [15]. These electrodes pick up electrical signals that occur in the brain and are recorded for later examination. Among these recorded signals, physicians look for sudden bursts of activities or abnormal waves that point to epilepsy in a patient. As EEG recording during seizures are labor intensive, physicians concentrate on detecting interictal seizure activity, that helps to reach a diagnosis.

Interictal epileptiform transients are usually represented by a sharp spike which is followed by a slow wave. But not all epileptiforms exhibit this kind of morphological feature. Epileptiform transients also include sharp waves, spikes and multiple spike and slow waves. Some artifacts also tend to look like epileptiform transients when they are not and require a trained physician to distinguish between them.

The 10-20 system is an internationally recognized system that defines the placement of electrodes for EEG recording [16]. This system was standardized by the International Federation in Electronencephalography and Clinical Neurophysiology in 1958. The system makes use of the relation between the location of the electrode and the cerebral cortex area that lies under the electrode. The distance between the electrodes is either 10 or 20 percent of the front-to-back or left-to-right distance of the skull. Each electrode has an alphabet assigned to it that indicates the brain lobe and numbers are used to indicate whether the electrode is attached to the left or right

hemisphere[16]. The electrical signal from each electrode is called a channel . A montage is generated by deriving a signal using two channels.

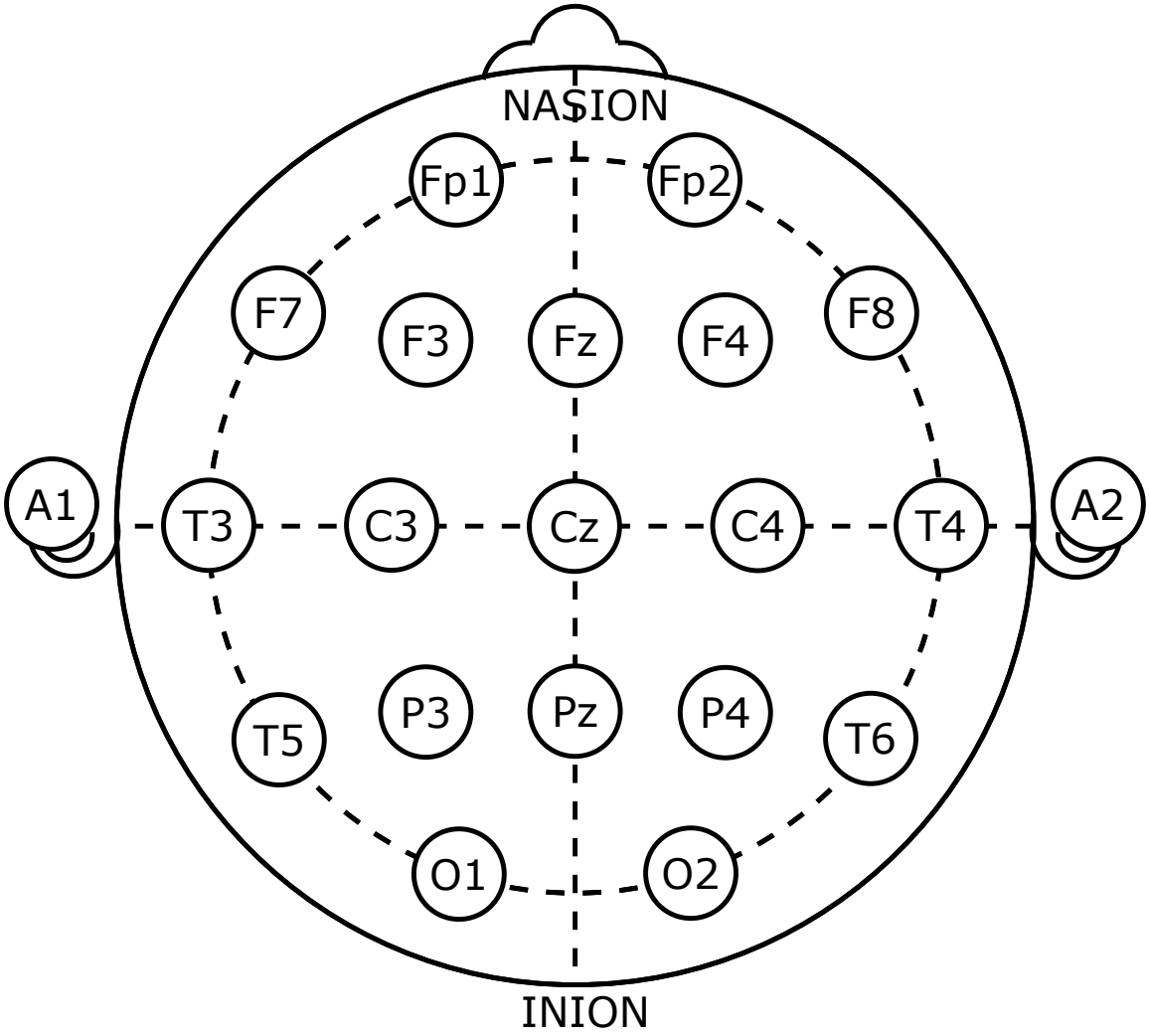


Figure 2.1: Diagrammatic Representation of 10-20 Electrode System [2].

Electrodes	Lobe
F	Frontal
T	Temporal
C	Central (Identification Purpose)
P	Parietal
O	Occipital

Table 2.1: Electrode Placements in 10-20 System. [16]

Figure 2.2: EEG Montage (as viewed in EDF Browser).

2.2 Data Information

The data used has been obtained using the 10-20 system described in the previous section. Brain activity of 200 patients has been recorded and stored using the European Data Format (EDF). Each recording is 30 seconds long and makes use of three sampling rates - 200 Hz, 256 Hz, 512 Hz. In addition to the EDF files, a Comma Separated Values (CSV) file containing the details regarding the AEP signals was provided. There are seven such files from seven consistent annotators. These files contain the start time and end time of the AEP signals, the montage and channel numbers.

Each segment of the signal enclosed within the start time and end time is said to be "Yellow Boxed". The yellow boxed portions are potential AEP signals. Each annotator assigned a confidence level to the yellow boxed signal. These confidence levels indicates the probability of the yellow boxed segment being an AEP. There are five such confidence levels. These confidence levels and the representative classes are show in Table 2.2.

Confidence Level	Definition	Assigned Class
201	Definitely not an AEP	Non-AEP
202	Not an AEP	
203	Not Sure/Do not know	Unknown Class
204	an AEP	AEP
205	Definitely an AEP	

Table 2.2: Five Confidence Levels Used by Annotators.

The duration of the supected AEP signals vary from 4-123 milliseconds. As each EEG recording had a different sampling rate, it was normalized to a common frequency of 256 Hz. There are 235 such yellow boxed annotations in total.

In order to facilitate access to the data of each annotation, the raw data (CSV and EDF files) were processed into a Matlab structure. This structure contains the following members:

- Event ID.
- Annotation start time.
- Annotation end time.
- Total time period of the annotation.
- Channel number.
- Montage ID.
- Confidence level assigned by each of the annotators.
- Montage signal.
- Annotated or "Yellow-Boxed" signal.
- Sampling rate.
- EDF Filename.

This structure allows an ease of access and expansion of the database in case of new data. It also allows the possibility of adding additional memebers for the different characteristic of the signal and the features that have been extracted from training.

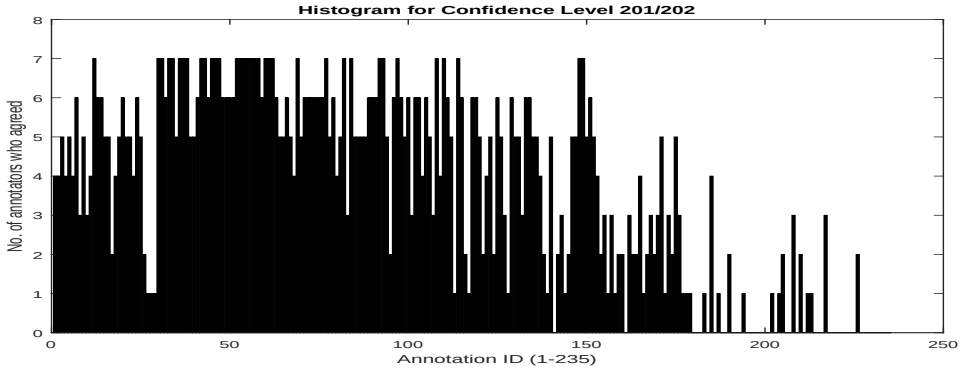


Figure 2.3: Histogram of Class 201/202.

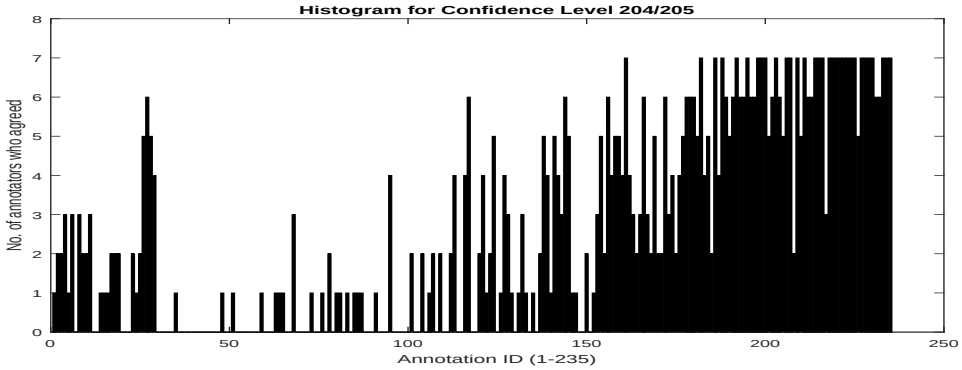


Figure 2.4: Histogram of Class 204/205.)

The histograms in Fig. 2.3 and Fig. 2.4 show the number of annotators who agreed that the annotated Yellow-Boxed signals were an AEP (204/205) or Non-AEP(201/202) respectively. These histograms to a certain extent show the consistency among annotators in assigning a confidence value to the yellow-boxed segment and in some cases a discrepancy in the confidence levels chosen by the experts. For example, there are yellow-boxed signals with Annotation IDS from 100 to 150 where there are cases of annotators having disagreements in choosing a confidence level representative of the two classes. The histograms in Fig. 2.3 and Fig. 2.4 are not perfect complements of each other due to the confidence level 203 chosen by a few annotators for some yellow boxed segments.

The problem with defining a clear cut solution to AEP classification is the wide variation in the shape of the waveforms representative of each confidence level. There is so much variation within and between the confidence levels that it is hard to define what an AEP looks like. These differences can be noted only by skilled neurologists in EEG record that runs for hours. This has

been visualized by the a few examples to represent each confidence level shown in Fig. 2.5 - Fig. 2.9.

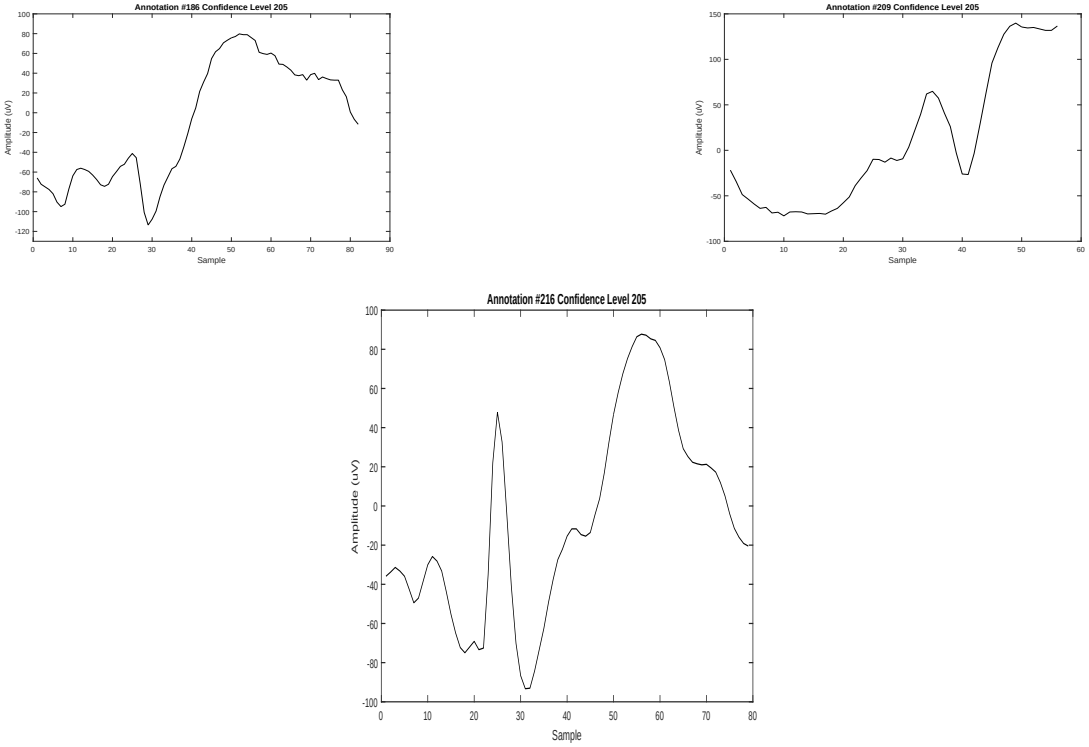


Figure 2.5: Representation of Annotated AEP's of Confidence Level 205.

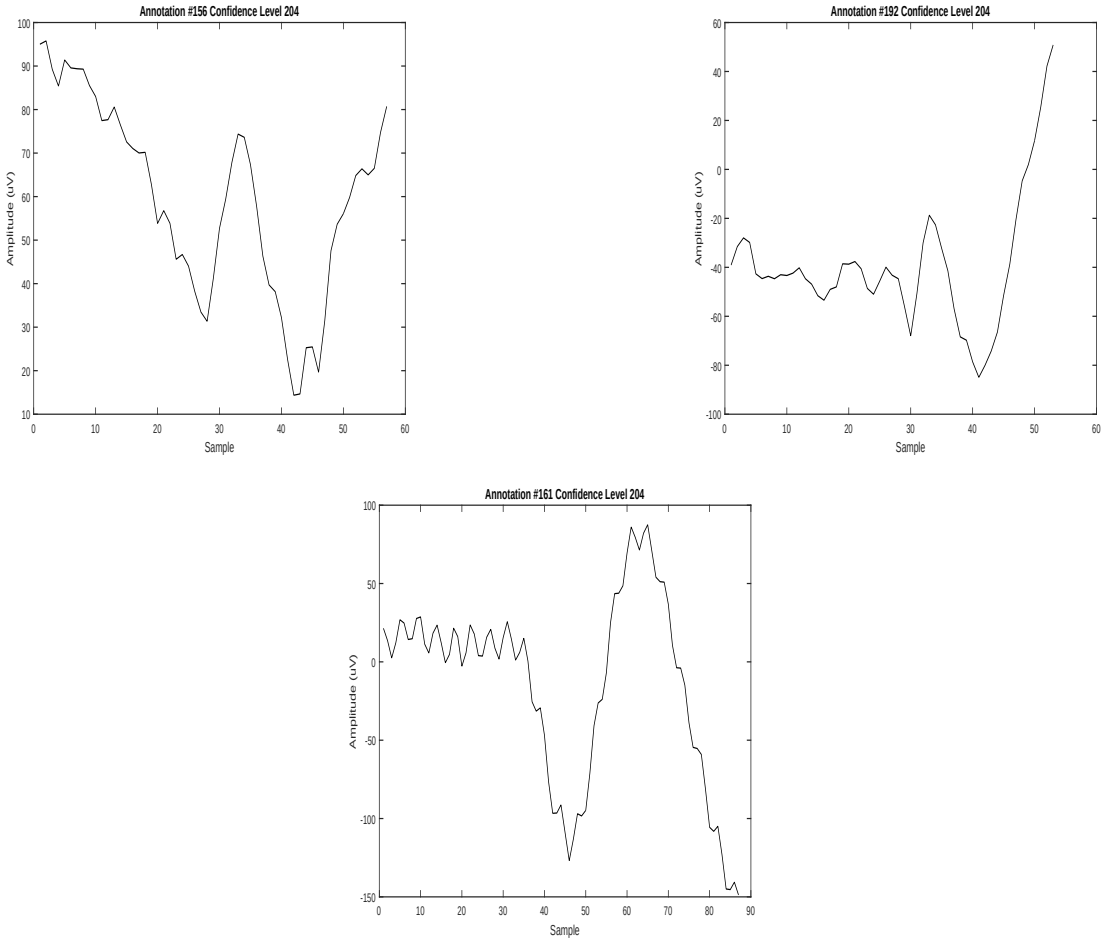


Figure 2.6: Representation of Annotated AEP's of Confidence Level 204.

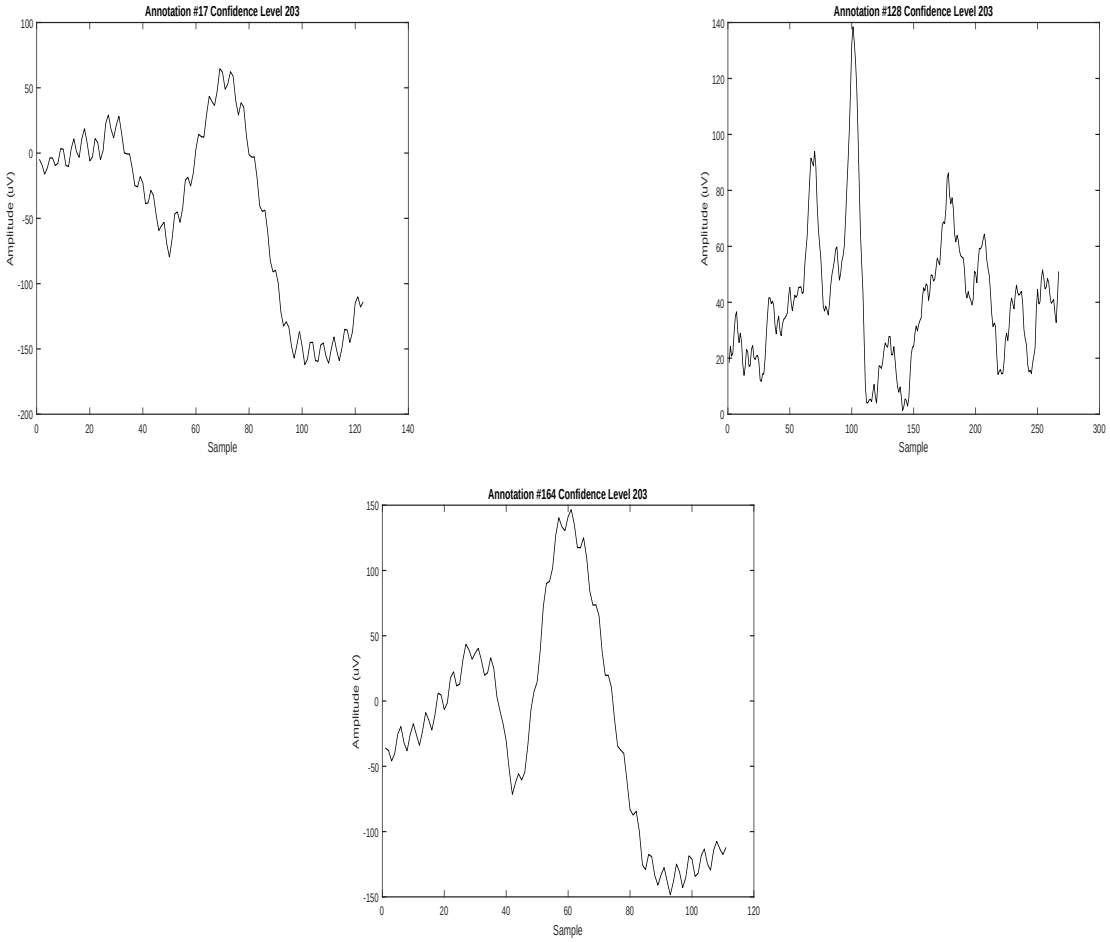


Figure 2.7: Representation of Annotated AEP's of Confidence Level 203.

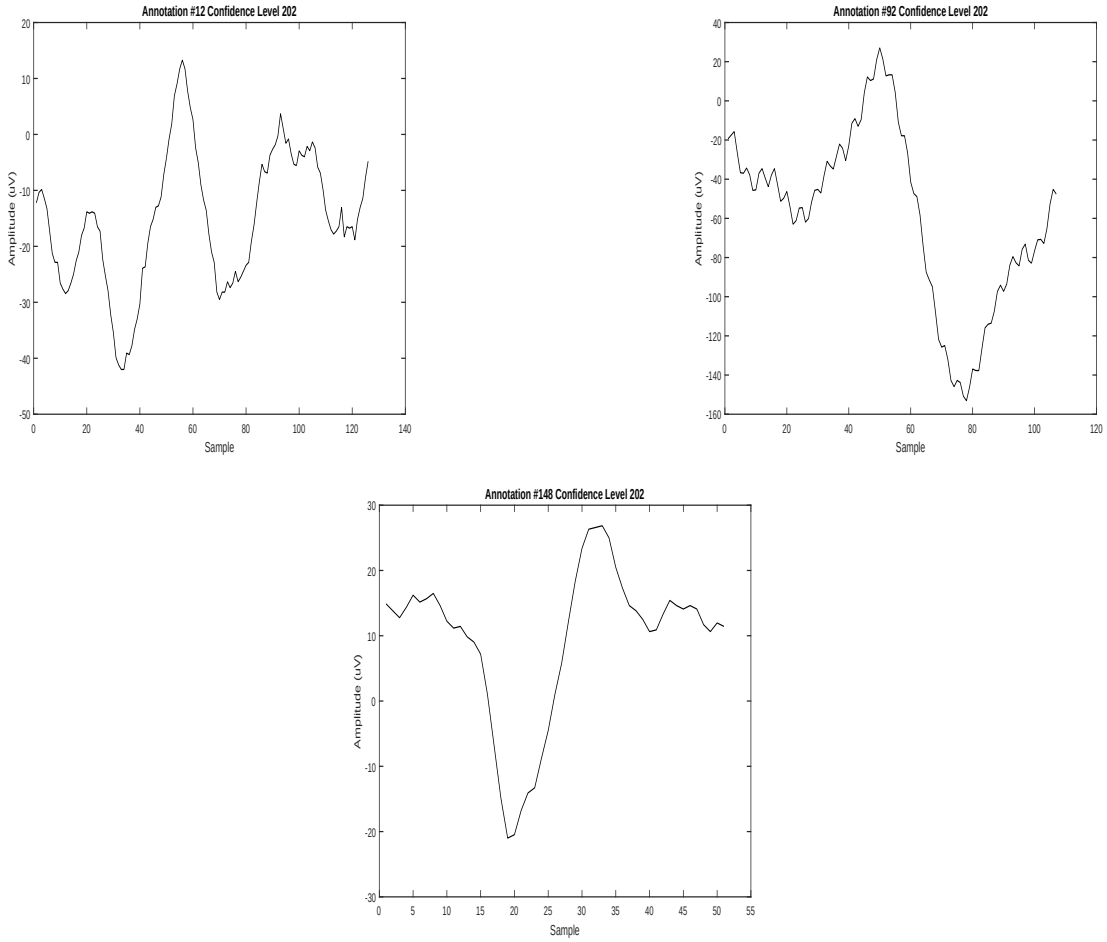


Figure 2.8: Representation of Annotated AEP's of Confidence Level 202.

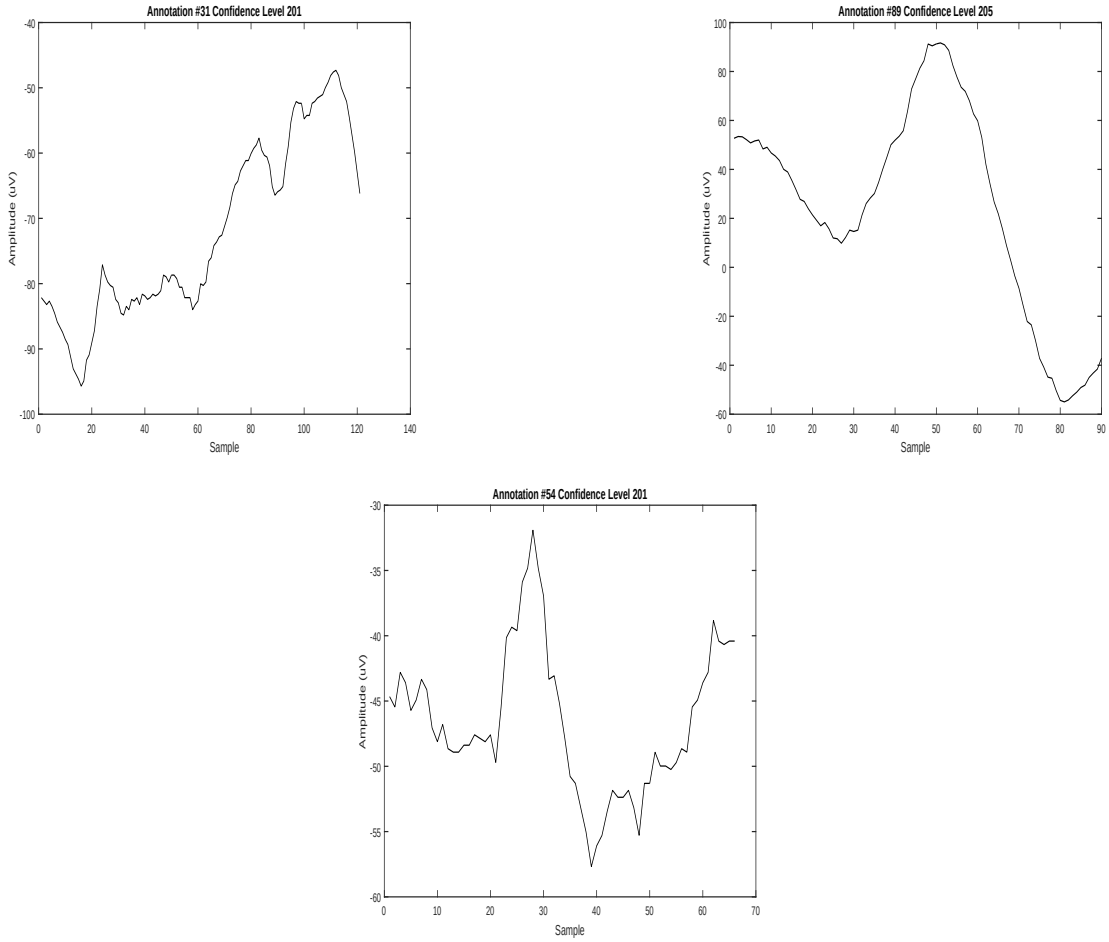
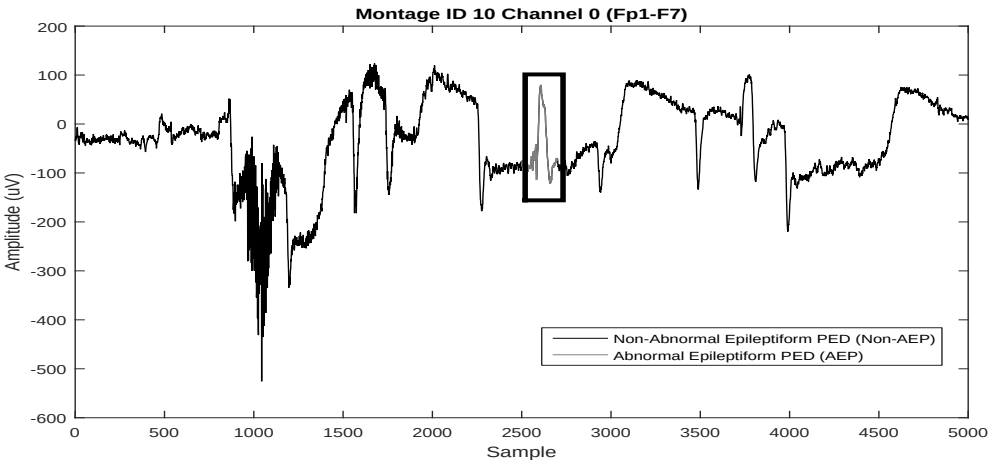


Figure 2.9: Representation of Annotated AEP's of Confidence Level 201.

2.2.1 Data Set for Yellow-Boxing

The data set for yellow boxing was created with the basis that signals could be classified into yellow-boxed or not to be yellow-boxed. For this purpose, all 235 annotations identified by the annotators were classified as yellow-boxed (YB). A chunk of the EEG recording which was not yellow-boxed was considered by default to be examples of signals that are not to be yellow-boxed. This decision is best represented by Fig. 2.10. Any artifacts that were similar to ETs but were dismissed by the annotators were also included. An automated process was used to divide this signal into segments of equal length to generate a pool of not yellow-boxed signals of a particular

time length. In order to ensure differentiability between not yellow-boxed and yellow-boxed segments in the same montage signal, the data were extracted only from the montage signals which contained yellow boxed segments.



Grey - Yellow Boxed Suspect AEP; Black - Not Yellow-boxed Data.

Figure 2.10: Yellow Boxed Segment.

2.2.2 Data for ET Classification

The data set for determination of yellow-boxed signals consists of the 235 annotated signals. These signals have been assigned a confidence level by seven different annotators. Hence, to simplify the problem, the annotations were divided into two groups: AEPs and non-AEPs. The Table 2.2 shows which confidence level is representative of which class.

In order to decide which class the annotation belonged to, the class with the highest representative count of confidence values was determined and that class was chosen. AEPs included signals with the highest class count of 204's and 205's, while non-AEPs include the signals with a high count of 201 and 202 confidence levels. Signals where the count of 204 and 205 were the same as 201 and 202 were considered inconclusive and were not included in the final data set. Examples of this classification method are shown in Table 2.3. This resulted in a final count of 228 signals, of which 89 were classified as AEPs and the rest as non-AEPs. All these signals have a different time period.

Annotation ID	Annotator ID							AEP	Non-AEP	Assigned
	1	2	3	4	5	6	7			
8	204	203	202	202	201	204	204	3	3	Inconclusive
15	202	203	201	202	201	201	205	1	5	Non-AEP
73	202	204	202	202	202	202	201	1	6	Non-AEP
142	202	203	204	204	205	205	202	5	2	AEP
233	205	204	205	205	205	205	205	7	0	AEP

Table 2.3: Examples of Classification Based on Confidence Levels

2.3 Power Spectral Density

The power spectral density function [17] of a signal shows the distribution of the signal's power over the different frequencies of the signal. This gives an idea about the variations of the signal power within different frequency ranges. The PSD of deterministic signals are derived by doing a fast Fourier Transform of the signal's autocorrelation. A much better estimate of a signal's PSD can be obtained by using Welch's method [18].

2.3.1 PSD Estimate from Autocorrelation

Autocorrelation [19] of a signal $x(k)$ is defined by,

$$r_{xx}(k) = \sum_{n=-\infty}^{\infty} x(n+k)x(n) \tag{2.1}$$

The autocorrelation is basically the cross-correlation of the signal with itself and gives an idea about the non-randomness of the signal. In order to determine the power spectral density of a signal, the autocorrelation of the signal is determined.

Taking the Fast Fourier Transform of the autocorrelation gives the power spectral density, which is given by,

$$S_x(f) = \int_{-\infty}^{\infty} r_{xx}(\tau)e^{-2\pi i f \tau} d\tau \tag{2.2}$$

When this method is applied to an EEG signal, which is not a deterministic signal, the spectral estimates calculated will include a lot of noise. This was improved upon in the Welch's PSD estimate.

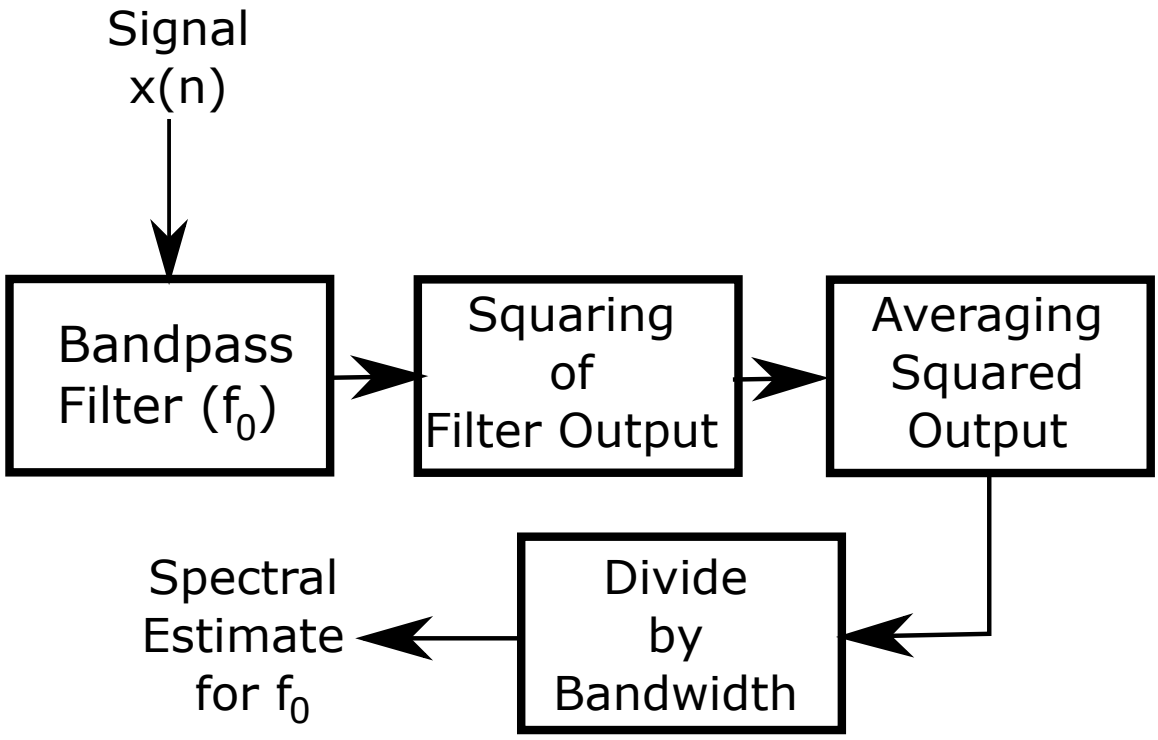


Figure 2.11: Block Diagram for Filter, Square and Average PSD Estimation Approach.

2.3.2 Filter, Square, and Average Approach for PSD Estimation

While PSD estimate from autocorrelation are usually used for deterministic signals, for signals that have random components, a form of averaging and smoothing is required [20]. This PSD estimation can be designed as a form of filtering, where the signal is passed through a bandpass filter. The filter output is then squared and averaged, and is then divided by the filter bandwidth.

In order to meet the purpose of studying the variation of the signal's frequency content with frequency, it is necessary to choose filters of different central frequencies. By determining the estimate for a frequency f across a range of frequencies and then plotting them, the PSD estimate for the full range of frequencies in the signal can be shown. Welch's method for power spectral density estimation details the steps in involved in mathematically modelling this process.

2.3.3 Welch's Power Spectral Density Estimate

Peter D. Welch [18] proposed a method for estimating the PSD of a signal that looked to improve upon periodogram spectrum estimation. A method was outlined for the application of the

fast Fourier transform algorithm to the power spectra estimation by sectioning the signal, taking modified periodograms of these sections and averaging them. This method is also computationally efficient.The steps involved in Welch's method of Power Spectral density estimation are detailed below.

- A signal of length L is divided into K segments X_k of length N, such that these segments overlap each other by M samples. The segments are chosen such that it covers the entire signal.
- After obtaining the segments, a data window W is applied to each segment to get a sequence.

$$S_k(i) = X_k(i) * W(i) \quad \forall\ i = 0, 1....., N-1 \tag{2.3}$$

  where N is the length of each segment.

- The Fourier Transforms of the windowed sequences are obtained.

$$F_k(n) = \frac{1}{N} \sum_{j=0}^{N-1} S_k(j) e^{\frac{-2\pi i j n}{L}} \tag{2.4}$$

- The modified periodograms are then obtained.

$$Y_k(f_n) = \frac{N}{U} |F_k(n)|^2 \tag{2.5}$$

  where,

$$f_n = \frac{n}{N} \tag{2.6}$$

$$U = \frac{1}{N} \sum_{j=0}^{N-1} W^2(j) \tag{2.7}$$

- The periodograms are then averaged to obtain the spectral estimate.

$$P(f_n) = \frac{1}{K} \sum Y_k(f_n) \tag{2.8}$$

The power spectral density estimate calculated using Welch's method was used for extracting features. The variations of the power spectral densities within classes AEP's and Non-AEP's have

been visualized in the Fig. 2.12 - Fig. 2.16. Each figure shows the power spectral density for each confidence level (201-205). Although there are similarities in the shape of the PSD for each confidence level, there are instances where the power levels vary.

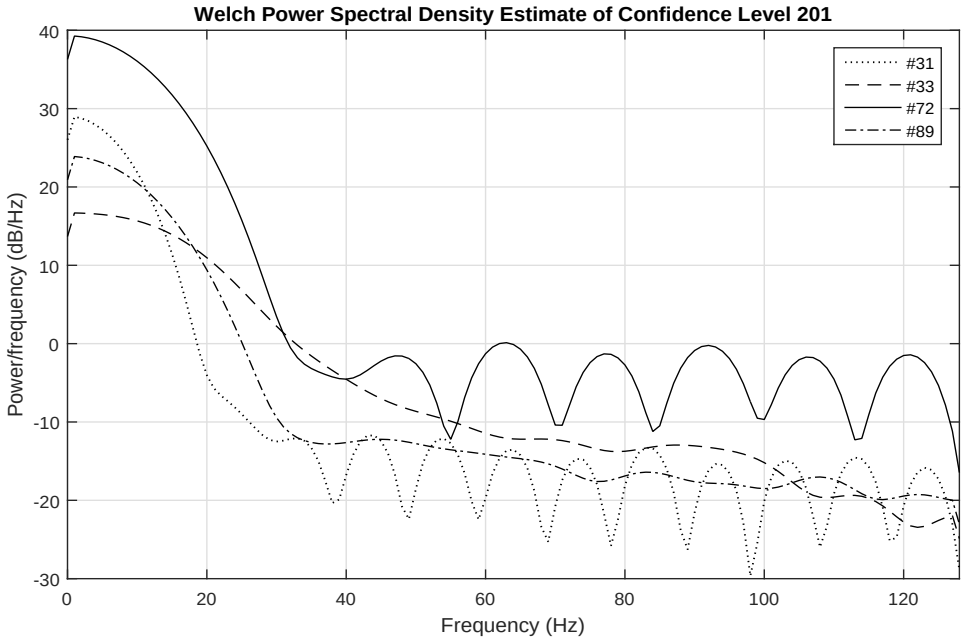


Figure 2.12: Power Spectral Densities of Confidence Level 201 Examples.

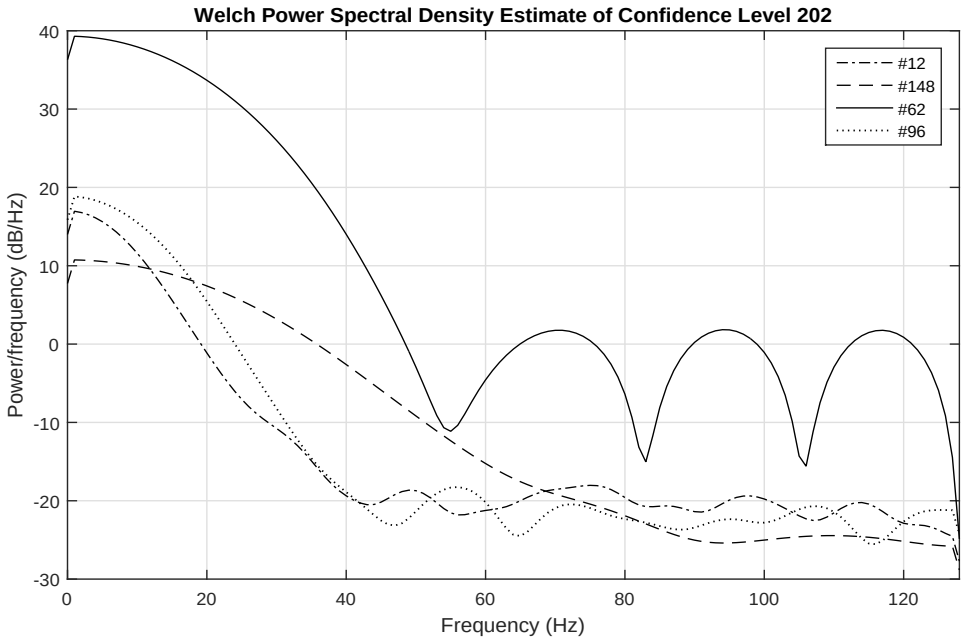


Figure 2.13: Power Spectral Densities of Confidence Level 202 Examples.

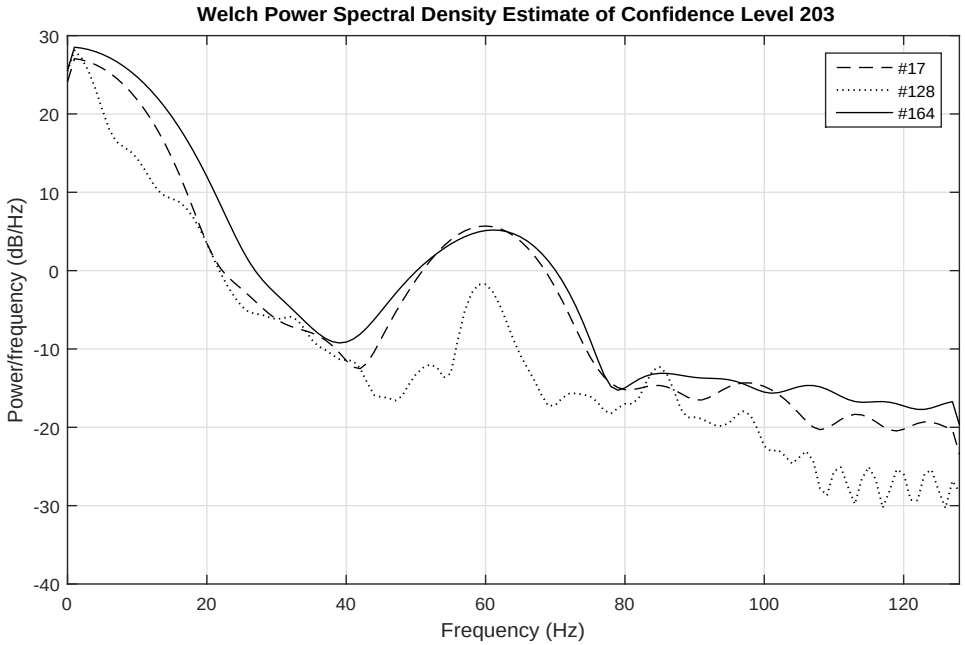


Figure 2.14: Power Spectral Densities of Confidence Level 203 Examples.

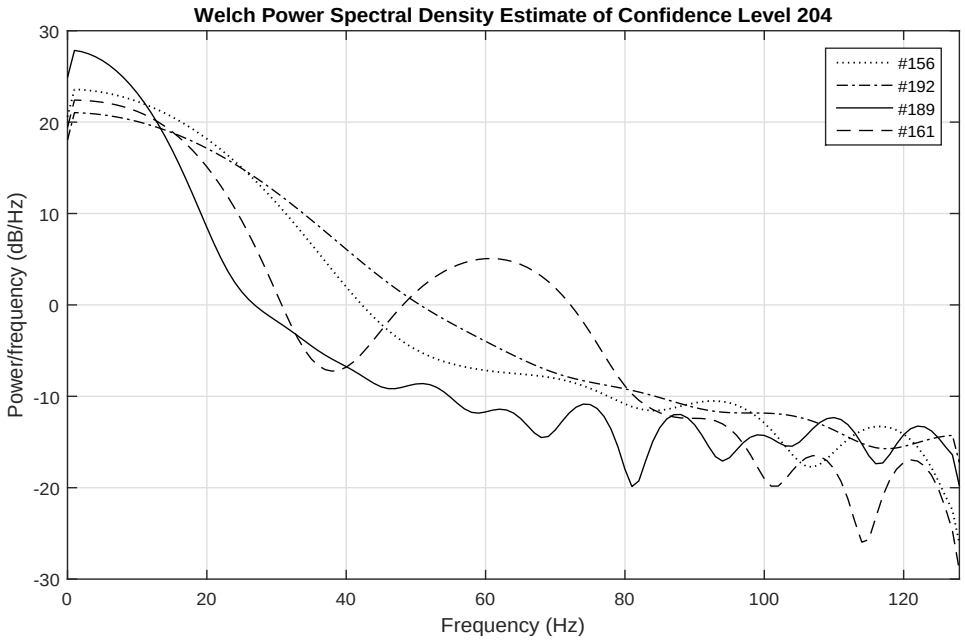


Figure 2.15: Power Spectral Densities of Confidence Level 204 Examples.

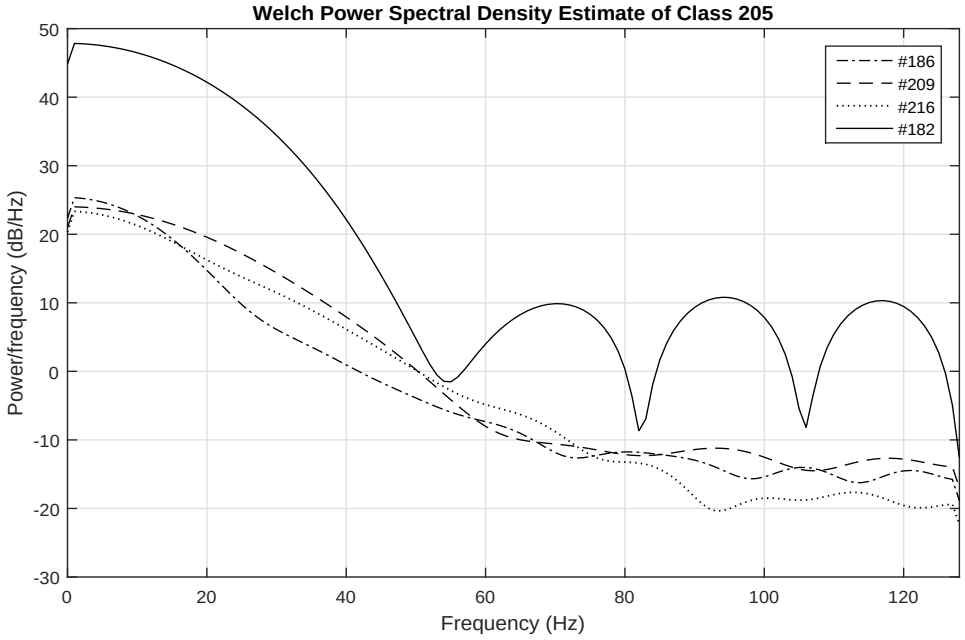


Figure 2.16: Power Spectral Densities of Confidence Level 205 Examples.

A careful observation of all five also shows that there are some similarities and variation in

the PSDs for each as shown in Fig. 2.17, where an example of each confidence level is used. This also gives an idea regarding the variation of PSD among the yellow-boxed signals. Fig. (2.18) shows the different PSDs obtained from five examples of the class containing not to be yellow boxed segments.

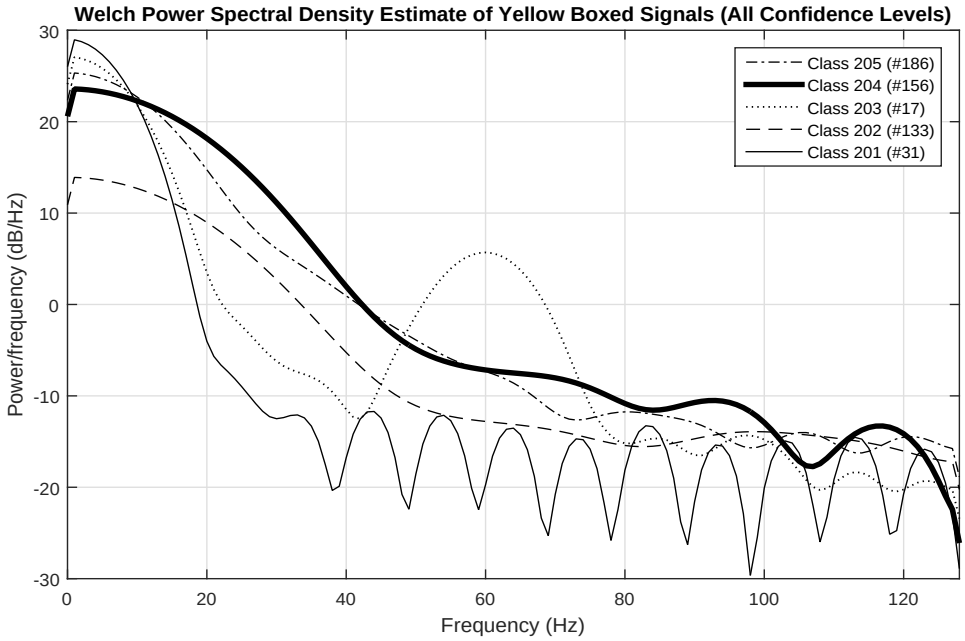


Figure 2.17: Power Spectral Densities of Yellow Boxed Examples (One From Each Confidence Level).

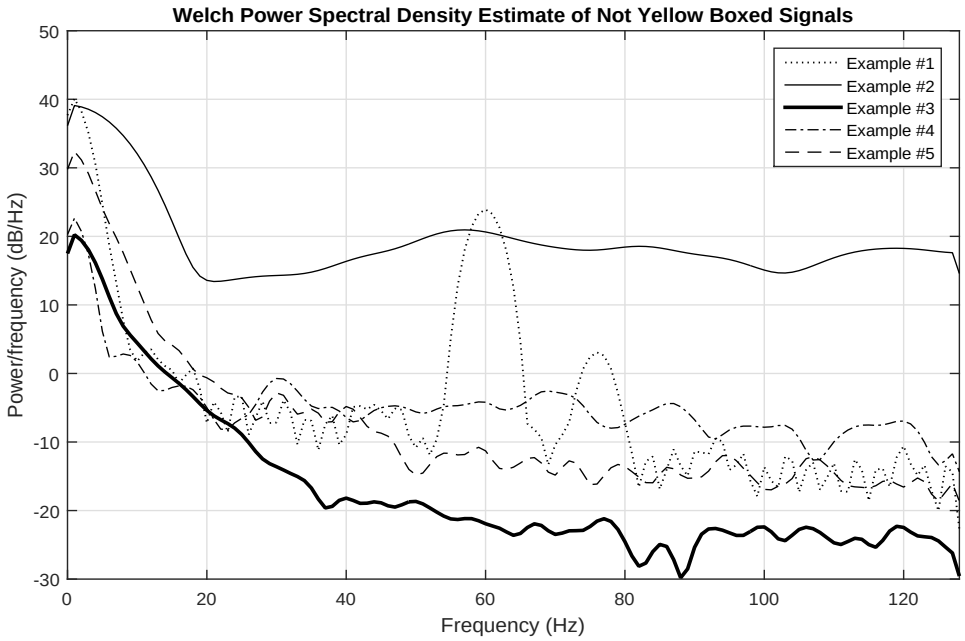


Figure 2.18: Power Spectral Densities of Not Yellow Boxed Examples.

2.3.4 Spectral Entropy

Entropy is defined as a measure of randomness or lack of order. This resulted from Claude Shannon's development of "Information Entropy".This concept was applied to determine changes in the EEG power spectrum. Spectral Entropy [21] is an application of Shannon's Entropy to the Power Spectrum Density of a signal. Power spectral entropy is information entropy that is able to quantify the spectral complexity of an uncertain system[21].

If the power spectral estimate is denoted by P, then the Spectral Entropy is calculated using the formula:

$$S = \sum p_k log(p_k) \tag{2.9}$$

where p_k is the normalized PSD estimates for k frequencies. The estimates are normalized in such a way that the sum of the normalized estimates is 1.

$$\Sigma p_k = 1 \quad \forall \quad k = 1, ...N \tag{2.10}$$

This normalization is achieved by using the formula:

$$p_k = \frac{P_k}{\sum P_k} \quad \forall \quad k \tag{2.11}$$

The spectral entropy provides an idea about the complexity and unpredictability of the signal. Considering that epileptic transients are unpredictable in nature and due to the occurrence of inter individual variations in the EEG frequencies, it is a significant characteristic that can be used for identifying epileptic transients.

2.3.5 Spectral Shape Descriptors

EEG analysis techniques that make use of the shape of the signal have been researched. There are also considerable similarities and variations in the power spectral density of epileptic transients. In order to exploit these characteristics, two heuristics are used as spectral shape descriptors. These two characteristics are spectral flatness and skewness.

Wiener Entropy or Spectral Flatness is commonly used in audio signal processing techniques. It is a measure of how tonal (pointy) or noisy (flat) the spectrum is. The spectral flatness is defined as the ratio of the geometric mean of the power spectrum to the arithmetic mean of the power spectrum. The geometric mean is the s^{th} of the product of s numbers. The mathematical formula is given by:

$$W = \frac{(\Pi P_k)^{\frac{1}{N}}}{\frac{1}{N}\sum P_k} \tag{2.12}$$

Spectral Skewness is a measure of asymmetry about the mean of the spectrum. The spectral skewness gives an idea regarding in which range of frequencies the power of the signal is concentrated. The skewness of a spectrum is calculated using,

$$Skew = \sqrt{N}\frac{\sum(P_k - P_{mean})^3}{(\sum(P_k - P_{mean})^2)^{\frac{3}{2}}} \tag{2.13}$$

The mean of the spectral density estimates is denoted by P_{mean} in Eqn. 2.13. The above two heuristics are used as spectral descriptors [13] to determine similarities in the power spectral density between signals.

2.4 Artificial Neural Networks

2.4.1 Feed forward Neural Networks

Feed forward networks are a popular format of artificial neural network structures. The design of a feed forward network involves the consideration of a number of factors. These forward networks are trained by supervised learning to provide outputs in a desired way. A popular method of training feed forward network is the Generalized Delta Rule based on back propagation.

The feed forward network [22] is composed of a hierarchy of units, that are organized to form consecutive layers. The input is fed to these interconnected layers and an output is obtained based on the mathematical relation between the input values and the weights between the layers. Besides the input and output layer, which contains the input and output units, the layers between them are called hidden layers. These hidden layers can be of more than one layer and contain hidden units. Most problems that use feed forward networks have successfully used a single hidden layer in the past.

There is no set rule that defines the number of hidden units per layer. Common rules of thumb include choosing a number greater or equal than the number of input units. A popular rule of thumb is to have *(2d+1)* units in the hidden layer [22], where d is the number of input units. The units may or may not have a bias unit. A bias unit is stand alone input to a unit in the hidden and output layers. It is similar to an unit input with a weight w.

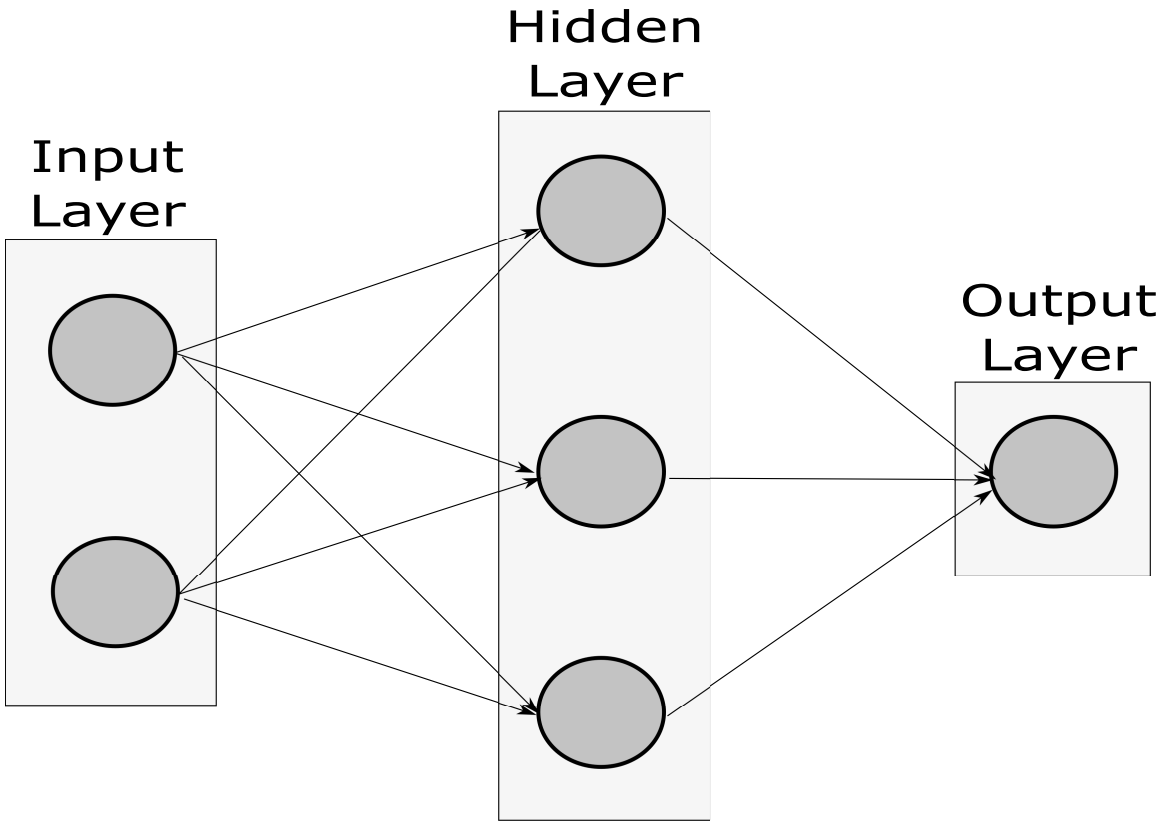


Figure 2.19: Artificial Neural Network.

2.4.2 Classifier Networks

Feed forward networks have been used to define complex functions for the purpose of classification. These feed forward networks usually have a binary output unit using values (0,1) or (-1,1), based on the activation function used. An activation function defines the output of the unit as ON/OFF based on the input to the unit. Since the sigmoid activation saturates at 0 and 1 while the tanh saturates at -1 and +1, the tanh activation function has been used so as to avoid zero output in units.

For a given set of weights w connecting weights from one layer to a unit j in the next layer, the output of the unit for a tanh activation function is given by,

$$net_j = w^T \bar{i} \tag{2.14}$$

where $\bar{i}$ is the vector containing the inputs to unit j.

$$o_j = tanh(net_j) \tag{2.15}$$

The network is trained using the Generalized Delta Rule with back propagation of the error function. Instead of initializing the weights randomly, the weights are initialized using the random optimization method.

2.4.3 Random Optimization Method

Random optimization method is a method used for the determination of weights of neural networks by randomly searching the surface of error for the best possible local minimum. This local minimum could be the global minimum or the weights could be used as initial weights for a supervised training algorithm to find a better minimum in the error surface. The weights in the region of search belong to a set R.

The steps involved in the random optimization method are as follows:

- Select an initial set of weights w(0). Let i be the iteration number.
- Generate a Gaussian random vector ξ.
- If $w(i) + \xi \in R$, and the training set error function E of $w(i) + \xi$ is less than the error function of w(i), then weights are updated to $w(i) + \xi$.
- i is incremented by 1 and the steps are repeated until maximum iterations have been completed.

The error function used is the same as the error function used for back propagation which is explained in the next section.

2.4.4 Training by Generalized Delta Rule

The generalized delta rule looks for a local minimum that is proportional to the error function. The procedure for generalized delta rule (GDR) with backpropagation are as follows:

- Obtain the outputs o_j for all the units in the network by feeding input vector i.
- The weights to the output layer are updated using Eq. 1 and Eq. 2 from Table 2.5.

- The weights to the hidden layer are then updated using Eq. 1 and Eq. 3 from Table 2.5.
- Keep repeating until the error is below a threshold value or the maximum number of iterations have been reached.

Eq. 1	Error	$E^p = \frac{1}{2}\sum_j (t_j^p - o_j^p)^2$
Eq. 2	Weight Correction	$\delta^p w_{ji} = \epsilon \sigma_j^p o_i^p$
Eq. 3	For Output Units	$\sigma_j^p = (t_j^p - o_j^p) f_j'(net_j^p)$
Eq. 4	For Internal Units	$\sigma_j^p = f_j'(net_j^p) \sum_n \sigma_n^p w_{nj}$

t - Target Output, f'- Derivative of activation function

Table 2.5: GDR Training Equations[22].

The learning rate ϵ does not have a particular choice. Too high a learning rate will cause the neural network to saturate at sub optimal solution while too low a learning rate will slow down the training process. The neural network can be trained by feeding one pattern vector at a time from the training set (train by pattern) or correcting the weights after determining the weight corrections for all the inputs and then applying them (train by epoch). When training by epoch, the sum of the squares of all the errors (TSS Error) are calculated and used as a measure to show the training of the neural network.

Chapter 3

Classification of Yellow Boxed Signals

3.1 Method

For the classification of Yellow-Boxed signals, the signal is to be classified as either an AEP or a Non-AEP. For this purpose, the signals obtained from the Yellow Boxing Detection method (discussed in Chap. 4) are to be used and classified on the basis of its features. A neural network with a single hidden layer was used for this purpose. After translating the five confidence levels provided by the EEG annotators to a much more simple two class problem, the neural network was trained using the extracted features. Since the data set contained features extracted from segments of varying sample length as shown in the Fig. 3.1, it gave an option to classify segments of different time periods that is obtained from the detection method.

3.2 Data Analysis

The data for yellow boxed signal classification were modified from a five class set to a two class set. Class AEP contained AEP signals (Confidence levels of 204/205) and Class Non-AEPs contained non-AEPs (Confidence levels 201/202). This data set containing 235 signals was normalized to a frequency of 256 Hz and were of length ranging from 40 to 350 samples.

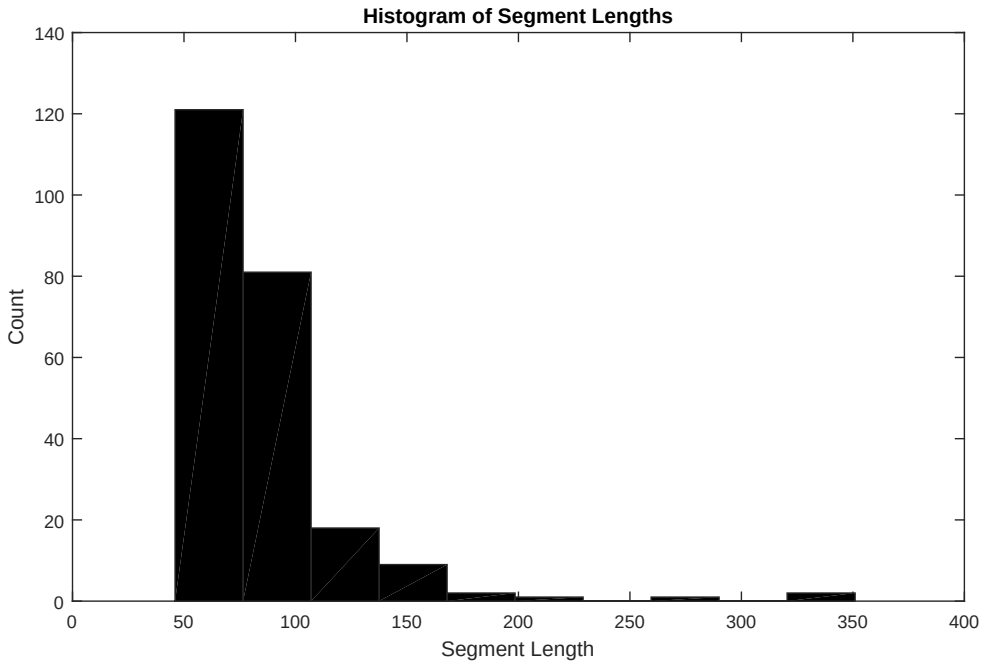


Figure 3.1: Histogram of the Segment Length of potential AEPs.

A histogram (Fig. 3.1) of the signal length showed that over 85 percent of the signals were had a signal length under 100 samples. This data set was trimmed to rid of signals that annotators were unsure about. This resulted in a final dataset containing 228 signals.

3.3 Feature Selection and Extraction For Yellow Box Classification

Past research has been performed that use the spectral powers of the EEG signal as a whole and the ratio of the powers of the sub bands along with the EEG signal's spectral entropy. These sub bands have been clinically defined in EEG literatures and are called Delta (δ), Theta (ϑ), Alpha (α), Beta (β) and Gamma (γ). An absence of strict frequency ranges was found while studying these details. In the end, the frequency ranges defined by [6] were used. These ranges are Delta (0.5-4Hz), Theta (4-8Hz), Alpha (8-13Hz), Beta (13-25Hz) and Gamma (25-40Hz).

It has been discussed by Dastidar [23] that the sub bands give much better information

regarding the underlying variations in EEGs when they are analyzed separately. Thus, Spectral entropy of each of the five sub bands were calculated from the PSD of the signal and their ratios with respect to the spectral entropy of the EEG signal were determined.

In addition to this, the heuristic spectral shape descriptors, spectral flatness and skewness were also determined for the signal's PSD. In order to define a classifier that can be trained to differentiate between AEPs and Non-AEPs, the following features have been used. They are

- Spectral Entropy of the signal and the five sub bands, according to Eq. 2.9.
- Ratio of the Spectral Entropy of signal sub bands to that of the signal.
- Spectral Flatness of the Power Spectrum, according to Eq. 2.12.
- Spectral Skewness of the Power Spectrum, according to Eq. 2.13.

This resulted in a 13-dimensional feature vector for each of the signals in the data set. The final feature set containing 228 such feature vectors was then used for training and testing of the neural network.

3.4 Neural Network

3.4.1 Design

A MLFF neural network was designed with one hidden layer. With an input feature vector of length 13, this resulted in a hidden layer containing 27 units and one output unit. The number of hidden units was decided by using a "Rule of thumb". This rule states that the the number of hidden units is *(2d+1)*, where d is given by the dimension of the input vector.In this case, the d for classification data set is 13 and hence *(2d+1)* = 27. Each hidden unit and the output unit has a bias.

A hyperbolic tangent activation function was used for each unit. The output unit also used a hyperbolic tangent function. Since tanh saturates at [-1,1], both sides of zero are used to point to one of the two classes. The desired output for a signal to be classified as an ET is +1, while it is -1 for a Non-AEP.

3.4.2 Training

The training of the neural network was done in two steps. In the first step, Random Optimization Method was used to bring the network weights to an localized area of minimum. From this point, the neural network was trained by back propagating the error to reach the best possible minimum error.

By using the weights obtained from the random optimization method as the initial weights for GDR, the neural network trains itself to look for a better local minimum.

3.4.3 k-Fold Cross Validation

Cross validation [24] is a method for evaluating the performance of a model and gives a better understanding of performance than the error. In k-fold cross validation, the data are divided into k folds or subsets. Then, $k-1$ folds are used for training the neural network and then tested on the fold that was left out. This is repeated until the trained neural network has been tested on all the folds.

In this way, each fold acts as the testing set exactly once and as the training set $k-1$ times. This gives an statistically valid prediction regarding how the trained neural network would perform when it sees data not included in the training set.

3.5 Results

3.5.1 Training Parameters

The weights of the hidden layer and the output layer were initialized with values between ± 0.5. The initialized neural network was then trained to find a generalized minimum location using Random Optimization method.

The neural network weights obtained from ROM was then used in a back propagating learning method. The neural network was trained with a learning rate of 10^{-3} and a momentum rate of 0.2. In order to allow the neural network to saturate at a minimum such that the error function is low, a maximum iterations of 100000 iterations was used.

3.5.2 Creating k-Folds

The ratio of feature vectors representative of Non-AEPs to AEPs was 1.64 and hence, the data set was not balanced to contain equal representations of AEPs and Non-AEPs. The above ratio was maintained while dividing the data into k number of folds for cross validation. The value of k was varied between 4 and 9. The number of feature vectors in each class and their respective classes are tabulated in Table 3.1.

k value	Subsets	Class AEP	Class Non-AEP	Ratio (Non-AEP:AEP)	Total
4	1	23	35	1.5217	58
	2	22	34	1.5455	56
	3	22	35	1.5909	57
	4	22	35	1.5909	57
5	1	18	28	1.5556	46
	2	18	27	1.5000	45
	3	17	28	1.6471	45
	4	18	28	1.5556	46
	5	18	28	1.5556	46
6	1	15	23	1.5333	38
	2	14	23	1.6429	37
	3	15	23	1.5333	38
	4	15	23	1.5333	38
	5	15	23	1.5333	38
	6	15	24	1.6000	39
7	1	12	20	1.6667	32
	2	13	20	1.5385	33
	3	13	20	1.5385	33
	4	13	20	1.5385	33
	5	13	20	1.5385	33
	6	13	20	1.5385	33
	7	12	19	1.5833	31

k value	Subsets	Class AEP	Class Non-AEP	Ratio (Non-AEP:AEP)	Total
8	1	11	18	1.6364	29
	2	11	17	1.5544	28
	3	12	17	1.4167	29
	4	11	18	1.6364	29
	5	11	17	1.5544	28
	6	11	17	1.5544	28
	7	11	17	1.5544	28
	8	11	18	1.6364	29
9	1	10	15	1.5000	25
	2	10	16	1.6000	26
	3	10	15	1.5000	25
	4	9	16	1.7778	25
	5	10	16	1.6000	26
	6	10	15	1.5000	25
	7	10	16	1.6000	26
	8	10	15	1.5000	25
	9	10	15	1.5000	25

Table 3.1: Distribution of Class Vectors during k-fold Cross Validation.

The above subsets were used for each value of k for validation. These subsets were then combined to form different variations of the training and testing set. The training and testing sets were then modified to exhibit maximum variance by using PCA while not reducing the feature vector to a lower dimensional vector of length k.

3.5.3 Classification Performance

The neural network is trained on the training data set and training can be visualized by plotting the TSS error with respect to the iterations. This graph has been plotted for each value of k. For each value of k, the figures show the error plot for each k^{th} training set used. The TSS error

plots can be found from Fig. 3.2 - Fig. 3.7 for each value of k. Each of the graphs shows the error plot for the k ANNs for each k. From the graph, it was inferred that the neural network trains well on the training set reaching local minimum. This shows that the neural network has been trained to differentiate between AEPs and Non-AEPs and the performance of this trained neural network is further discussed.

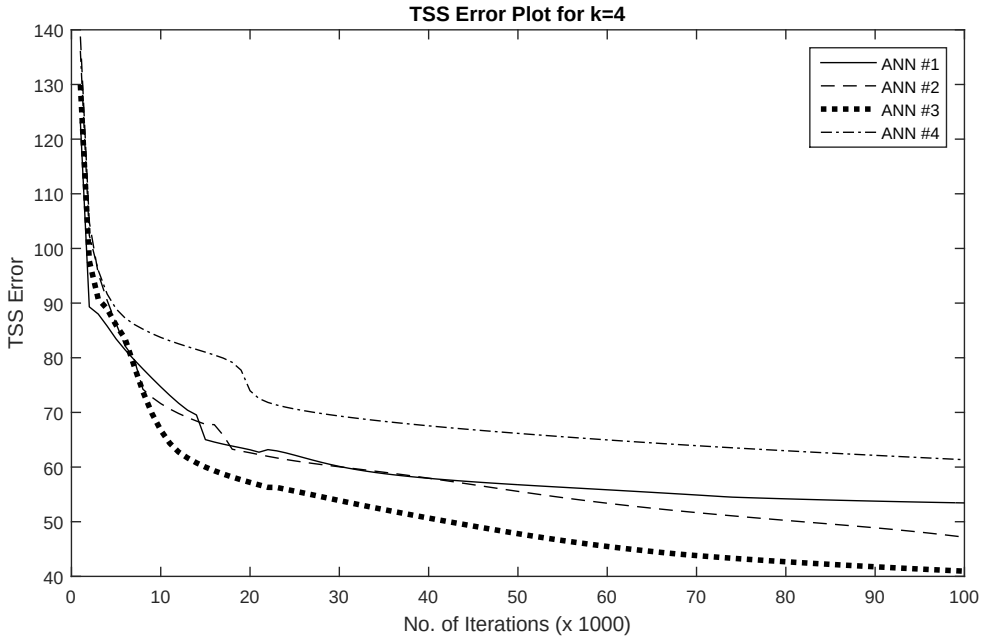


Figure 3.2: Training of Neural Network for k=4.

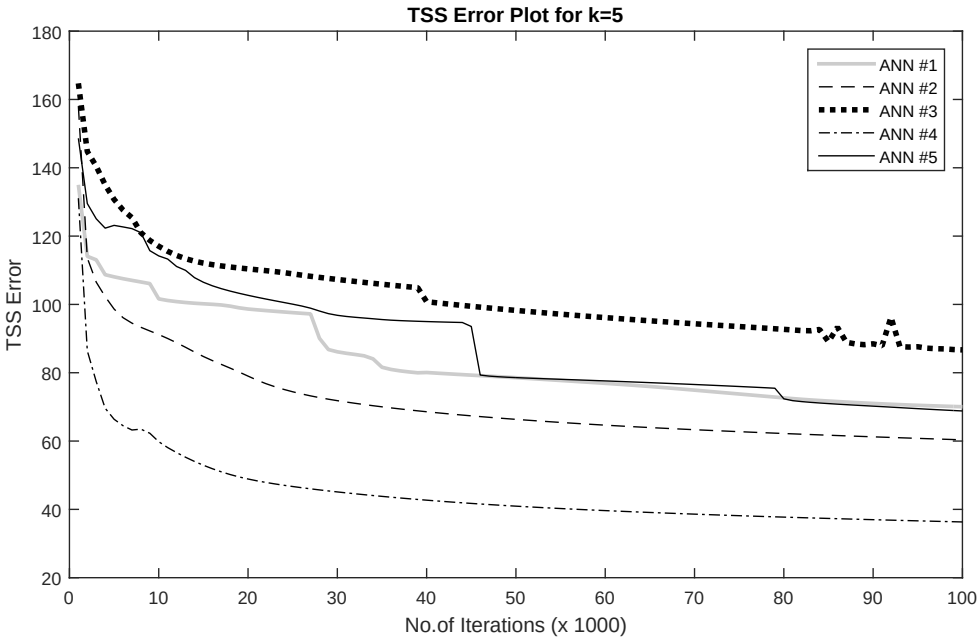


Figure 3.3: Training of Neural Network for k=5.

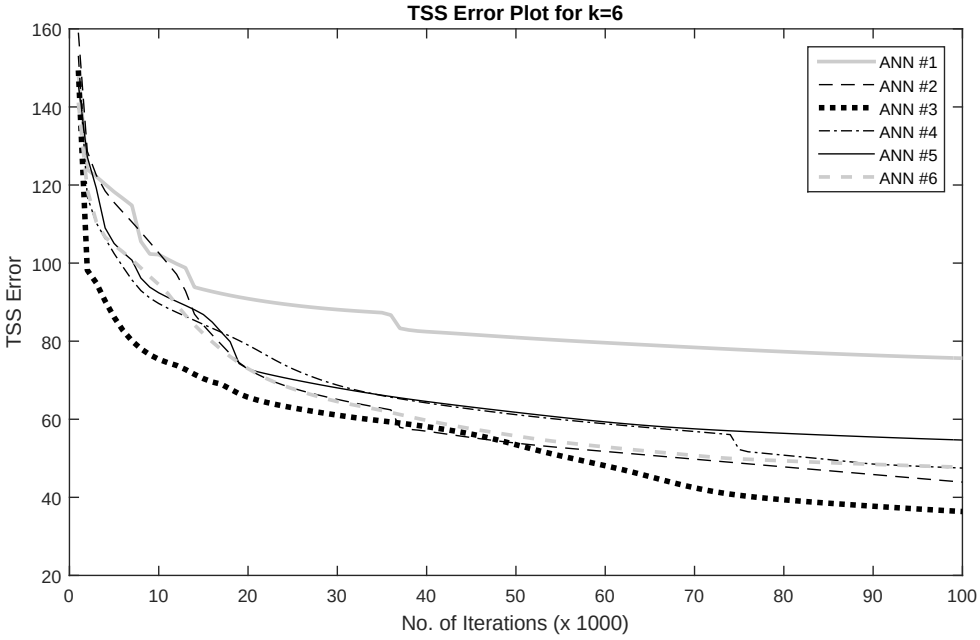


Figure 3.4: Training of Neural Network for k=6.

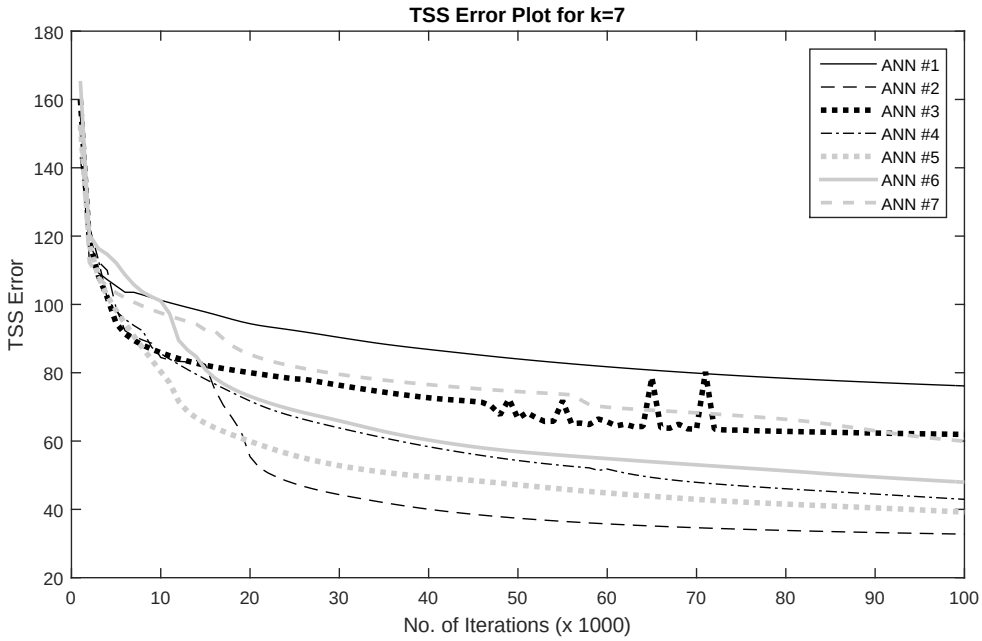


Figure 3.5: Training of Neural Network for k=7.

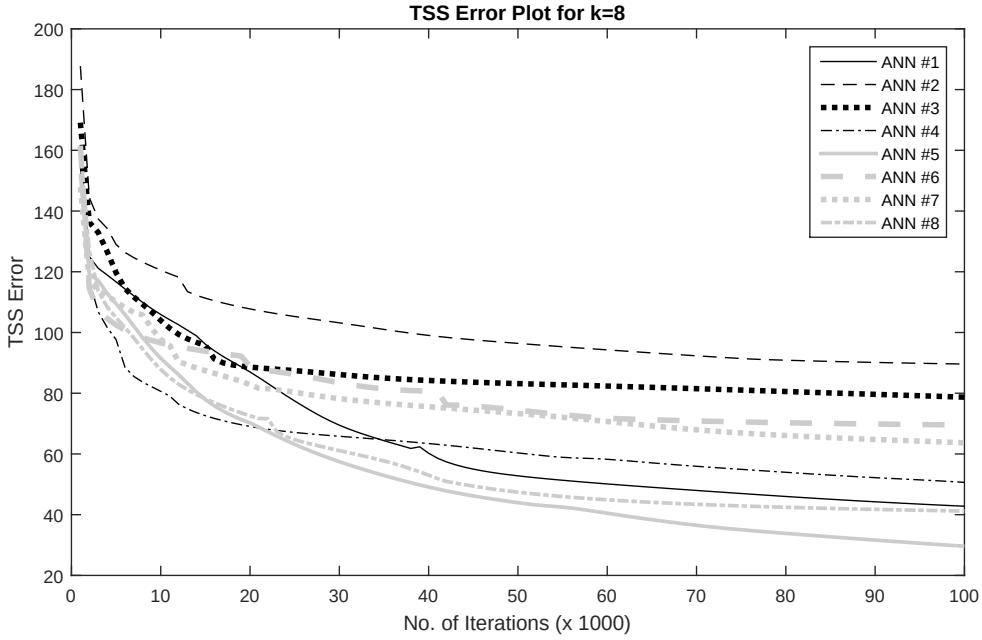


Figure 3.6: Training of Neural Network for k=8.

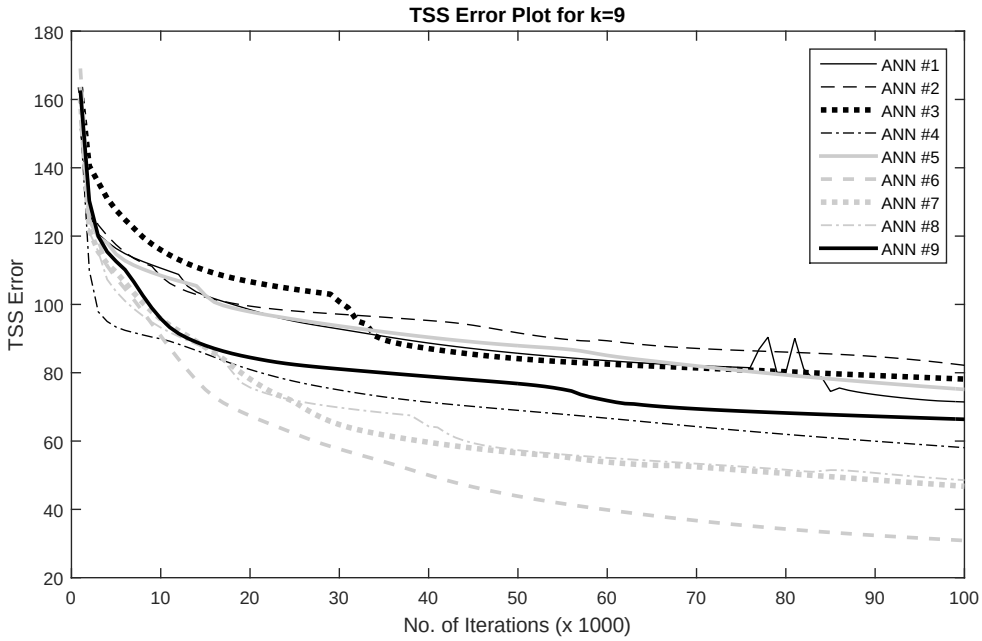


Figure 3.7: Training of Neural Network for k=9.

The results of the neural network after training are tabulated Table 3.2. A confusion matrix is drawn up for the neural network classifier and is represented in Fig. 3.8.

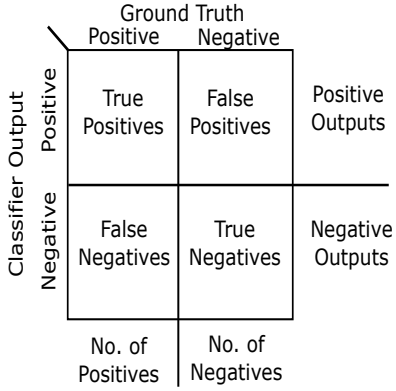


Figure 3.8: Confusion Matrix Depiction.

The definitions for the terms are explained below.

- A **True Positive** is obtained when the classifier output and the ground truth of the input vector are both positive.
- A **True Negative** is obtained when the classifier output and the ground truth of the input

vector are both negative.

- A **False Positive** is obtained when the classifier output is positive and the ground truth is negative.
- A **False Negative** is obtained when the classifier output is negative and the ground truth is positive.

The performance of the neural network was evaluated using performance parameters [25] derived from the confusion matrix. These performance parameters are,

- **Sensitivity :** Called the True Positive Rate gives the rate at which positives are correctly identified.

$$Sensitivity(TPR) = \frac{True \quad Positives}{No. \quad of \quad Positives} \tag{3.1}$$

- **Specificity :** Called the True Negative Rate gives the rate at which negatives are correctly identified.

$$Specificity(SPC) = \frac{True \quad Negatives}{No. \quad of \quad Negatives} \tag{3.2}$$

- **Precision :** Called the Positive Predictive Value gives the accuracy rate of the detected positives.

$$Precision(SPC) = \frac{True \quad Positives}{Positive \quad Outputs} \tag{3.3}$$

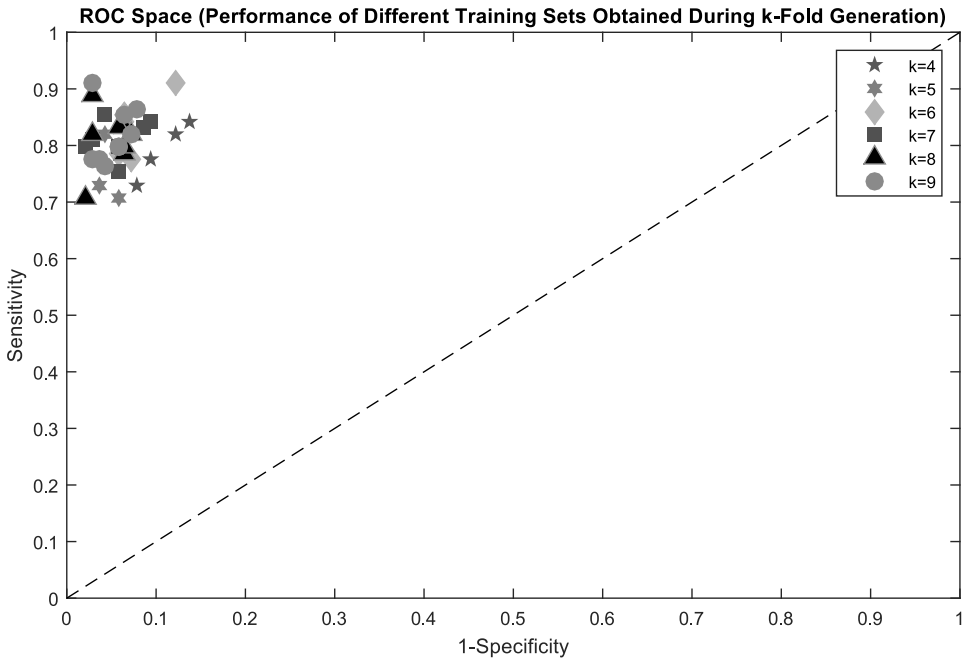
k value	Trial #	Sensitivity(%)	Specificity(%)	Precision(%)
4	1	73.03	92.09	85.33
	2	84.27	86.33	79.79
	3	87.77	82.02	81.11
	4	84.15	90.65	77.53
5	1	70.79	94.24	88.73
	2	78.65	93.53	88.61
	3	73.03	96.40	92.86
	4	82.02	95.68	92.41
	5	78.65	93.53	88.61
6	1	77.53	92.81	87.34
	2	85.39	93.53	89.41
	3	87.77	91.01	82.65
	4	89.41	93.53	85.39
	5	84.27	93.53	89.29
	6	78.65	94.24	89.74
7	1	75.28	94.24	89.33
	2	84.27	90.65	85.23
	3	79.78	94.24	89.87
	4	80.90	97.12	94.74
	5	85.39	95.68	92.68
	6	79.78	97.84	95.95
	7	83.15	91.37	86.05

k value	Trial #	Sensitivity(%)	Specificity(%)	Precision(%)
8	1	82.02	97.12	94.81
	2	70.79	97.84	95.45
	3	78.65	93.53	88.61
	4	83.15	93.53	89.16
	5	88.76	97.12	95.18
	6	79.78	93.53	88.75
	7	82.02	91.81	87.95
	8	83.15	94.24	90.24
9	1	76.40	95.68	91.89
	2	79.78	94.24	89.87
	3	79.78	94.24	89.87
	4	85.39	93.53	89.41
	5	77.53	96.40	93.24
	6	91.01	97.12	95.29
	7	82.02	92.81	87.95
	8	86.52	92.09	87.54
	9	77.53	97.12	94.52

Table 3.2: Performance Parameters of Trained Neural Networks.

From Table. 3.2, it was observed that the sensitivity and specificity values range from 70% to 90% and above 90% respectively. A better idea about the performance of the trained neural networks can be obtained by visualizing it as an ROC space [25]. An ROC space is a plot drawn between the sensitivity and (1-specificity) values. This space can be used to infer the performance of trained models.

Considering the fact that each training set contains a different combination of k-folds, an ROC space can be plotted using the different training sets as a variation component. This ROC space is shown in Fig. 3.9. This ROC space gives an idea regarding how good the performance of the trained neural network is. An ideally trained neural network would have both sensitivity and specificity of 1.0 or 100%. Thus, a plot in the upper left corner of the ROC space represents an



This ROC space was generated while considering the difference in the feature vectors present in the training set during k-Fold cross validation. Each symbol in the legend represents a value of k. For example, the performance of the trained neural networks from 4-fold cross validation are represented by pentagrams.

Figure 3.9: ROC Space for Training Sets From k-Fold Validation.

ideally trained network. From Fig. 3.9, it can be noted that the plots are concentrated towards the left in the upper half of the ROC space. The performance estimates for each k in the cross validation is tabulated in Table 3.3.

k value	**Sensitivity(%)**	**Specificity(%)**	**Precision(%)**
4	79.21	89.21	82.64
5	76.63	94.68	90.24
6	83.71	92.57	87.97
7	81.22	94.45	90.55
8	81.04	94.96	91.27
9	81.77	94.80	91.06

Table 3.3: Performance Estimates for Classification.

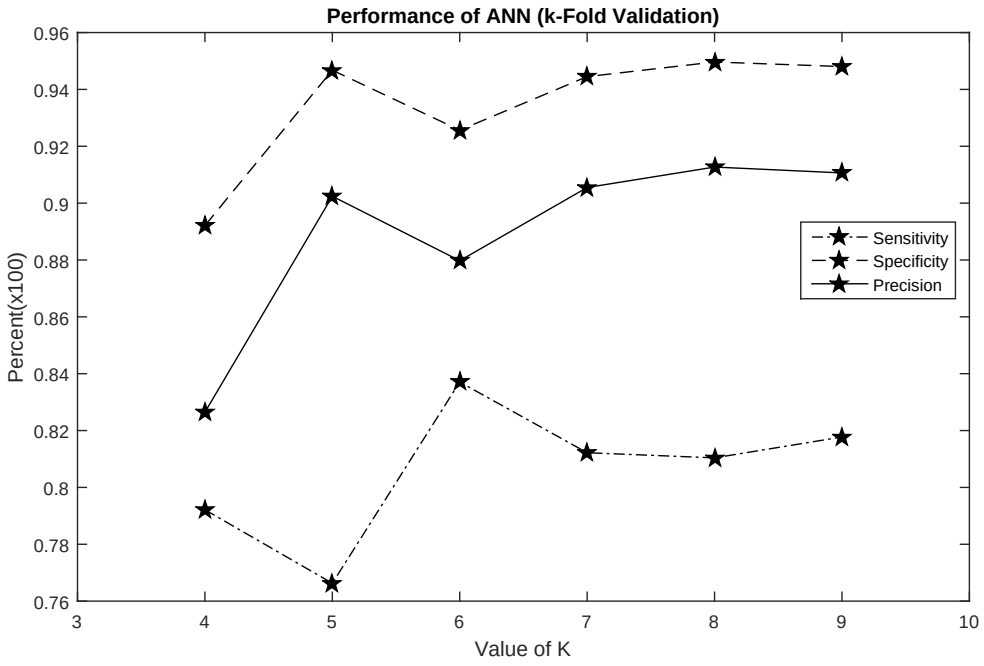


Figure 3.10: Plot of Performance Estimates of Each k.

From Table. 3.3, the average performance of a trained ANN, from the using the training set generated from the data set, were estimated. Thus, a neural network trained using this data set is estimated to have a sensitivity of 80.60%, specificity of 93.44% and a precision of 88.96%. The plot in Fig. 3.10 shows how the performance parameters have varied over the different values of k. As k increases, the folds tend to become smaller resulting in larger training sets and small test sets. Hence, there appears to be an improvement in the performance as a result of the larger training set.

3.6 Conclusion

The results documented mainly revolved around using three characteristics of spectral densities. With an average sensitivity of 80.60% and specificity of 93.44%, the investigation of other features extracted from the power spectral density could result in higher and ideal performance parameters.

From the trained neural networks, the best neural network was chosen. The network, which had the best performance in comparison to the others was selected and used as the classificatoin

neural network for the purpose of classifying yellow box segments. For this purpose, ANN #6 from k $\bar{9}$ fold cross validation was used (from Table 3.2, sensitivity = 91.01, specificity = 97.12 and precision = 95.29).

Chapter 4

Detection - Yellow Boxing

4.1 Method

In order to use a neural network for the purpose of yellow boxing potential AEPs, a sliding window is placed across the signal. This sliding window is of length n and is slid across the signal with an overlap of m. A neural network is used to determine if the segment of the signal inside a window is a potential AEP or not.

If the segment is an potential AEP, the window skips to the end of the segment and the overlapping movement of the window is continued. If two adjacent segments are determined to be potential AEPs, they are concatenated to form a longer segment. This allows the detected segment to grow in place of a sliding window of variable size.

4.2 Data Analysis

4.2.1 Dataset Creation

In order to create a dataset representative of the problem, examples of two classes, yellow-boxed and not yellow-boxed, were selected from the EEG record files using the details of the 235 annotations provided by the neurologists. The 235 annotations used for the classification problem form the representation of the class yellow-boxed for detection. Hence, any part of the signal that was not annotated will become a part of the not yellow-boxed class.

In order to determine the variation in the performance of different window sizes, two window sizes were used. A window of length 60 and 100 samples were chosen. These lengths were used based on the reasoning that a window of 60 samples could catch the smallest of potential AEPs while a window of sample length 100 could improve the speed of the detection process by covering a larger area. Also, from previous analysis of the data, it was found that just over half the yellow boxed segments had a sample length under 50 and the next largest grouping of yellow boxes were found in the range of 50-100 samples.

The non-annotated part of the signal was extracted and then divided into segments of length N. These segments were then used to represent the not yellow boxed class signals. The yellow boxed (annotated) signals were then divided into two. This decision led to the possibility of forming pairs of two classes by considering the adjacent segment of a segment of interest. The pairs formed are,

- Not yellow-boxed segment followed by another not yellow-boxed segment.
- Not yellow-boxed segment followed by an yellow-boxed segment.
- Yellow-boxed segment followed by a not yellow-boxed segment.
- Yellow-boxed segment followed by another yellow-boxed segment.

4.2.2 Use of Contextual Information of Yellow Boxed Segments

The pairing of adjacent segments led to the possibility of taking into consideration what happens around an yellow box, leading to contextual information. This was so as to mimic the process of how physicians execute the process of yellow-boxing. The physicians do not look at the segment as a stand alone representation but pass through the entire signal marking up potential AEPs. Thus, two different sets were created. One where the features were extracted with respect to the pre-contextual information of the segment of the interest and the second set where the post-contextual information was considered. The details of the two datasets are as given below and a representative figure is shown in Fig. 4.1.

- **Pre-contextual Information Dataset:** This data set divides the data set into two classes: A class containing {yellow-boxed, yellow-boxed} pairs and {not yellow-boxed, yellow-boxed} pairs and a second class containing {not yellow-boxed, not yellow-boxed} and {yellow-boxed,

not yellow-boxed} pairs. The basis of this class partitioning was to determine if the second segment of the pair was to be yellow-boxed or not.

- **Post-contextual Information Dataset:** This data set considered the post EEG activity and was divided into two classes with one class containing {yellow-boxed, yellow-boxed} and {yellow-boxed, not yellow-boxed} pairs. The second class was a representation of {not yellow-boxed, not yellow-boxed} and {not yellow-boxed, yellow-boxed}. Here, the possibility of yellow boxing the first segment of the pair is considered.

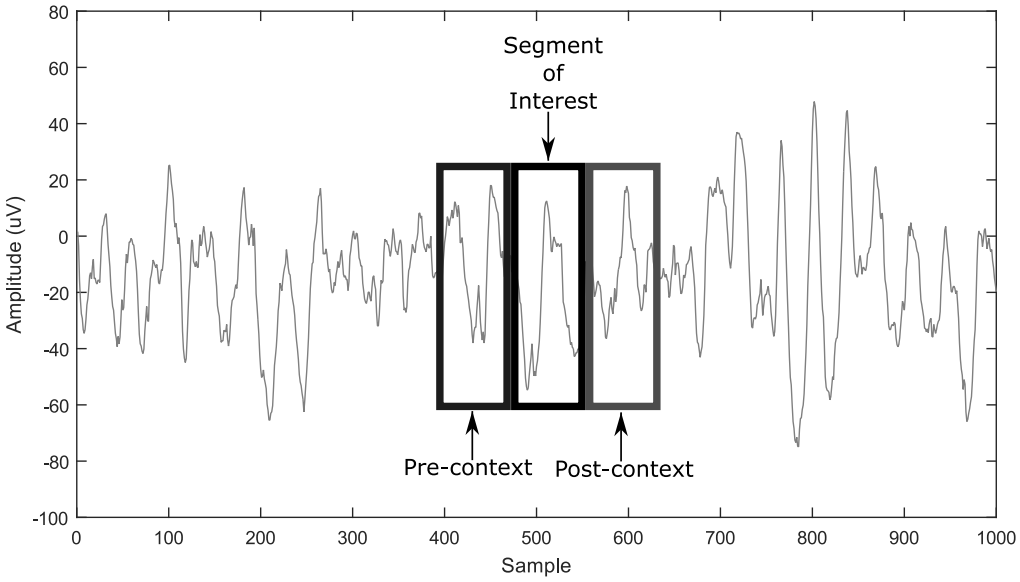


Figure 4.1: Representation of Pre and Post Contextual Signals.

4.3 Feature Selection and Extraction

For the process of detection, the feature characteristics used in classification were also considered and variations of spectral entropy and spectral flatness were used. The exact features used are,

- Spectral entropy of each segment in the pair (Eq. 2.9).
- Ratio of spectral entropy of contextual segment to the sum of the spectral entropies of the two segments.

- Ratio of the spectral entropies of the subband of the contextual segment to the sum of the subband spectral entropies of both segments.
- Spectral flatness of both segments (Eq. 2.12).

This resulted in a feature vector with a dimensionality of 11. The count of feature vectors representative of class yellow-boxed was 444 and representing the class of signals not to be yellow-boxed was 19053. The ratio of not yellow-boxed to yellow-boxed was 42.91. This called for balancing the data set while, if possible taking into consideration the variations that occur in EEG activity from movement of body parts and so on. The variance of the feature vectors in the training set were also maximized by principal component analysis [26].

4.4 Balancing of Data Set

4.4.1 Random Selection

In order to ensure equal representation of data during training, so that the neural network is not saturated to one output or class, the data is balanced to contain the same number of feature vectors from each class. In this class, balancing is done so as to ensure that neural network does not classify all vector in class not yellow-boxed, even though such a decision would result in high specificity but low sensitivity values. One of the methods used for balancing is to choose a feature vector from the majority class (Not yellow-boxed) for every feature vector in the minority class (yellow-boxed) at random. The balanced data set is then used for training and is tested on the entire data set to gauge its performance while generalizing between classes.

4.4.2 Balancing using Clusters

Forming clusters allows the option of choosing feature vectors that are representative of majority class. The k-means clustering method [27] has been used to form clusters of the majority not yellow-boxed class. The procedure for the k-means algorithm are,

- Choose the number of cluster c.
- Choose c different cluster centroids.

- Use a similarity measure (Euclidean distance) to determine similarity of the feature vectors to all of the cluster centers.
- Assign each feature vector to the cluster closest to it.
- Recalculate the cluster centers.
- Repeat until the clusters do not change or number of iterations has been maxed out.

The above algorithm chooses cluster centers at random from the dataset. In order to ensure that best possible centroid are chosen, the k++ means algorithm [28] is used. The k++ means algorithm chooses initial cluster points that are at furthest distance from each other and uses the k-means algorithm to form clusters. The k++ algorithm helps to achieve a faster convergence to the cluster solution.

4.4.3 Near-miss 2 Method

Near-miss 2 method selects majority class examples which have the smallest average distance to the three farthest minority classes examples [29]. Near-miss method is one of the methods proposed by Zhang and Mani as a k-Nearest Neighbor approach to unbalanced data distribution. The choice of distance measures varies. For the detection dataset, Euclidean distance is used as a distance measure.

4.5 Neural Network

4.5.1 Design

The neural network was designed with one hidden layer. Making use of the *(2d+1)* rule of thumb, the hidden unit has 23 units and one output unit. The tanh activation function is used and each unit in the hidden layer and the output unit has a bias. The desired output is +1 to be classified as a segment to be yellow boxed and -1 to be classified as not to be yellow boxed.

4.5.2 Training

The neural network was trained using the training sets obtained from the different balancing methods. The weights were initially determined using ROM and then, the neural network was trained

by back propagation.

4.6 Results

4.6.1 Training Parameters

The weights of the neural network were initialized using values between ±0.5 and chosen using the random optimization method. The weights from the neural network were then updated during back propagation. A learning rate of 10^{-3} and a momentum rate of 0.2 was used, with a maximum iteration limit of 100000.

4.6.2 Training and Performance of Pre-contextual Dataset

While using the pre-contextual dataset, it was observed that the neural network failed to train and differentiate between the classes in the training set when using windows with a length of 100 samples. The trained neural network had a sensitivity of 61% and a specificity of 89.26%. The TSS error plot for training is shown in figure. Thus, the approach using pre-contextual dataset was avoided and the post-contextual dataset was used for further performance testing due better performance estimates.

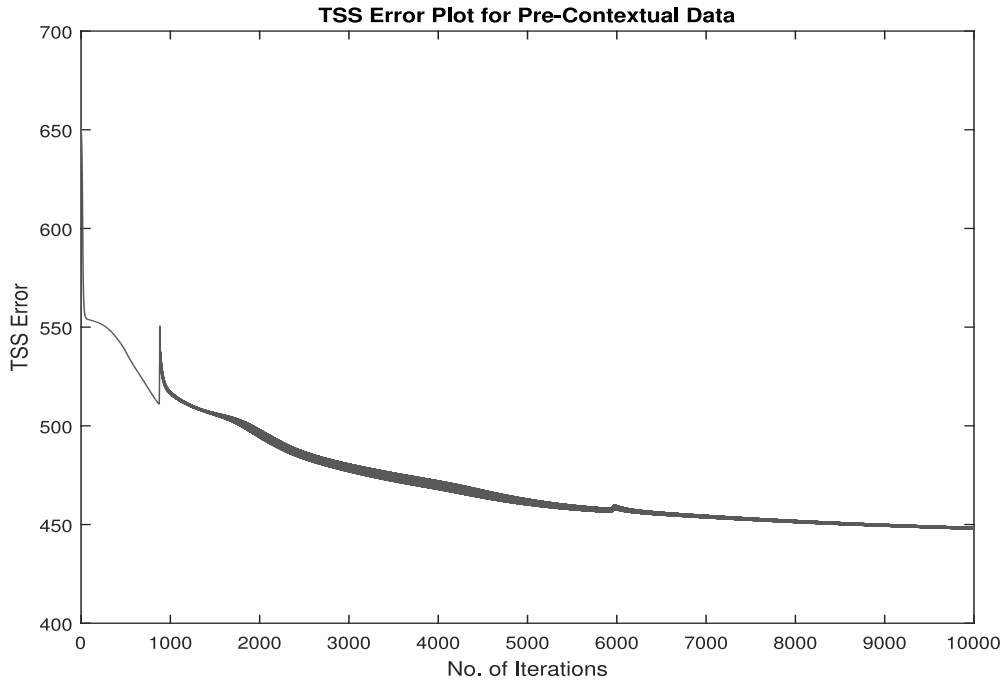


Figure 4.2: Training of Neural Network for Pre-contextual Data.

4.7 Training and Performance of Post-contextual Dataset

4.7.1 Choosing Balancing method of best performance

Before checking the difference in performance due to window length, the neural network performance when trained using the balanced datasets from the different methods used was evaluated. A window length of 60 samples was used and the training of the neural network was visualized by the TSS error plot as shown in Fig. 4.3. This TSS error plot was drawn for each training set obtained from the balancing methods used (random sampling, clustering and near-miss 2 method).

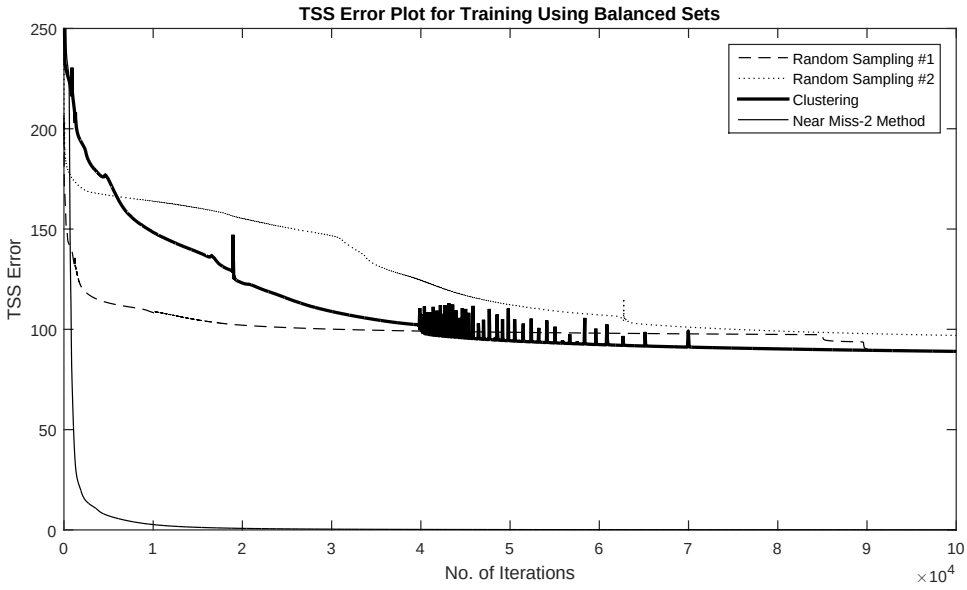


Figure 4.3: TSS Error Plot for Detection Sets Using Different Balancing Methods.

The TSS error plot gives a measure regarding the training of the neural network. The lower the final TSS error is, the better the neural work trains using that particular training set.

From the TSS error plots, it can be observed that the neural network trained on the balanced set obtained from near-miss 2 method had the best learning experience shown by how far the TSS error dropped during learning. In order to estimate the performance, the trained neural network from each balanced data set was tested on the entire Post-contextual Dataset. The performance estimates are tabulated in the Table 4.1

Balancing Method	**Sensitivity(%)**	**Specificity(%)**	**Precision(%)**
Random Sampling #1	91.20	94.26	18.06
Random Sampling #2	95.49	90.40	12.12
Clustering #1	92.10	92.95	15.05
Near-miss 2	99.32	3.30	1.37

Table 4.1: Performance Parameters depending on Balancing Method.

From Table 4.1, it can be inferred that even though, the balanced set from near-miss 2 method did an excellent job while learning, it fails to generalize between the two classes. Due to its low specificity and precision values, the near-miss 2 balancing method was rejected. The

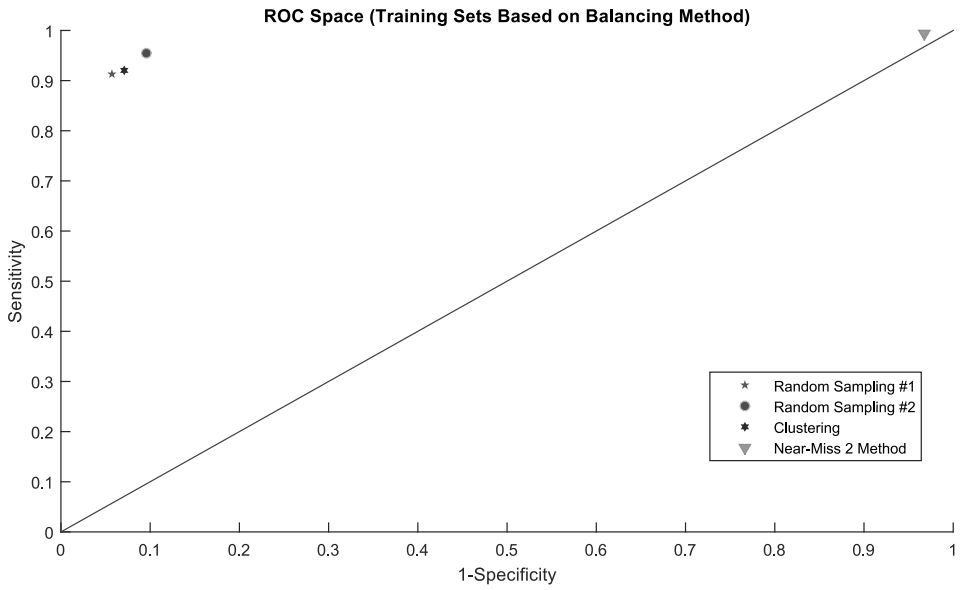


Figure 4.4: ROC Space Using Different Balancing Methods as a Variation Component.

balanced data based on clustering and from random sampling were further verified to check the performance of the ANN. An ROC space plotted in 4.4 gives a better representation of how good the neural networks trained on the datasets obtained from different balancing methods. While the neural networks obtained from random sampling and clustering are plotted towards top left corner of the ROC space, the trained neural network from the Near-Miss 2 method is plotted towards the right corner indicating a dangerously low specificity value and hence, a horrible performance when it comes to the test data.

4.7.2 Cross Validation and Window Length

After choosing the balancing methods, the ANN performance was cross validated for two window lengths: 60 and 100 samples. Both balancing methods did not result in a huge difference in the performance and the performance parameters of the trained neural networks are tabulated in Table 4.2.

Window Length	Trial #	Sensitivity(%)	Specificity(%)	Precision(%)
100	1	96.17	90.83	24.18
	2	97.75	93.92	26.80
	3	96.40	94.33	28.38
	4	95.72	94.26	28.00
	5	95.27	92.92	24.26
	6	97.52	93.99	27.42
60	1	91.20	94.26	18.06
	2	95.03	90.86	12.60
	3	94.13	91.84	13.79
	4	96.16	90.10	11.87
	5	94.13	89.25	10.62
	6	97.52	83.34	7.01

Table 4.2: Performance Parameters for Window Lengths.

Although, the sensitivities and specificities are in the 90s, the precision values tell a different story. Due to the high amount of vectors in the not yellow boxed class, misclassification of over 1500 vectors still resulted in a specificity of 90%. Hence, the low precision values can be used to infer that the though the neural networks were unable to differentiate between classes, it still resulted in a huge number of false positives.

4.8 Conclusion

In order to accomplish yellow boxing of potential AEPs using a neural network, it required the creation of a data set that would best represent the parts of the EEG that were or were not yellow boxed. For this purpose, a large data set containing the not yellow boxed portions of the EEG was created. After training neural networks on the training sets obtained from data balancing, it was observed that sensitivity and specificity values were above 90% (with the expeption of the near-miss 2 balancing method). Although these values indicate really good performance, the precision values show that despite such high values, the neural network struggles to be accurate while determining yellow boxes.

The absence of a medical definition of a not yellow boxed portion of the EEG probably contributed to the failure to improve the neural networks ability to pick out yellow boxes. Also, the use of clustering for balancing did not result in a huge difference compared to those balanced data sets obtained randomly when used for training. A medically defined set of not yellow boxed signals

containing normal EEG activity and EEG artifacts would help to prepare an efficient feature set that could be used for training.

Chapter 5

Overall Performance

5.1 Method

The final objective is to implement the neural networks obtained from the classification and detection training procedures and use them to detect ET's (interictal seizure activity) in patients. From the previous two chapters, two different neural networks have been obtained.

- **Detection Neural Network:** This neural network decides if a segment of the signal, covered by the sliding window is to be yellow boxed or not (Potential AEP detection).
- **Classification Neural Network:** This neural network classifies the yellow boxed segment into an AEP or a non-AEP.

For this purpose, a window is slid across the EEG montage and each window is first checked by the detection neural network of being guilty of a suspect AEP and if yes, is then checked by the classification neural network as to whether it is an AEP or not.

The output of such a method is the marking of AEPs in the EEG and not a diagnosis of epilepsy. The method was tested only on the montage signals that had annotated potential AEP's in them. This was done so as to gauge the performance of the method on the set of signals of which its entire training and testing data was derived from, in order to train the neural networks.

5.2 Design Considerations

During the process of designing this method, the use of a sliding window gave rise to three design parameters that needed to be determined. These parameters would decide how the signal would be processed for the process of detection and classification. These parameters are

- **Window Size:** It has been explained and analyzed from the data that the AEP's tend to be of varying time periods (number of samples). A window size should be selected such that it is big enough to extract features related to epileptic activity. Hence, the window size L that is slid across the signal is of importance.

- **Overlapping Samples** The window is to be slid across the signal while making sure that it does not just divide the signal into X number of segments. This is done as segment of the signal window at the n^{th} could contain information that could display epileptic characteristics with the part of the signal right outside the window. Thus, the number of overlapping samples O is chosen such that $O < 50\%$ of L.

- **Adjacency Limit** Adjacent segments are windowed together. The limit on the number of samples (C) allows for concatenating consecutive potential AEP very close together. This is based on the fact that individual interictal seizure activity does not occur one after the other. It also allows the yellow box segment to grow beyond the window size L.

5.3 Overall Performance in detecting AEPs

In order to evaluate the performance, the neural network with the best performance parameters were chosen from the trained neural networks. In order to obtain a rough estimate of the specificity, the number of negatives per montage signal is assumed to be

$$N = \frac{No.\,of\,Samples\,in\,the\,Signal}{Window\,Size\,(L)} - No.\,of\,Positives \tag{5.1}$$

The value of N is calculated for each montage signal that is used for testing and is summed up to get the total number of negatives. The number of positives was decided to be equal to the number of AEP's (obtained from the classification dataset).

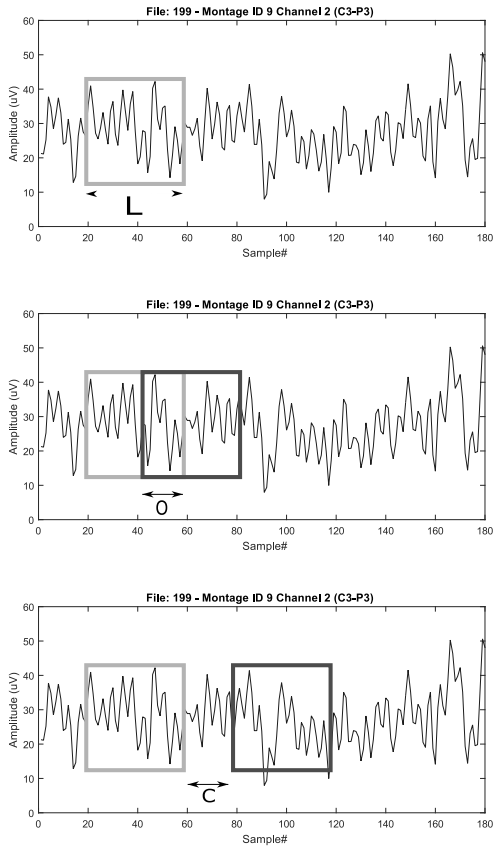


Figure 5.1: Visualization of Window Size, Overlap and Concatenation Sample Limit.

In order to calculate performance parameters, True Positives are the number of AEP's detected correctly and False Positives are the number of falsely detected AEP's. The number of AEP's missed will constitute the False Negatives. To obtain the True Negative count, the following procedure is followed.

- The number of samples(A) in all the detected AEP's during testing is determined.
- The total number of samples (B) in the signals used for testing are determined.
- The value of $\frac{B-A}{Window\ Size}$ gives the number of True Negatives. The specificity values calculated from this is at best an approximation.

Since, the specificity is an approximation, another scoring mechanism was determined. This mechanism, called "Overboxing Score", determines the percentage of the average ratio of sample length of annotated AEPs in the data set to the detected AEP hits while testing. Hence, a score of X indicates that, on an average the annotated length provided by the neurologists is X percentage of the detected length during the automated process.Hence, this score gives an idea about the average overboxing that occurs during detction.

For detection, the neural networks from both window sizes (60 and 100 samples) were chosen to gauge the effect of window size on performance. The results are tabulated in Table 5.2. The adjacency limit for the tabulated results was set at 60 samples.

Window Size	**ANN # ANN #**	**Sensitivity(%) Sensitivity(%)**	**Specificity(%) Specificity(%)**	**Precision(%) Precision(%)**	**Hits (True Positives)**	**Overboxing Score(%)**
60	1	66.29	92.94	21.61	59	23.38
	2	56.18	91.07	16.67	50	26.69
	3	39.33	95.07	20.11	35	25.45
	4	39.33	94.74	15.84	35	22.86
	5	62.92	94.75	15.47	56	19.93
	6	38.20	94.01	18.89	34	26.73
100	1	33.71	97.20	24.19	30	28.86
	2	30.34	95.21	21.43	27	26.84
	3	23.60	94.41	15.33	21	26.79
	4	13.48	96.74	10.90	12	25.57
	5	65.17	98.80	17.96	58	21.81
	6	52.81	96.80	22.47	47	31.20

Table 5.2: Performance Parameters for Window Size=60 and 100.

From the parameter estimates, it was inferred that the performance was much better for a window size of 60 samples when comparing the sensitivity values. The specificity and precision values

had low variation. Comparing the Overboxing Scores also led to the conclusion, that the detection algorithm tended to overbox the AEP segments that are defined as a hit. Thus, for the next step of evaluation, a window size of 60 was used but the adjacency limit between two consecutive segments was varied. The results (sensitivity and precision) are shown in the Table 5.3

Overlap Samples	**Sensitivity(%)**	**Precision(%)**	**Hits (True Positives**	**Overboxing Score**
50	66.29	21.61	59	23.38
60	66.29	21.38	59	23.38
70	66.29	21.77	59	22.96
90	64.04	24.26	57	23.88
100	64.04	23.65	57	23.88
150	39.33	20.83	35	30.12

Table 5.3: Performance for different Adjacency Limits (Using ANN #5-Window Size 60)

From the results, it can be noted that the sensitivity of the method drops as the adjacency limit becomes large. This is probably due to the fact the AEP's usually do not last that long and have a very short sample length(time period).

Although the sensitivities are in the range of 60 with specificities in the high 90s, visual inspection showed that the result was in part due to "over boxing" of signals as can be noted from the values of the overboxing score. There were cases where long segments of the signal were classified as an AEP to detect an ET of around 50-80 samples. One such instance is shown in Fig. 5.2.

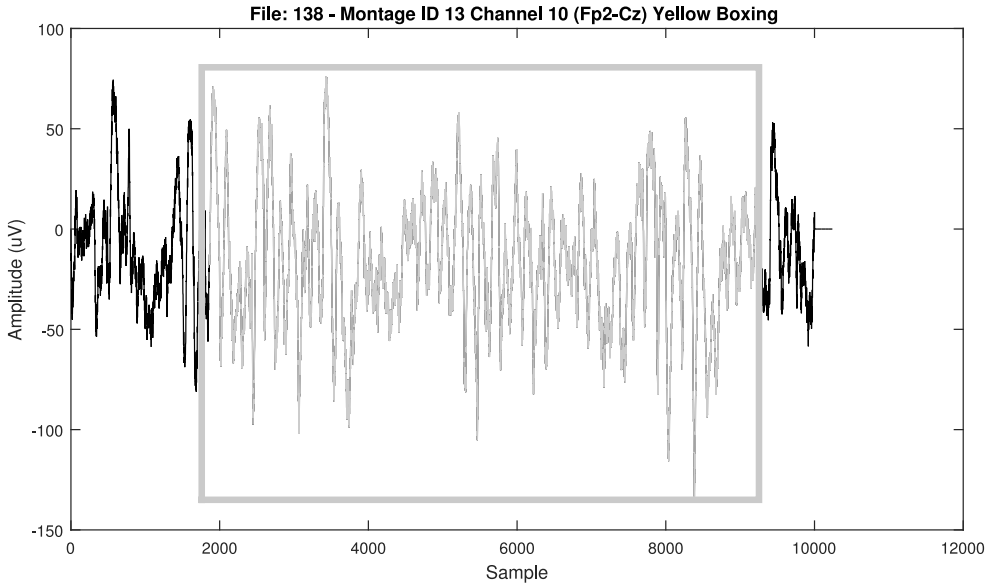
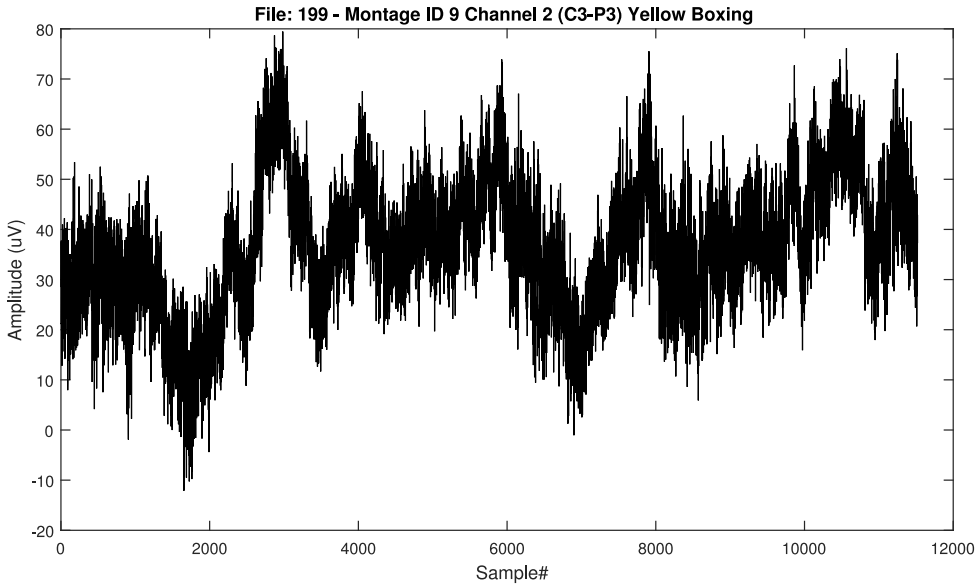
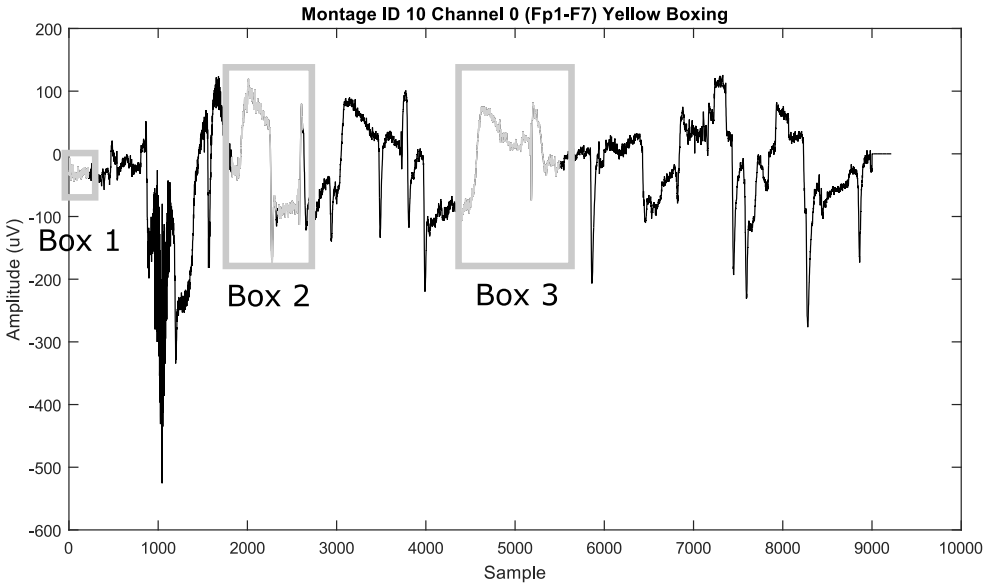


Figure 5.2: An example of Over-Boxing.



This signal does not contain any AEP artifacts and the neural network was able to define all segments as Non-AEP activity.

Figure 5.3: An example of Perfect Non-Detection.



The above signal contains one AEP artifact (Box 2) but also detected two non-AEP artifacts as AEPs (Box 1 and Box 3).

Figure 5.4: An example of Detection of AEP.

5.4 Conclusion

From Table. 5.2, it can be inferred that there is not a large variance in the precision values across the sliding window lengths of 60 and 100. Despite this, a look at the sensitivity values show that a window length of 60 was capable of picking out more AEPs than a window of length 100. But even if the sensitivity values are high, the specificity values determined were at best an approximation of the ANN ability to pick out AEPs and hence, was not enough to reach a conclusion regarding the detector performance.

For this purpose, the proposition of the overboxing score gives a much better estimate by comparing the length of expert defined yellow boxed AEPs and detected AEPs. While comparing the length, it gave an idea about how much a detected AEP is overboxed. Combining this with the number of AEPs detected (True Positives) gives a much better idea about the performance.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

6.1.1 Classification of Yellow Boxes

The development of a feature set independent of the duration of the yellow boxes allowed for their classification, free of the duration. This allowed the classification neural network to train to identify AEP signals regardless of how long the segment was. With the classification network reaching sensitivities and specificities above 90%, the neural network managed to effectively differentiate between AEPs and non-AEPs with a few false positives and negatives. There was also an improvement in the classifier performance with the increase in the training set size. While increasing the values of k in k-fold cross validation, the reduction in the size of each fold, led to an increase in the training set used. This increase in training set size let to an increase in performance of the neural network. Thus, training the classifier by feeding it more examples of each class would help improve its performance.

6.1.2 Yellow Box Detection

With the classifier pretrained to ignore the duration of potential AEPs, this allowed yellow boxing while allowing the box to grow accordingly. Instead of using a window of variable length, a fixed length window was used and adjacent boxes were concatenated if the duration between occurence was close to each other. With AEPs having durations of order of milliseconds, this meant

any threshold for combining two adjacent boxes would be a very small number of samples. At best this threshold was decided upon while considering the sample range of potential AEPs. By allowing the box to grow, this fed potential AEPs of varying length to the classifier.

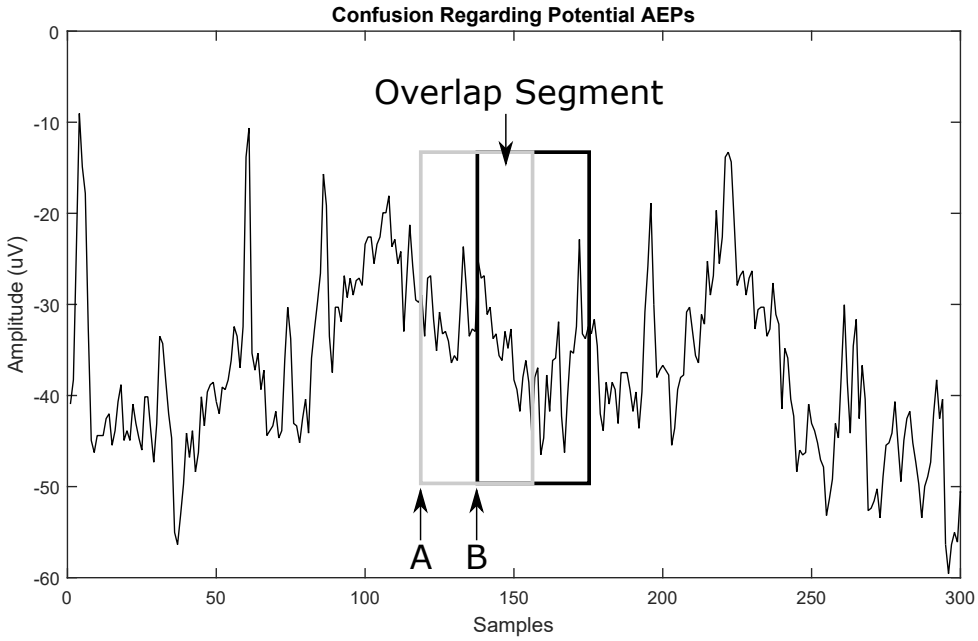
Due to the absence of a medical definition of a not yellow boxed EEG signal, this led to the usage of any portion of the EEG that was not yellow boxed as representative signal of a not yellow boxed EEG. Despite the huge size and substitute selection process for good members representative of not yellow boxed portions, the neural network trained well enough but had low values of precision. Eventhough generalization was achieved to a certain extent, the absence of a defining feature to seperate potential AEPs from the rest of the EEG resulted in the low precision rates despite high sensitivity and specificity.

6.1.3 Overall Detector Performance

Thus, an automated EEG detector was designed by making use of two neural networks work concurrently. The detection neural network decides if a portion of the EEG signal inside a sliding window is to be yellow boxed or not. Conditions were provided to allow for piecing together yellow boxes close to each other. After yellow boxing, the yellow boxed segments are then checked and classified as AEP or non-AEP. The end result was the boxing of AEP segments.

The documented results show that the detector was able to detect atleast 60% of the AEPs marked by the experts. But despite this performance, a detailed study of the results show that it also chose portions of the EEG determined as normal EEG activity or non-AEPs. In addition to this, the detector tended to overbox AEPs as shown by instances were expert marked AEPs were detected with excess samples.

The yellow boxing of certain normal EEG activity segments was attributed to the set up of the data set used for detection. Although the detection data set was generated keeping in mind the length of the window, the different segments of signals that arise from a sliding window was not considered. For example, a window placed at sample A would be defined as portion not to yellow boxed. But on moving the window over to sample B, while maintaining overlap, could result in features that point it to being a potential AEP. An illustration is shown in Fig. 6.1.



A and B are sample numbers indicating the position of the sliding window. The overlap segment indicated could result in features that indicate yellow boxing at position B but not at A.

Figure 6.1: Illustration of Detection Challenge Due to Overlap.

6.2 Future Work

The features utilised were limited to spectral entropy, flatness and skewness. Therefore, future work would involve the study of other spectral characterisitics and how the addition and omission of different features affects the ability of the neural network to differentiate between AEPs and non-AEPs. With clinical subbands shown to display epileptic activity medically, the variation of features with respect to one or more subbands could also be considered while defining features. With the study revolving around feature extraction from the power spectral density, features obtained from other higher order spectra such as bispectra (Fourier transform of third order cumulant) and polyspectra (higher order cumulants) could be considered along with features extracted from wavelets. A detailed study of smaller window lengths can be performed to analyse performance changes in the detector.

The challenge faced during detection, an absence of a definition of unboxed EEG signals, could be remedied and used as part of the data set used for detection. The addition of well defined

yellow boxes and unboxed signals could help deal with the precision issue of detection.

The detector could also be designed to learn by verification with experts. Testing the detector on unseen data and verifying the results with the help of an expert could improve the performance while allowing the detector to learn with each exposure to a new form of AEP.

Bibliography

[1] MedicineNet. Epilepsy, 2016. URL `http://www.medicinenet.com/script/main/art.asp?articlekey=3285`.

[2] Jing Zhou. *A study of automatic detection and classification of EEG epileptiform transients*. PhD thesis, Clemson University, 2014.

[3] K.P. Indiradevi, Elizabeth Elias, P.S. Sathidevi, S. Dinesh Nayak, and K. Radhakrishnan. A multi-level wavelet approach for automatic detection of epileptic spikes in the electroencephalogram. *Computers in Biology and Medicine*, 38(7):805816, 2008. doi: 10.1016/j.compbiomed.2008.04.010.

[4] Bruce Scheepers, Peter Clough, and Chris Pickles. The misdiagnosis of epilepsy: findings of a population study. *Seizure*, 7(5):403–406, 1998.

[5] Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

[6] Zafer Iscan, Zmray Dokur, and Tamer Demiralp. Classification of electroencephalogram signals with combined time and frequency features. *Expert Systems with Applications*, 38(8):1049910505, 2011. doi: 10.1016/j.eswa.2011.02.110.

[7] Jaime F. Delgado Saa and Miguel Sotaquirá Gutierrez. EEG signal classification using power spectral features and linear discriminant analysis: A brain computer interface application. In *Eighth Latin American and Caribbean Conference for Engineering and Technology*, pages 1–7, 2010.

[8] Alexandros T. Tzallas, Vaggelis P. Oikonomou, and Dimitrios I. Fotiadis. Epileptic spike detection using a kalman filter based approach. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 501–504. IEEE, 2006.

[9] Hamid R. Mohseni, A. Maghsoudi, and Mohammad B. Shamsollahi. Seizure detection in EEG signals: a comparison of different approaches. *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006. doi: 10.1109/iembs.2006.260931.

[10] Anusha K.S., Mathew T. Mathews, and Subha D. Puthankattil. Classification of normal and epileptic EEG signal using time and frequency domain features through artificial neural network. *2012 International Conference on Advances in Computing and Communications*, 2012. doi: 10.1109/icacc.2012.21.

[11] Ahmad Mirzaei, Ahmad Ayatollahi, Parisa Gifani, and Leili Salehi. Eeg analysis based on wavelet-spectral entropy for epileptic seizures detection. *2010 3rd International Conference on Biomedical Engineering and Informatics*, 2010. doi: 10.1109/bmei.2010.5639894.

[12] Dragoljub Gajic, Zeljko Djurovic, Jovan Gligorijevic, Stefano Di Gennaro, and Ivana Savic-Gajic. Detection of epileptiform activity in EEG signals based on time-frequency and non-linear analysis. *Frontiers in Computational Neuroscience*, 9, 2015. doi: 10.3389/fncom.2015.00038.

[13] Anne-Claire Conneau and Slim Essid. Assessment of new spectral features for EEG-based emotion recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4698–4702. IEEE, 2014.

[14] Inan Güler and Elif Derya Übeyli. Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. *Journal of Neuroscience Methods*, 148(2):113–121, 2005.

[15] Ernst Niedermeyer and F. H. Lopes da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.

[16] Trans Cranial Technologies. *10-20 Positioning Manual*. Trans Cranial Technologies, 2012.

[17] Douglas K. Lindner. *Introduction to Signals and Systems*. McGraw-Hill, 1999.

[18] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[19] George E.P. Box and Gwilym M Jenkins. *Time Series Analysis: Forecasting and Control, Revised Edition*. Holden-Day, 1976.

[20] Otis M. Solomon Jr. Power spectral density computations using welch's method. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1991.

[21] T. Inouye, K. Shinosaki, H. Sakamoto, S. Toi, S. Ukai, A. Iyama, Y. Katsuda, and M. Hirano. Quantification of EEG irregularity by use of the entropy of the power spectrum. *Electroencephalography and Clinical Neurophysiology*, 79(3):204–210, 1991.

[22] Robert J Schalkoff. *Artificial Neural Networks*. McGraw-Hill Higher Education, 1997.

[23] Hojjat Adeli, Samanwoy Ghosh-Dastidar, and Nahid Dadmehr. A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy. *IEEE Transactions on Biomedical Engineering*, 54(2):205–211, 2007.

[24] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145, 1995.

[25] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[26] Ian Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2002.

[27] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[28] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[29] Ronaldo C. Prati, Gusotavo E. A. P. A. Batista, and Maria Carolina Monard. Data mining with imbalanced class distributions: concepts and methods. In *Indian International Conference on Artificial Intelligence 2009*, pages 359–376, 2009.

[30] G. Widman, T. Schreiber, B. Rehberg, A. Hoeft, and C. E. Elger. Quantification of depth of anesthesia by nonlinear time series analysis of brain electrical activity. *Physical Review E Phys. Rev. E*, 62(4):48984903, Jan 2000. doi: 10.1103/physreve.62.4898.

[31] Pari Jahankhani, Vassilis Kodogiannis, and Kenneth Revett. EEG signal classification using wavelet feature extraction and neural networks. *IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)*, 2006. doi: 10.1109/jva.2006.17.

[32] Marzia De Lucia, Juan Fritschy, Peter Dayan, and David S. Holder. A novel method for automated classification of epileptiform activity in the human electroencephalogram-based on independent component analysis. *Medical and Biological Engineering and Computing*, 46(3): 263272, Nov 2007. doi: 10.1007/s11517-007-0289-4.

[33] V. Srinivasan, C. Eswaran, and N. Sriraam. Artificial neural network based epileptic detection using time-domain and frequency-domain features. *Journal of Medical Systems*, 29(6):647660, 2005. doi: 10.1007/s10916-005-6133-1.

[34] Alexandros T., Markos G., Dimitrios G., Evaggelos C., Loukas Astrakas, Spiros Konitsiotis, and Margaret Tzaphlidou. Automated epileptic seizure detection methods: a review study. *Epilepsy - Histological, Electroencephalographic and Psychological Aspects*, 2012. doi: 10.5772/31597.

[35] Jing Wang and Guanghua Xu. Some highlights on epileptic EEG processing. *Recent Patents on Biomedical Engineeringe BIOMENG*, 2(1):4857, Jan 2009. doi: 10.2174/1874764710902010048.

[36] Jonathan J. Halford, Robert J. Schalkoff, Jing Zhou, Selim R. Benbadis, William O. Tatum, Robert P. Turner, Saurabh R. Sinha, Nathan B. Fountain, Amir Arain, Paul B. Pritchard, et al. Standardized database development for EEG epileptiform transient detection: EEGnet scoring system and machine learning analysis. *Journal of Neuroscience Methods*, 212(2):308–316, 2013.

[37] Jing Zhou, Robert J. Schalkoff, Brian C. Dean, and Jonathan J. Halford. Morphology-based wavelet features and multiple mother wavelet strategy for spike classification in EEG signals. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3959–3962. IEEE, 2012.

[38] Michal Teplan. Fundamentals of EEG measurement. *Measurement Science Review*, 2(2):1–11, 2002.

[39] Aihua Zhang, Bin Yang, and Ling Huang. Feature extraction of EEG signals using power spectral entropy. In *2008 International Conference on Biomedical Engineering and Informatics*, volume 2, pages 435–439. IEEE, 2008.

[40] Anthony M. Murro, Don W. King, Joseph R. Smith, Brian B. Gallagher, Herman F. Flanigin, and Kimford Meador. Computerized seizure detection of complex partial seizures. *Electroencephalography and Clinical Neurophysiology*, 79(4):330333, 1991. doi: 10.1016/0013-4694(91) 90128-q.

[41] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2001.

[42] Manoj Kumar Mukul, Rajkishore Prasad, and Fumitoshi Matsuno. Relative spectral power and power spectral density changes in motor rhythm for movement imagination. In *ICCAS-SICE, 2009*, pages 1611–1616. IEEE, 2009.