12-2016

Studying the Use of Hidden Markov Models in the Detection and Classification of EEG Epileptiform Transients using LPC features

Kartikeya Shrikant Bagalore
Clemson University

Studying the Use of Hidden Markov Models in the Detection and Classification of EEG Epileptiform Transients using LPC features

---

A Thesis
Presented to
the Graduate School of
Clemson University

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

---

by
Kartikeya Shrikant Bagalore
December 2016

Accepted by:
Dr. Robert J Schalkoff, Committee Chair
Dr. Carl Baum
Dr. Richard Groff

ABSTRACT

The process of identifying the presence of an AEP (Abnormal Epileptiform Paroxysmal) in a subject's EEG, normally done by neurologist experts, is a particularly long one and involves considerable financial expenses. This research aims to pave an automatic method of detecting and classifying streams of EEGs as to whether or not it has any AEPs present in it. This is a two step process, where step 1 is the classification problem and step 2 is the detection problem.

There are many different activities on the EEGs, and the classification task helps to identify which of these activities are AEPs. So, this task involves training 2 HMMs to classify all given artifacts into 2 classes, AEP or NonAEP. LPC features extracted from the spike have been used to train the HMMs.

The detection task is to find out the presence of ETs(Epileptiform Transients) from a patient's EEG. For detection, two HMMs have been trained on examples taken from two classes, the ETs and the Non-ETs. The ETs class is all the Yellow Boxed annotations provided to us by the experts. The Non-ET class data has been formed by taking into consideration all the data which has not been marked as an ET. In this task, LPC features extracted from the spike and the contextual information has seen to provide good results.

For validation of the system, a cascaded structure of four HMMs is formed. The first two HMMs are for detection and the next two classify the detected ETs. Test EEG signals, having both AEPs and NonAEPs are passed through this system, and the AEPs are marked and identified. The results have been compared to the annotations marked by experts.

ACKNOWLEDGMENTS

First and foremost, I express my deepest gratitude towards Dr. Robert J. Schalkoff for his continuous support and guidance throughout my research. I am also thankful to him and Dr. Jonathan Halford for providing us with a complete set of data and problem specifications from a real world. I want to show my gratitude towards Dr. Groff and Dr. Baum for readily being a part of my advisory committee. Everyone in the research group also deserve to be thanked for the help and encouragement that they offered.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Problem of Epilepsy

In medical literature, epilepsy has been defined very precisely and succinctly as follows : *Epilepsy is a disorder of the brain characterized by an enduring predisposition to generate epileptic seizures and by the neurobiologic, cognitive, psychological, and social consequences of this condition. The definition of epilepsy requires the occurrence of at least one epileptic seizure.* [6]
An epileptic seizure is a transient occurrence of signs and/or symptoms due to abnormal excessive or synchronous neuronal activity in the brain. Epilepsy is the 4^{th} most common neurological disease in the world. Only migraine, stroke and Alzheimer's syndrome have a higher rate of occurrence. To be able to grasp how high the prevalence of epilepsy actually is, some numbers are presented below from the Epilepsy Foundation. [16]

- 65 million : The total number of people suffering from epilepsy around the globe.
- 3 million : The total number of people suffering from epilepsy in the United States.
- 1 in 26 : The odds of the total number of people who will suffer from epilepsy at some point of time in their lives.
- 0.04% to 0.1% : The number of people on earth suffering from an active seizure at any time.
- 150,000 : New cases of epilepsy diagnosed in United States every year.
- 6 out of 10 : Cases of epilepsy where the root cause of epilepsy is unknown.

The most common tool used for diagnosis of epilepsy is the routine scalp Electroencephalogram (rsEEG). In the EEG of any subject, the presence of epileptiform activity confirms the diagnosis of epilepsy. This epileptiform activity resembles a spike or a sharp wave discharge. It is also called as an epileptiform transient.(ET) [1][12]. These spikes are 20-70 ms in duration, and are followed by sharp waves of 70-200 ms duration. Some of the ET's have a longer duration,around 150-350 ms which are together called as spike-and-slow-wave-complex. The diagnosis of epilepsy is done by an Electroencephalographer EEGer by visual inspection of a subject's EEG . Though typical EEGs are only 10-20 minutes in duration, there are some studies showing improved results for recording longer than 20 minutes This makes the diagnosis notoriously time consuming. Also, there is considerable disagreement among experts reading the same record. [24][10]

There are a wide variety of morphologies of ET's, all of which indicate the presence of Epilepsy. Also, these waveforms may resemble other activities (wicket spikes, exaggerated alpha activity, small sharp spikes and other sleep related activities) and other artifacts (extra-cerebral potentials arising from eye blink, eye movement, gritting of teeth, muscle etc.) These are some of the reasons why there is not a single system which automatically detects ET's with a high accuracy and other performance measures like sensitivity and specificity. [10]

1.2 Previous Research

Since the 1970s, considerable efforts have been made to design a system capable of automatically detecting ETs from raw EEG. Initially, template matching was one of the popular methods for most detection and classification tasks. An EEG segment and a model ET waveform would be compared, and their cross-correlation would be calculated and scored. Depending on a certain threshold, the classification decision is made.

Some examples of discriminative classifiers used in EEG classification are Bayes classifiers and Support vector machines.(SVM). [14] Also, logistic regression and naive Bayes have been used for the purpose of detecting artifacts from the EEG. Many other examples of classifiers include Linear Discriminant Analysis, Binary Decision Trees, Nearest Neighbor techniques, etc. The very popular neural net or Multilayer Perceptron has also been studied to a great extent for the purpose of this task. But, all the above mentioned classifiers are sensitive to the various problems inherent in EEG classification tasks like severely unbalanced data, huge variations in training data, and the time

course of the data.[14]

The above mentioned classifiers have all been used with a huge variety of features such as amplitude values of EEG, Band Powers(BP)[17], Power Spectral Density(PSD)[4], Hjorth Parameters, Autoregressive(AR), Hjorth and Adaptive Autoregressive Parameters(AAR)[4]. The latter 2 features are used in time domain neural networks, as it is very important to exploit time information when performing any kind of EEG analysis.

HMMs are the most popular and heavily researched tool in the area of speech recognition. More recently,they have also been applied to different tasks in EEG classification and analysis. These methods apply HMMs to some features which are changing in time. These time changing features are usually extracted by an AR model, or by other signal processing techniques. Some examples are frequency means, peak frequency detectors, Hjorth Parameters, etc. There has been research which conclude that HMMs are very effective classifiers when used for detecting changes in non-stationary signals.

1.3 Overview of Research

The main aim of this research is to be able to automatically detect ETs from the given data, and to classify these transients as AEP or Non-AEP. The data files contain yellow boxes marked by experts, which are called as annotations, and all of these are ETs. The data files are the EEG recordings of 200 patients, of which 99 files contain some type of ETs in them, and the other files do not. The first task is to find out the location of these ETs in all the files, called as the detection task. The second task is the classification task, where the ETs are appropriately classified into 2 classes, AEP or Non-AEP.

An HMM-based recognition system, analogous to a speech recognition system has been built, and the performance has been compared for different configurations of the model. To incorporate the time information of the signal, Linear Predictive Coding (LPC) features have been extracted and used in training of the HMMs. The methodology for classification is exactly similar, where LPC features from the annotated yellow boxes are used to train the HMM. For each class, be it in detection or classification, an HMM is trained on the samples belonging to that class. The unclassified samples from the test data are then fit to both HMMs, and a score is obtained. The unclassified sample is assigned the class of the HMM with the higher score.

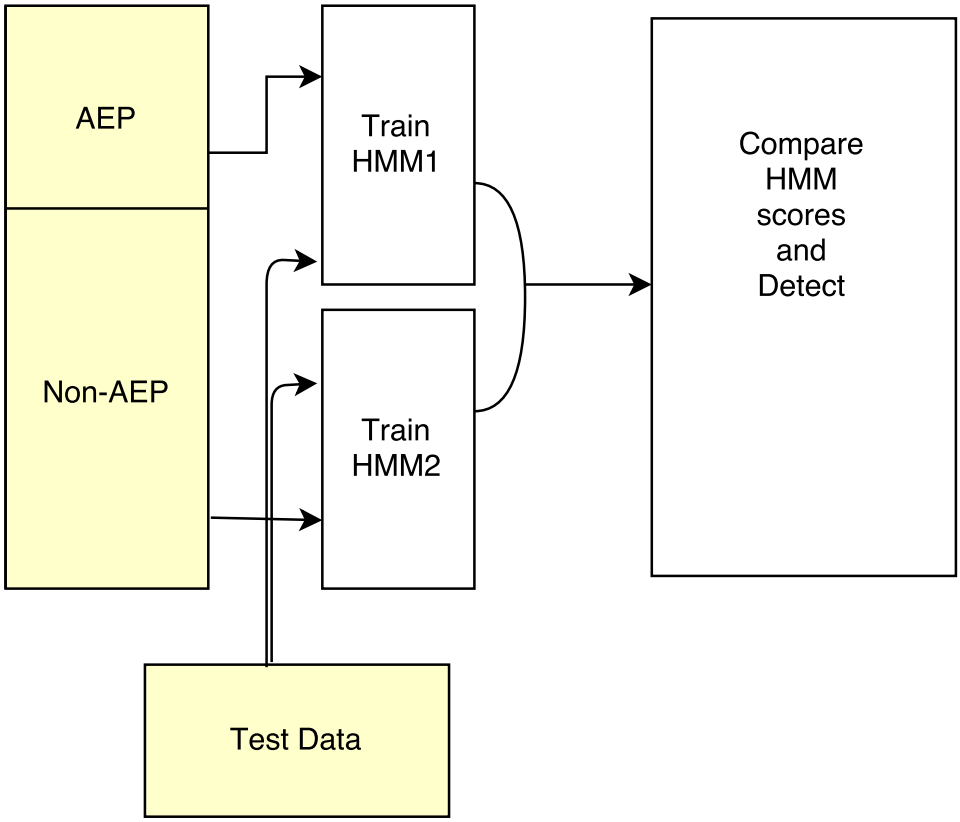

Figure 1.1: An Overview of HMM-based EEG Detection

Chapter 2

Background Information and Data Acquisition

2.1 Electroencephalography(EEG)

Electroencephalography(EEG) is the most reliable and commonly used tool for the prediction of epilepsy. Apart from epilepsy, EEG recordings are also used to identify other medical conditions related to the functioning of brain like coma, sleep disorders, brain death, etc. In the past, even tumors and strokes used to be identified by the EEG, till the arrival of more sophisticated tools. An EEG records the electrical activity in the brain, and is used as a monitoring device to view the voltage fluctuations resulting from ionic current within the neurons of the brain[20]. An important property of the EEG is that it is non-invasive. This means that there is no pain during the EEG recording, and can be easily obtained. But, the human brain is enclosed in a covering of skull, cerebrospinal fluid, scalp and hair. So, only the high potential electrical activity, generated by a large number of neurons discharging simultaneously can be recorded, which is generally the case during an epileptic seizure. Otherwise, the EEG activities are usually very small, having an amplitude in microvolts(uV) and the frequencies of interest are always under 30 Hertz(Hz).

The International 10-20 system is the most popular system for EEG measurements. Small metal discs called as electrodes are placed on the scalp at positions determined by this system. This method implies taking measurements between different points on the scalp, which cover 10% to 20%

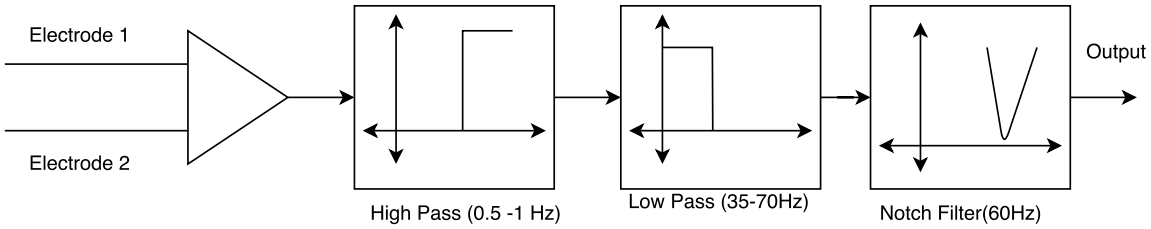

Figure 2.1: EEG Signal from Electrodes

of these distances, and hence the name. The EEG machine uses differential amplifiers to produce "channels", where the electrodes are the input to the differential amplifier. The output of the differential amplifier is the fed to 3 filters, a high pass, a low pass and a notch filter. The high pass filter (0.5Hz -1Hz) and the low pass (35-70Hz) are used to remove any high or low electrical activity outside our range of interest. The notch filter(60Hz) removes noise from ambient electrical power. As all EEG signals lie within a range of 100Hz, a typical sampling rate used is 256Hz - 512 Hz, which satisfies the Nyquist criterion. The output is then stored in computer systems after digitizing them. [24]

The 10-20 system has been developed to ensure standardized reproducibility for comparison of studies between different subjects and studies. Totally, 21 electrodes are attached to different sites on the scalp as shown in Figure 2.2. The letters associated with these sites are given in the Table 2.1 below. Furthermore, even numbers (2,4,6,8) mean electrodes connected to the right hemisphere of the scalp,and the odd numbers (1,3,5,7) are connected to the left hemisphere. The letter 'z' is used to an electrode placed along the midline of the scalp.

Letter	Lobe
F	Frontal
T	Temporal
C	Central
P	Parietal
O	Occipital
A	Earlobes
Pg	Nasopharyngeal
Fp	Frontal Polar

Table 2.1: Names and Letters of Sites on the Scalp

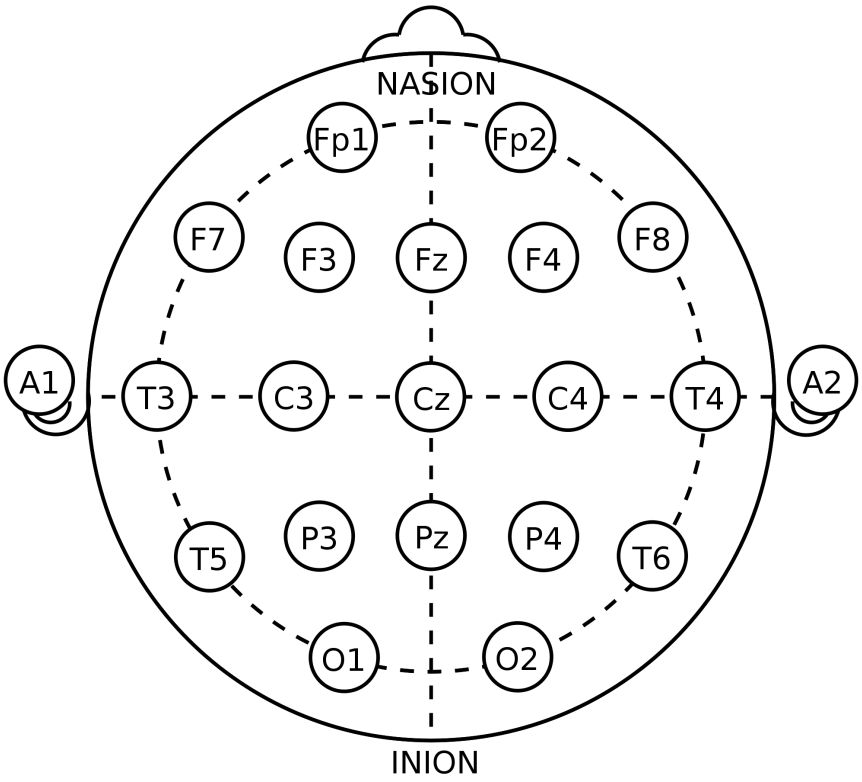

Figure 2.2: The International 10-20 System

In the 10-20 system, the manner in which each pair of electrode is connected is called as a montage. So, each montage will use one of the below mentioned three standard recording derivations:

- Common reference: An electrode, usually A1 or A2, the ear electrode, is chosen as a reference electrode. Then, each amplifier gives the difference between a scalp electrode and the reference electrode. This reference electrode should be common to all the electrodes.

- Average reference: The average of all the electrodes is calculated and then passed to a high value resistor. Then, the output of the amplifier will be the difference of a scalp electrode and this average value.

- Bipolar: The channel is simply the difference between any 2 electrodes, and the montage consists of differential data of a series of electrode pairs. These electrodes are usually in straight lines across the head, or transversally across the head.[20]

2.2 Details of the Data

The dataset has been obtained with the aim of creating a standardized EEG database to train, test and compare the performance of different automated spike detection systems. This database adheres to the International 10-20 system. A 100 minute long EEG recording has been formed by concatenating 200, 30 second epochs in the European Data Format(.edf) [11]. Each 30 second file belongs to a separate patient. This file was sent out to 18 experts who are board-certified by the American Board of Clinical Neurophysiology. The final dataset has been formed over 3 phases :

- Phase 1: The experts were asked to mark one yellow box over over any epileptiform discharges throughout the complete 100 minute recording. They were allowed to use any montage, but were limited to only one channel per discharge.
- Phase 2: All the annotations were collected and clustered using the "yellow box algorithm" described by Waters, C. in this paper [23]. Any events marked by at least 2 experts, were included in this clustering. A month later, the experts were asked to classify each event (previously marked annotations) on the Likert scale shown in Table 2.2.
- Phase 3: Phase 3 was to make the data more reliable. There were some inconsistencies in the data over the previous 2 phases. Sometimes, when asked to take a second look at the events, the experts would make a contradicting assignment. Also, some experts completely missed out a few events, but confirmed them as AEPs when asked to take a second look. These are type of inconsistencies which were extracted among the 2 phases of the data, and were sent back to the experts for reconsideration after 3 months.

After these 3 phases, each expert was given a score based on consistency, and a list of the best 7 experts (most consistent 7) was formed. This is the data that has been in this research. The Likert scale is shown in the Table 2.2 :

Raw Data

Along the 200 edf files and the scorer data, a separate file containing the following columns has been obtained. [24] Columns :

- Annotation ID : A unique ID for each annotation (yellow box).

Score	Meaning
201	Definitely not an AEP
202	Not an AEP
203	Unsure
204	AEP
205	Definitely an AEP

Table 2.2: Likert scale for Classification of Epileptic Transients

- Data Set ID : Name of the file to which this event belongs. (1-200)
- Start second : The start time of each event in the 100 minute long recording.
- End second : The end time of each event in the 100 minute long recording.
- Montage ID : The montage used by the experts.
- Channel number : The channel on which the event was viewed. A file describing the formation of each montage has also been obtained, and any yellow box annotation can be reproduced for viewing by using the Montage ID and the channel number.
- Score : All columns contain the score given by each expert, one column per expert.

2.3 Processed MATLAB Struct

The raw data was converted to a MATLAB struct for easy access during further processing and calculations. All the signals were loaded into MATLAB using the 'edfread.m' MATLAB function. The `records` variable stored the recordings, and the `hdr` variable stored the header information about the particular file. The electrode information and sampling frequency was obtained from this variable. All the files were resampled to a common frequency of 256 Hz, to provide normalization. The MATLAB struct, called as "`overall_db`" has the following fields :

- Fields 1 - 6: The first 6 fields are the same as the columns in the raw file described in the above list 2.2.
- Fields 7-13 : The next seven fields are the scores by the best 7 experts.
- Field 14 : `Complete_Montage_Signal` is the field containing the complete signal using the particular montage and channel as by the experts themselves during the annotation task.

- Field 15 : This field "`spike`" contains each annotation individually.
- Field 16,17 : The 2 fields `prespike` and `postspike` contain 50 samples before and after the annotated yellow box. This data was obtained to make any calculations involving the use of contextual information, if necessary.
- Field 18 : `sample rate` This field contains the sampling frequency.
- Field 19 : The field `inferred_groundtruth` states the true label of the annotation. As described earlier, each of the best 7 experts have assigned a score from 201-205 to all the annotations. By taking a vote amongst the score given by the experts, the true class is decided as the class appearing maximum number of times. So, for each annotation, the true class is the "mode" of all the 7 scores given by the best 7 experts.

From this processed struct, it is very easy to query, plot and run algorithms on the annotated signals, the background signal, and also enables efficient feature extraction. In the following figures, a few examples of the annotated spikes from all classes are shown. Also shown is the background contextual information, so that it helps to visualize how an expert would classify these signals just by visual inspection.

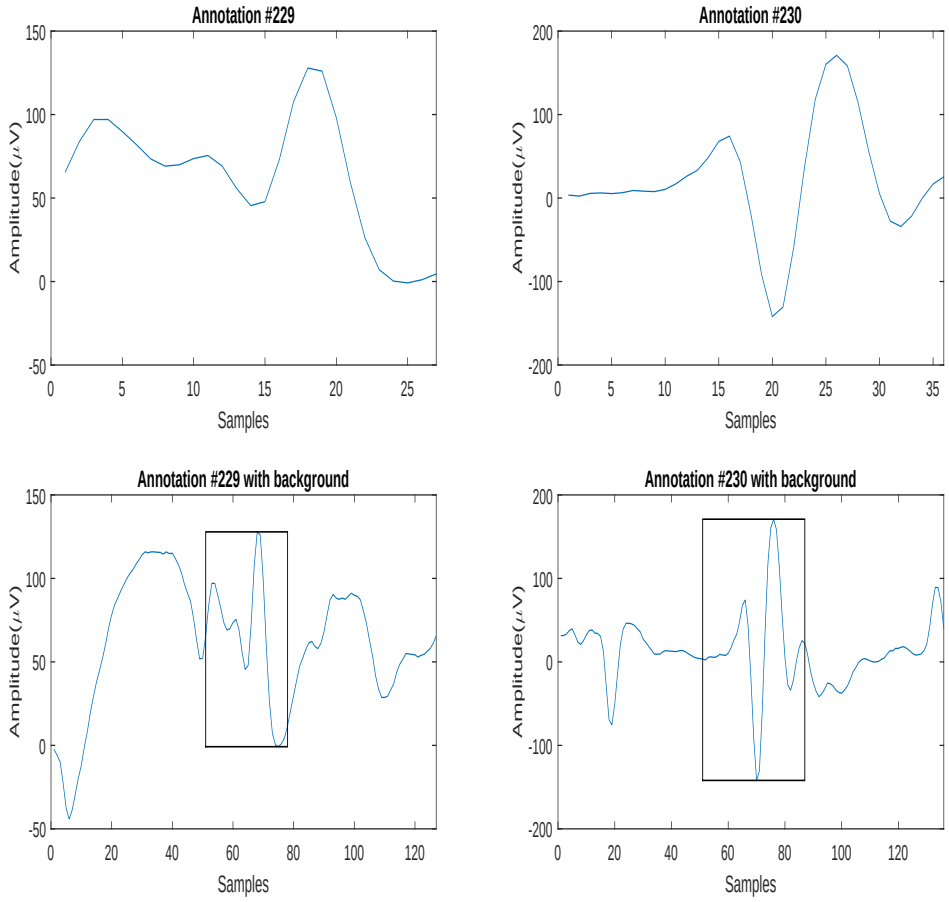

Figure 2.3: An example showing 2 Annotated Yellow Boxes of class 205. The images on the bottom show the annotations with some background contextual information, similar to what an expert would see on an EEG monitor. All 7 experts agree that this YB is definitely an AEP.

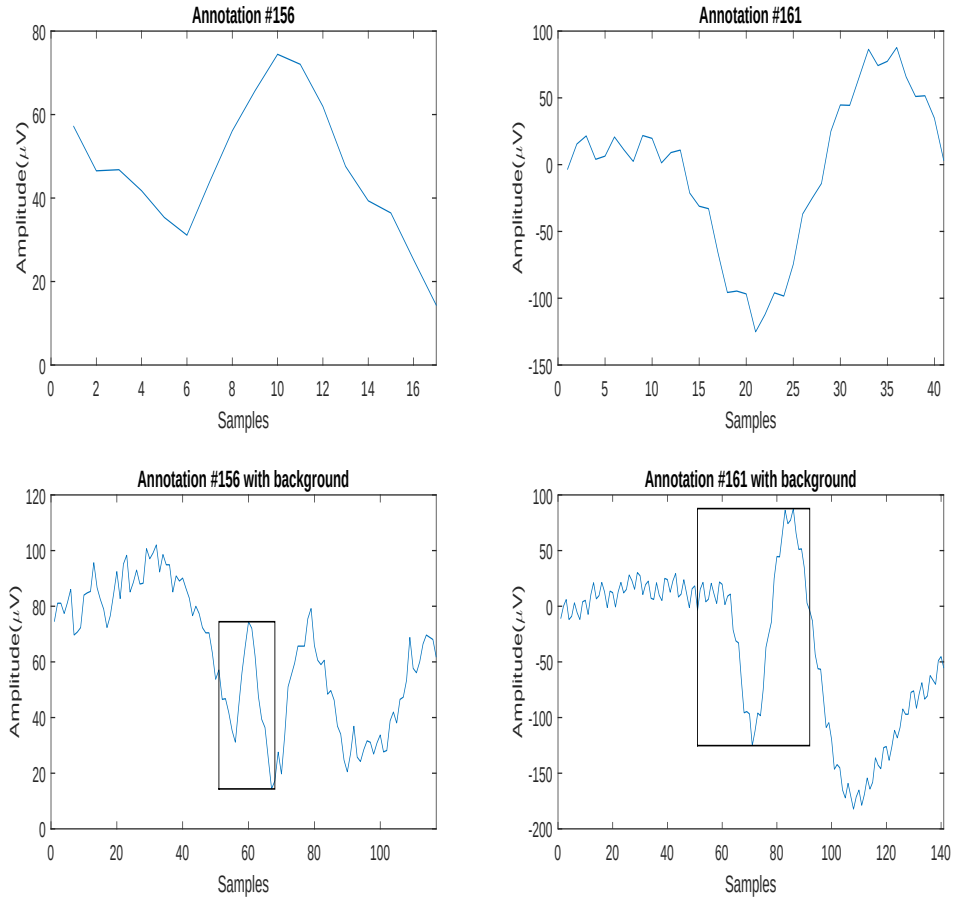

Figure 2.4: An example showing 2 Annotated Yellow Boxes of class 204. The images on the bottom show the annotations with some background contextual information, similar to what an expert would see on an EEG monitor. These 2 YBs have been marked as "Likely to be an AEP" by 5 out of 7 experts.

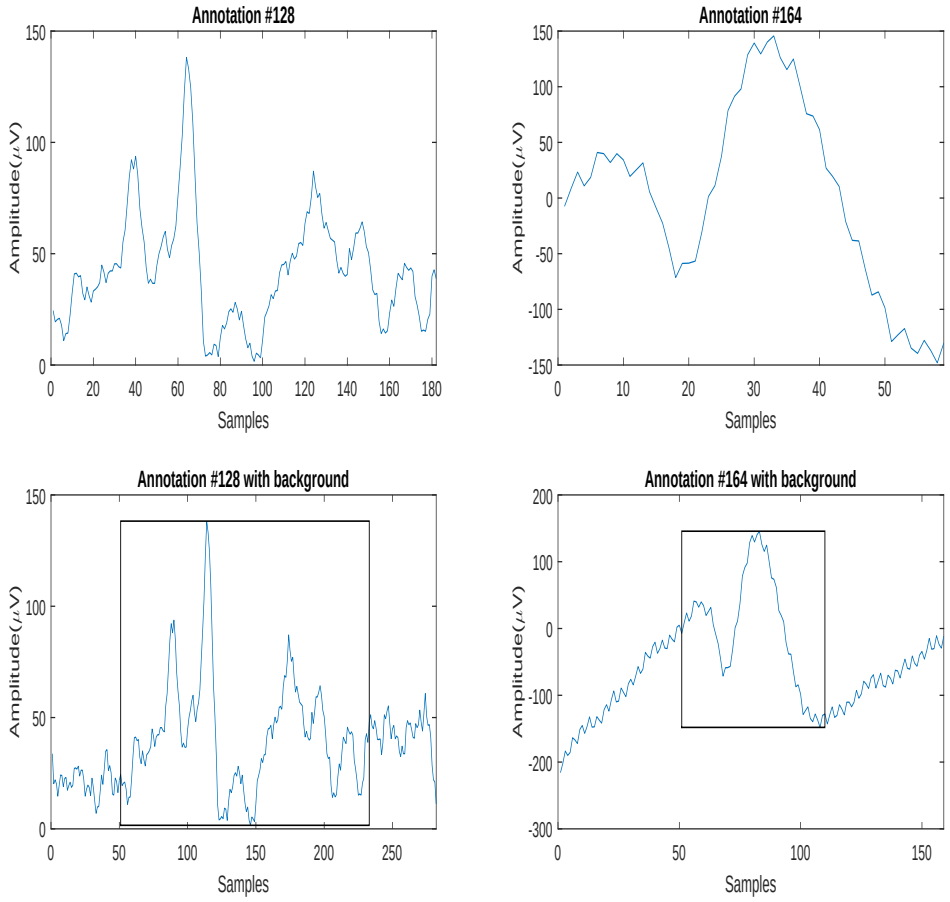

Figure 2.5: An example showing 2 Annotated Yellow Boxes of class 203. The images on the bottom show the annotations with some background contextual information, similar to what an expert would see on an EEG monitor. 3 out of 7 experts said they were "not sure" about the class of these YBs, which means they could go either way.

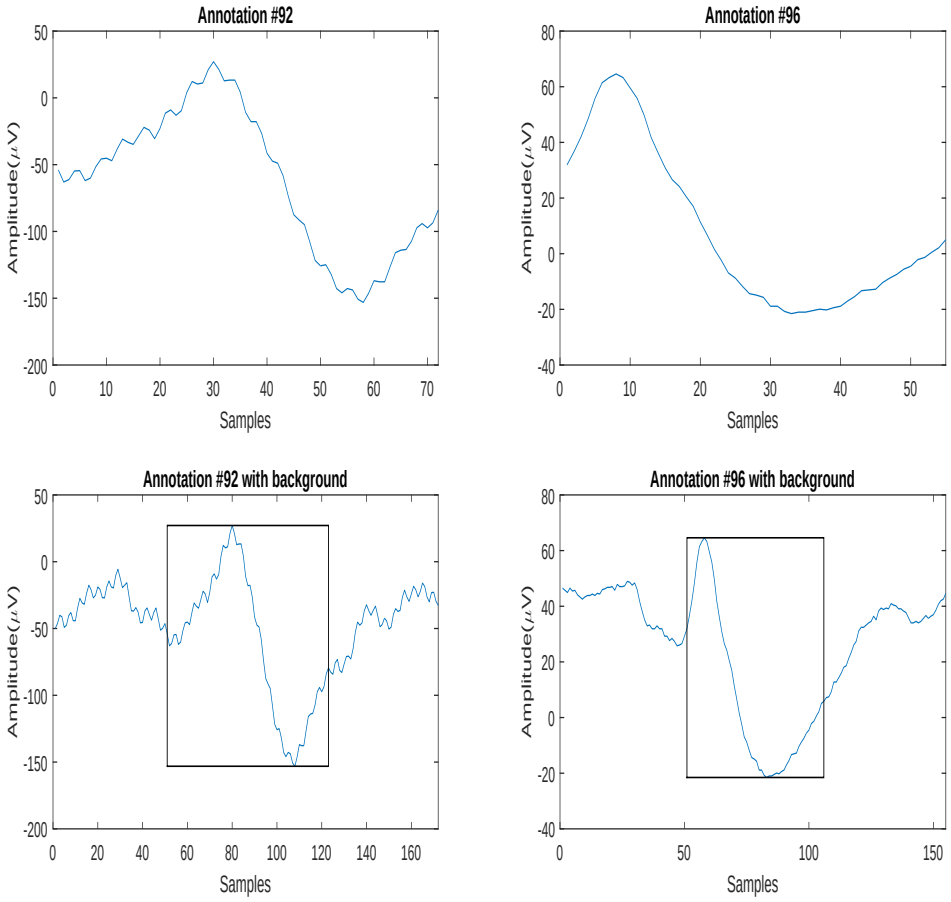

Figure 2.6: 2 Examples of Annotated Yellow Boxes (YBs) of class 202. The images on the bottom show the annotations with some background contextual information, similar to what an expert would see on an EEG monitor. These 2 YBs have been marked as "Not an AEP" by 6 out of 7 experts.

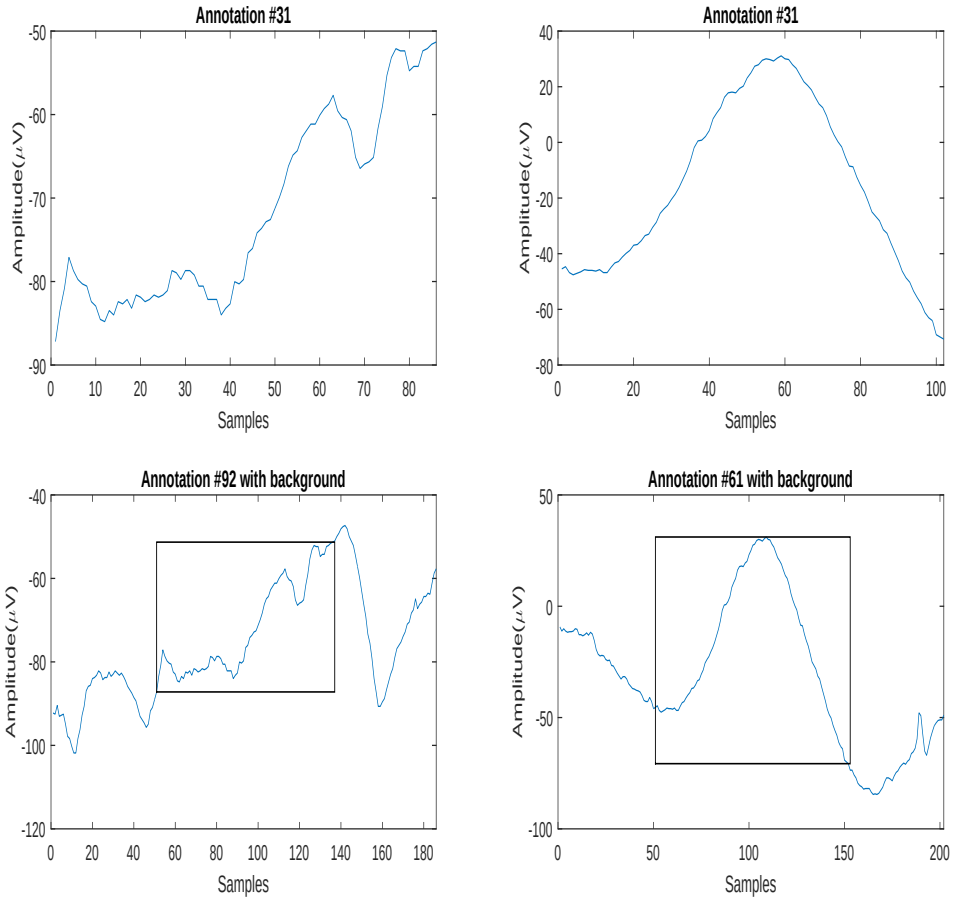

Figure 2.7: 2 Examples of Annotated Yellow Boxes (YBs) of class 201. The images on the bottom show the annotations with some background contextual information, similar to what an expert would see on an EEG monitor. These 2 YBs have been marked as "Definitely Not an AEP" by 6 out of 7 experts.

Chapter 3

Theoretical Outline for LPC and HMM

3.1 Linear Predictive Coding (LPC)

The theory of linear predictive coding (LPC) has been studied and applied to the analysis of time series, and the earliest papers are Atal [2] and Makhoul [15]. Some favorable properties of LPC which make it popular in speech recognition systems, seismic analysis, and other time series analysis are :

- LPC is analytically tractable. It is mathematically precise, and the computation is simple and straight forward to implement.
- In recognition systems, it has been shown that LPC performs better than other recognizers like filters-banks spectral analysis approach [19].

In the field of EEG analysis, it is important that the EEG signal be easily described in a simple mathematical form, which would preserve all the characteristics of the signal. Considering the above factors, an LPC model called as the 'all pole model' is the most popular model, and will be briefly discussed. The book by Rabiner [19] has summarizes the mathematical details.

The basic idea of Linear Prediction can be stated as: The current sample can be closely approximated as a linear combination of past samples. Mathematically, we can state

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p) \tag{3.1}$$

where 'p' can be any number from 1 to N-1 where N is the length of the current sample 's' and is called the order of the predictor. The above equation is converted to an equality, if we add an excitation term $Gu(n)$,

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \tag{3.2}$$

If we convert the above to the z-domain,

$$S(z) = \sum_{i=1}^{p} a_i z^{-i} S(z) + GU(z) \tag{3.3}$$

and the transfer function can then be found as

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \tag{3.4}$$

Linear prediction analysis calculates a set of coefficients $\{a_k\}$ for each window in the incoming EEG signal which can best model the signal. The spectral characteristics of the EEG vary a lot with time, and so we divide the EEG signal into small windows. The best set of coefficients minimizes the mean-squared prediction error over this short segment of the EEG signal. The linear combination of the past p samples is

$$\tilde{s}(n) = \sum_{i=1}^{p} a_i s(n-i) \tag{3.5}$$

and the error can then be defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{i=1}^{p} a_i s(n-i) \tag{3.6}$$

This leads us to the error transfer function

$$A(z) = \frac{E(z)}{S(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{3.7}$$

This can be represented graphically as shown in Fig

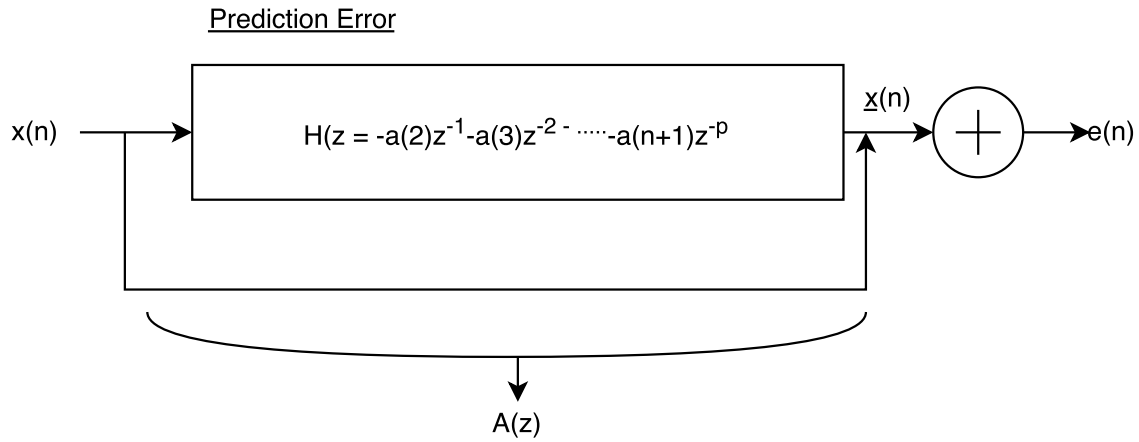

Figure 3.1: The Transfer Function for Prediction Error

When we have a sample s(n) actually generated by a linear set of coefficients, we can see that the excitation term Gu(n) is actually the prediction error e(n). Now, for calculating the coefficients over a short time, the short term signal and error segments at time n are set up as

$$s_n(m) = s(n + m) \tag{3.8}$$

$$e_n(m) = e(n + m) \tag{3.9}$$

The quantity to minimize is the mean squared error E_n given as

$$E_n = \sum_m e_n^2(m) = \sum_m [s(n) - \sum_{k=1}^{p} a_k s(n - k)]^2 \tag{3.10}$$

The minima is found by differentiating E_n with respect to each a_i and setting it equal to zero,

$$\frac{\partial E_n}{\partial a_k} = 0, \quad k = 1, 2, ...p \tag{3.11}$$

and hence

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k) \tag{3.12}$$

The covariance of $s_n(m)$ is given as

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{3.13}$$

So, the minimum mean squared error can be given as

$$\hat{E}_n = \sum_m s_n(m)^2 - \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m)s_n(m-k) \tag{3.14}$$

$$= \phi_n(0,0) - \sum_{k=1}^{p} \hat{a}_k \phi_n(0,k) \tag{3.15}$$

This minimum mean-squared error term (MMSE) can be solved in 2 ways, the covariance method and the autocorrelation method. these methods differ in their way of defining limits on m. The autocorrelation method is described shortly here. In this method, it is assumed that $s_n(m)$ is zero outside $0 \leq m \leq N-1$. [19]

$$s_n(m) = \begin{cases} s(m+n)w(m), & 0 \leq m \leq N-1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.16}$$

which leads to the short term covariance $\phi_n(i,k)$ being

$$\phi_n(i,k) = \sum_{m=0}^{N-1+p} s_n(m-i)s_n(m-k) \quad 1 \leq i \leq p, \quad 0 \leq k \leq p \tag{3.17}$$

or

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \quad 1 \leq i \leq p, \quad 0 \leq k \leq p \tag{3.18}$$

Since the auto-covariance function is symmetric, i.e $r_n(-k) = r_n(k)$,the LPC coefficients can be

expressed in matrix form as :

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \dots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \dots & r_n(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \dots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ \vdots \\ r_n(p) \end{bmatrix}$$

The above $p \mathrm{X} p$ matrix, called as the Yule-Walker equations can be solved using the Levinson-Durbin's algorithm. [13]. The linear predictive coefficients are very efficient models of the EEG samples, as seen in the Fig 3.2 - Fig 3.6

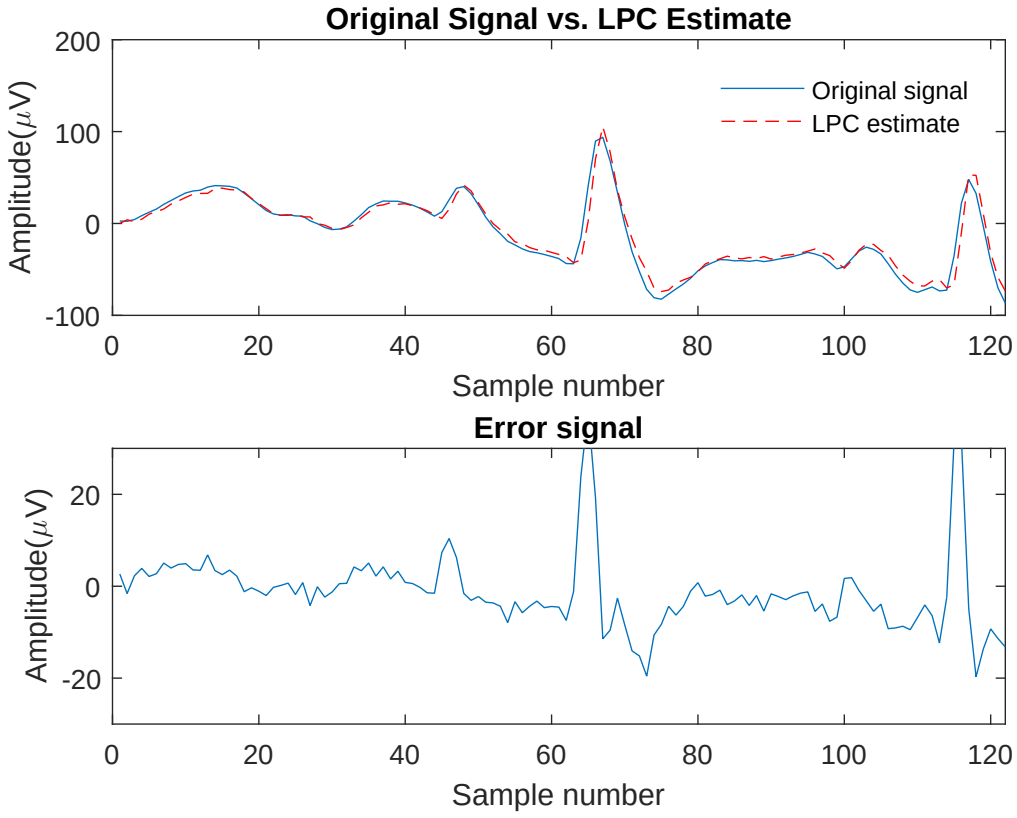

Figure 3.2: Comparison of original signal and the LPC estimate. The error signal is shown in the plot below. The above plot is of an YB annotated as 205

We can see in the error estimates of AEP samples that it is not doing a good job of modeling

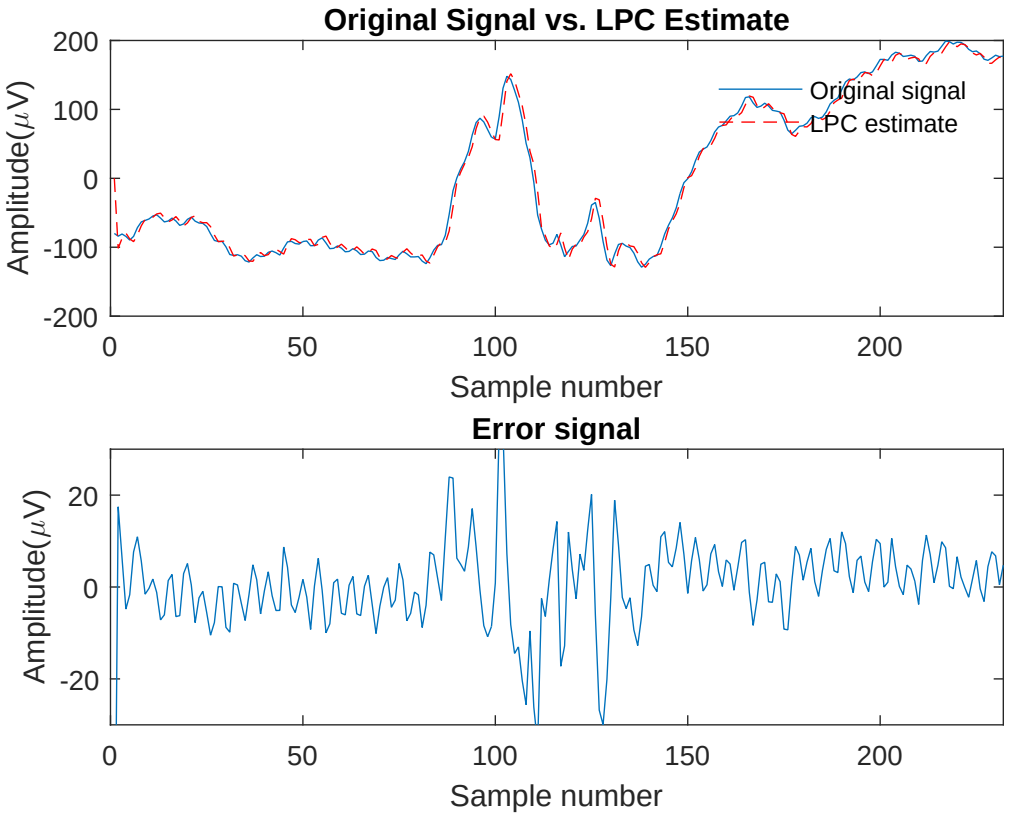

Figure 3.3: Another example of Comparison of original Signal and the LPC estimate. The error signal is shown in the plot below. The above plot is also of a YB annotated as 205

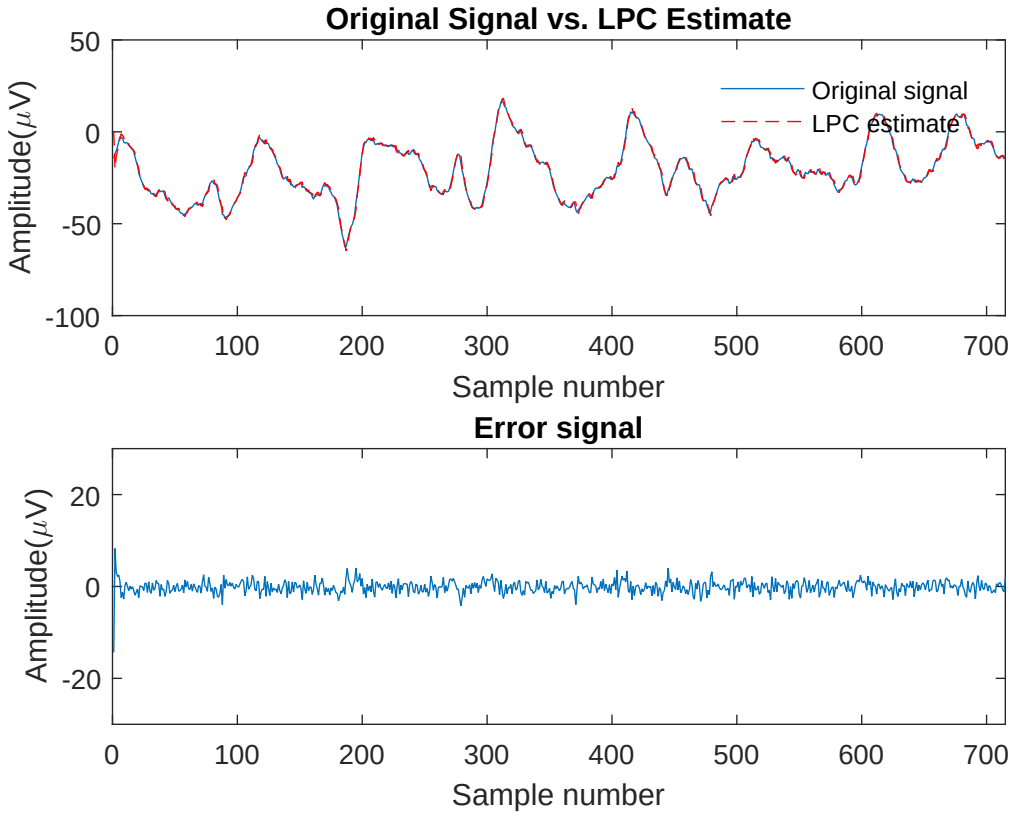

Figure 3.4: An example of Comparison of original Signal and the LPC estimate. The error signal is shown in the plot below. The above plot is of a YB annotated as 201

the signal properly when there are certain spikes or surges in the EEG. The error tends to be very high at these times. This property is utilized in some techniques performing thresholding over the energy of the error signal. Also, HMMs can incorporate this as a separate state, and provide better performance in classification. In the following examples of Non AEP yellow boxes and of non-annotated parts of the EEG, we can see that this behavior is absent.

From Figures 3.4, 3.5 and 3.6, we can see that the LPC estimates does a fairly good job in modeling the original EEG. The error signal has a very low magnitude as compared to the original signal. In some cases, where the signals have some artifacts which are not necessarily epileptic activity, the error signals have a large magnitude. So, this indicates that methods like template matching or thresholding would be prone to give a large number of false positives.

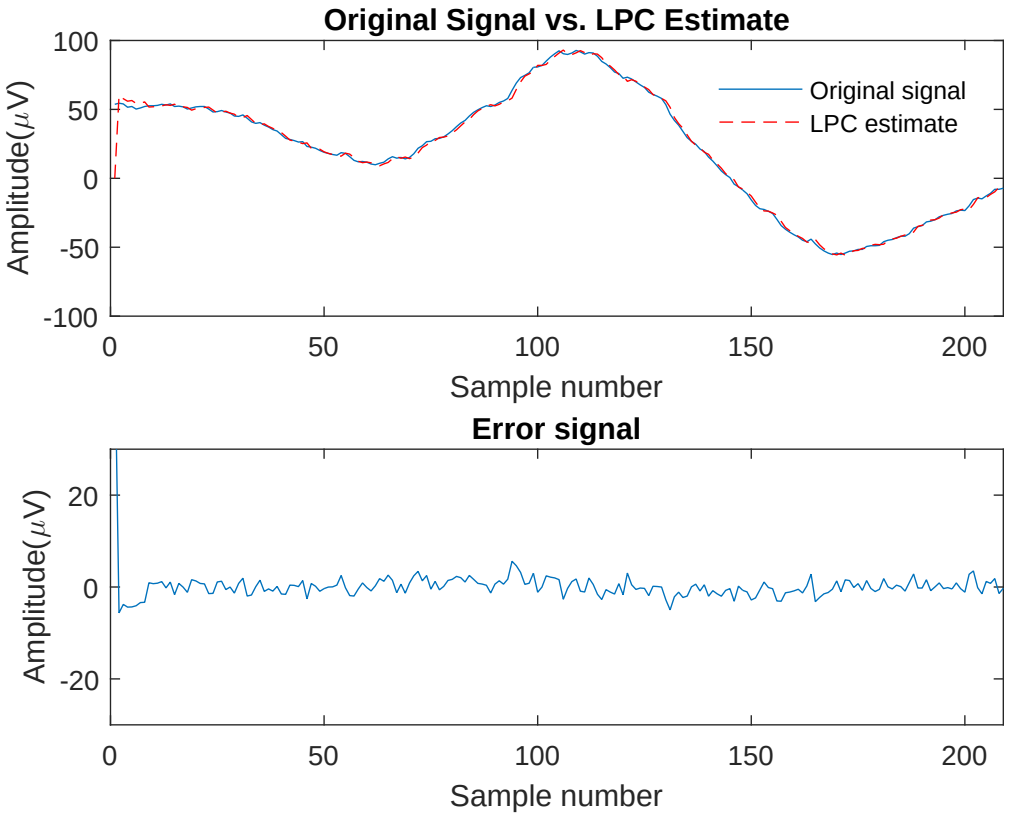

Figure 3.5: Another example of Comparison of original Signal and the LPC estimate. The error signal is shown in the plot below. The above plot is of a YB annotated as 201

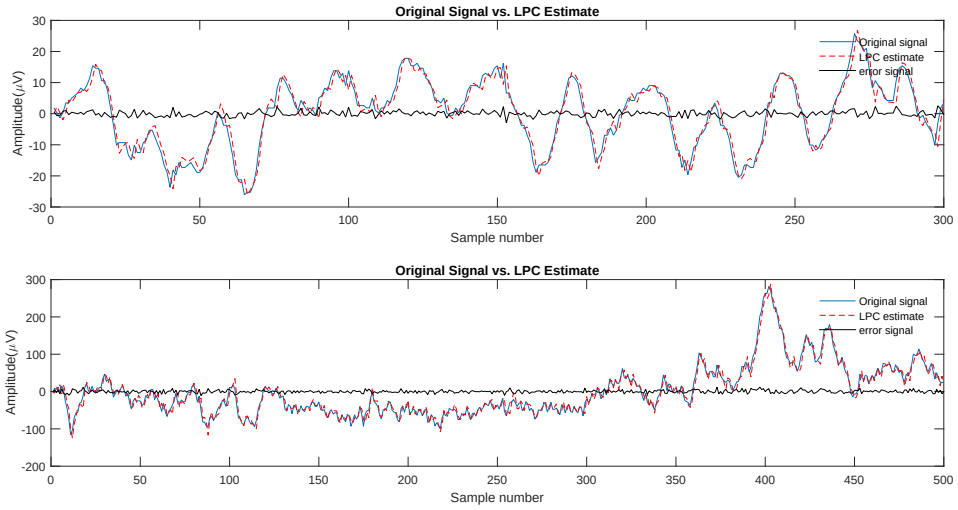

Figure 3.6: 2 examples of Comparison of original Signal and the LPC estimate. The error signal is shown in black. Both the signals are from the unannotated part of the EEG

3.2 Hidden Markov Models (HMM)

Introduced in the late 1960s, Hidden Markov models have been studied and researched extensively over the years. The tutorial by Rabiner [18] is one of the most important papers in the study of HMMs and their applications to Speech Recognition. Hidden Markov models are popular because they are very rich models mathematically, and if used properly, can be used to model many real life scenarios very accurately. Thus, on proper modeling and configuration of an HMM, they give very good results in important applications. In this research, an HMM, using LPC based features, have been applied to the problem of EEG detection and classification. Since the underlying mathematics of an HMM is so important for it's configuration, a short theoretical outline of HMMs has been provided.

Markov Chains

An understanding of Discrete Markov chains in presented, which can then be easily extended to Hidden Markov models. Let us consider an N state system, where each state is denoted as S_1, S_2, S_3....S_N. Now, consider that the system changes state from 1 state to another according a certain probability associated with each state. This change happens after every unit of time (t_1, t_2, t_3....), and the system can simply change it's state back to the same state. To describe this probabilistic

behaviour of a system, shown in Fig3.7, completely, we would require specification of the current state and all the previous states. But, some systems have a special property, called as the Markov property, which restricts this probabilistic description to just the current state and the previous state, given as :

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_j .....] = P[q_t = S_j | q_{t-1} = S_i] \tag{3.19}$$

In simple words, to go from state S_1 to S_2, the system needs probabilities only from states S_1 and

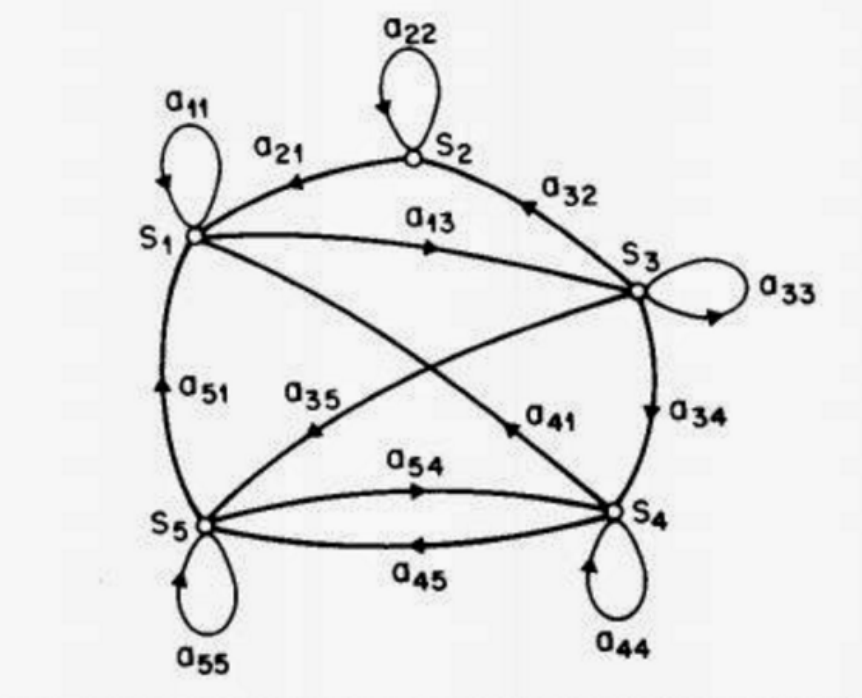

Figure 3.7: A simple 5 state Discrete Markov Chain [18]

S_2, and has complete independence from the probabilistic description of how to reach state S_1 in the first place. The set of probabilities of going from 1 state to another is called as the state transition probabilities (a_{ij}) which obey the standard stochastic constraints

$$a_{ij} \geq 0 \tag{3.20}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.21}$$

If each state corresponds to a physical event (like a phoneme, or the weather condition), then the output of the system is simple the set of states at each instant of time. Such a system is called as an observable Markov model.

Extension to Hidden Markov Models

The observable models are too restrictive to yield any real life applications. So, we introduce the Hidden Markov models here, which are a double embedded stochastic process, where the stochastic process is not observable as an output. The transition from an observable Markov model to an HMM is best explained with an example. [18].

The Fig 3.8 shows a general case of a system which can be modeled as an HMM. Consider it to be an N glass urns in a room, having a large number of colored balls in each urn. Let there be M distinct colors (4 here) of the balls. Now consider the following scenario, where one of the urns is selected by a random stochastic process, like rolling a 3 outcome dice. Once a particular urn is selected, a ball is picked out of the urn by a second random process. The color of the ball is recorded as an observation, and the ball is replaced. Again, a new urn is chosen by the first random process, and a new ball by the second process. This combination of 2 random processes is repeated, and a finite set of observations is obtained from the ball colors : O = {GREEN, RED, BLUE, YELLOW, YELLOW, BLUE ....GREEN}. This set of observations O is then modeled as the output of the given HMM. [[18]]. The simplest model would involve a 3 state HMM, with a state corresponding to the urn, and M color probability (given separately for each urn in fig3.8) associated with each state.

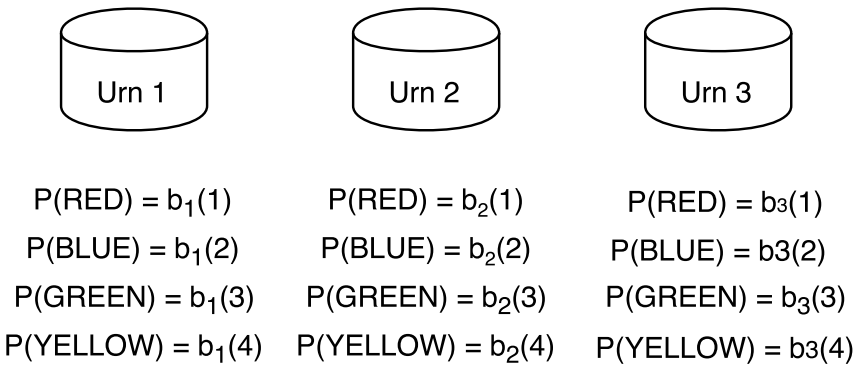

Figure 3.8: A 3 state ball and urn model, showing a general case of an HMM. [18].

Elements of an HMM

To be able to apply an HMM to actual applications, a formal definition of each element of an HMM is required. The HMM is characterized by the following elements:

1. N, the number of stated in a model. The number of states have a physical significance, and there are different arguments in choosing the number which are described later. This N is the hidden part of the HMM. Different types of HMMs are modeled by imposing different constraints on the states. An ergodic model requires a path from a state to every other state. The Baki's model allows transition to states having higher index than the current state, which is what most time series behave like.

2. M, the number of distinct observations per state. These are the observable outputs of the system being modeled. The individual states are denoted by symbols as V = $\{v_1, v_2, ...v_n\}$

3. A = $\{a_{ij}\}$, the state transition probability distribution. A gives us the probability of reaching a state from any other state.

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, \quad j \leq N \tag{3.22}$$

   For all left right models, we have a $a_{ij} = 0$ for transitions to any state having lower index than the current state.

4. B, the observation symbol probability distribution in state $j, B = b_j(k)$ where

$$b_j(k) = P[v_k \, at \, t | q_t = S_j], \quad 1 \leq j \leq N \tag{3.23}$$

$$1 \leq k \leq M \tag{3.24}$$

5. π, which is the initial state distribution where

   $\pi_i = P[q_1 = S_j]$

Thus, for a complete description of an HMM, we would require the 2 model parameters (N and M), the specification of the three probability measures A, B and π. The compact notation for describing an HMM is

$$\lambda = (A, B, \pi) \tag{3.25}$$

3.2.1 The Three Basic Problems of HMM

Jack Ferguson [7] has given a description of 3 problems for the analysis of an HMM, and suggested techniques and solutions for the same. The solutions are mathematically intense, and can be referred to at [18] or [7].

- Scoring Problem: Given an observation sequence O = O_1,O_1,O_2,O_3 ...O_T, and a model for our HMM as $\lambda = (A, B, \pi)$, how to calculate $P(O|\lambda)$. This is the probability of observing a given sequence, when we have already have designed and estimated the parameters of an HMM. A useful viewpoint is to look at it as scoring how well a given model fits the observation we have at hand.

- Optimizing Problem: Given an observation sequence O = O_1,O_1,O_2,O_3 ...O_T, we try to find the sequence of states Q =Q_1,Q_1,Q_2,Q_3 ...Q_T which could have produced the given sequence. There is no "correct" sequence to be found here, but we try to find the best sequence based on certain optimality criteria. Different criteria are used to serve different purposes like to find out the structure of the model, or to find the phonemes in a word,or get the statistics of a model,etc.

- Training Problem: Here we try to fit the parameters of the HMM (A, B, π) according to a given observation sequence. This is one of the most crucial problems to most applications, as we have to adapt and train the parameters to the observed sequence. The aim is to maximize $P(O|\lambda)$.

Identifying the problems described above as being analogous to steps in the detection and classification of Epileptic transients, and then building and tuning those models is the main goal of this research. The various choices in parameters, algorithms and methods for classification or detection have been covered in the next 2 chapters.

Chapter 4

Classification of Yellow Boxes

As described in Chapter 2, data from the EEGs of 200 patients has been obtained and been stored in a MATLAB struct for all further processing. This chapter describes a method to classify each annotation into one of the 2 classes,as an AEP or a Non-AEP. An HMM based classifier is designed and trained on the given data, and it's performance parameters for various configurations has been discussed. This diagram shows the various steps involved in the designing of a 2-class classifier.

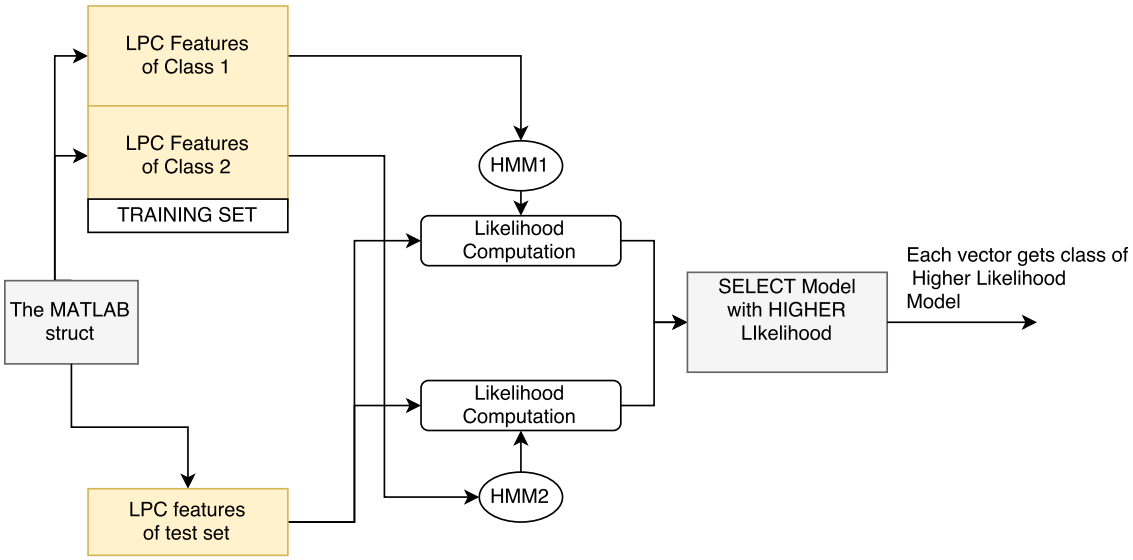

Figure 4.1: AN HMM Based Classifier for 2 classes of EEG signal

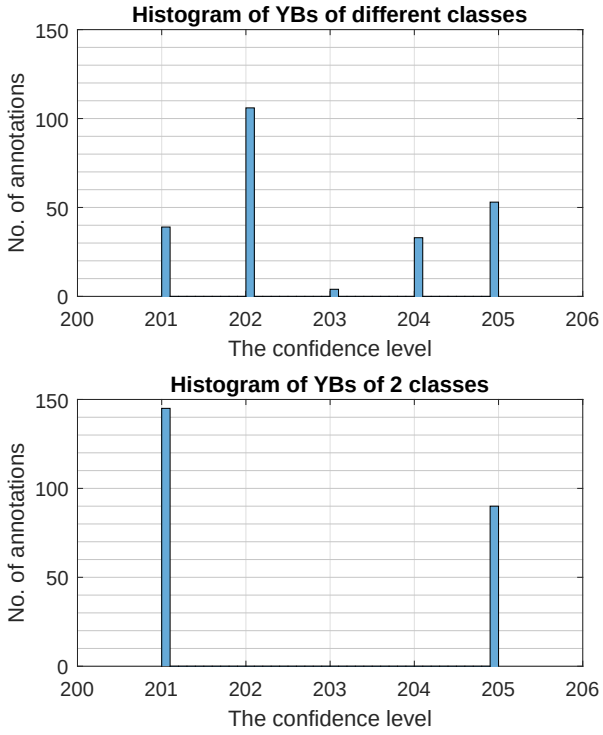

Figure 4.2: Histogram Showing Distribution of YBs Across Different Classes

4.1 The Classification Data

In our simulation, the MATLAB struct contains all the annotations in the `overall_db.spike` variable.The true class of each annotation has been stored in the `inferred_groundtruth`. This value is the mode of the scores given to a particular annotation by each of the seven experts. For the purpose of classification, the data is divided into 2 classes, AEP and Non-AEP based on the `inferred_groundtruth` value. A confidence level of 201 or 202 is assigned to class 1, and the 2nd class is of annotations having confidence level of 204 or 205. The histogram of Figure 4.2 shows the distribution of annotations across different classes, and after conversion to the 2 class problem. Hence, we can see that we have an imbalanced dataset for training, and the ratio of imbalance is maintained throughout the test set. This data is then divided into 2 parts, the test set and the training set.

4.2 Extraction of LPC features

The theoretical background for LPC feature extraction has been described in the previous chapter. LPC features are extracted from each of the annotated YBs and stored in a separate dataset. The order of the LPC (p) is varied from 6 to 12, and the performance is shown in the results section of this chapter. A rule of thumb is to have 2 linear prediction coefficients for each dominant frequency in the signal [5]. All further results shown are for LPC with p = 8. A limiting factor to the order of LPC is it has to be 1 less than the length of the signal. Since we have some 205 annotations as short as 11 samples, we have an upper bound for the LPC order of 10. The autocorrelation method described in chapter 2 has been used to find out the LP coefficients, and a feature vector of 235 X 8 has been formed. This is further divided into the training set and test set.

4.3 Implementation of the HMM

As shown in Fig4.1, a separate HMM has to be designed for the 2 classes in our data. HMM1 has been trained on the LPC features extracted AEP annotations from the training set. The LPC features extracted from the Non-AEP annotations have been used to train HMM2. Since an EEG signal is a time series, we use an left-right (Baki's) model of an HMM. So, transitions to states having a lower index than the current state is prohibited. There are many parameters in training of an HMM, and they are :

- Number of States (N) : One of the rules for deciding the number of states is to let it correspond to the number of identifiable sub-signals in a sample. This assigns a physical meaning to the state. We know that an AEP YB must have a spike, possibly followed by a slow wave, and some noise or artifacts. A safe choice of N = 3 has been made for HMM1. Since the Non-AEPs do not follow this particular behaviour, the HMM2 has been modeled using 4 states to allow freedom to accommodate different types of artefacts.

- Number of observations (M): It has been proved [18] that a single Gaussian cannot fit an EEG signal efficiently, and hence, in order to have sufficient data to make reliable estimates of the model parameters, we use multiple observation sequences. This is called as the Gaussian Mixture Model (GMM). M would be the number of mixtures in each state. Theory suggests that a higher model of M would provide for better modeling of the signal, but since we are

restricted by having some signals of very short duration, a high value of M would leave us with very few estimates for these signals. This causes errors in estimating the signal properly. A value of M = 4 is seen to provide better classification results than other values.

- The covariance matrix for these observations are constrained to be diagonal. So, all the states use a diagonal matrix. The reason is that diagonal matrices enable easy and efficient re-estimation of parameters.

- Minimum covariance : To prevent over-fitting of the data, the diagonal of the covariance matrix of the observation vectors must be floored to some small value, and has been set as 0.001.

- Minimum limit on parameter estimates : For each state, we should always have all non zero $b_j(k)$. This means that even though the k^{th} observation never occurs in a given state in the training set, there should always be a finite probability to allow occurrence of that observation during an unknown annotation from the test set. This has also been set to 0.001.

- Algorithm to find best state sequence: As described previously, there is not a single answer to find the best state sequence. The Viterbi algorithm is a dynamic programming algorithm which is used for this purpose. Papers by Viterbi [22] and Forney [8] introduced and described this algorithm.

- Convergence and No. of iterations: These are the 2 terminating conditions in the training of the HMM. The training of HMMs involves increasing the likelihood score of a model for the given set of observation vectors. The tolerance level means the level of increase in this score from the previous iteration, and has been set as 0.001. This means that if the increase in score for the current iteration is less than 0.001, the model is said to have converged. Most HMMs require around 25-40 iterations to converge, but it is dependent on other factors like the tolerance, the data, and proper choice of modeling parameters. A reasonable number of iterations is 50. The training stops when either of these 2 conditions is met.

The above section concludes the description of the training part of the HMMs. Once the 2 HMMs (λ^1 and λ^2) have been trained, we proceed to the testing part. Each vector from the test set is fed individually to the 2 HMMs. Then, for each vector, we calculate $P(O|\lambda^1)$ and $P(O|\lambda^2)$. Then, class assignment is simply selecting the class of the HMM with the higher likelihood score.

$$\text{Class of unknown vector } = argmax[P(O|\lambda^1, P(O|\lambda^2] \tag{4.1}$$

4.4 Cross Validation

Cross validation, or rotation estimation is necessary to give an insight on how the trained model would generalize to an independent dataset, and limits problems like overfitting and variability In k-fold cross validation, the dataset is randomly partitioned into k equally sized subsets. Out of these k subsets, one of the subset is used to test the model, and the other $k-1$ subsets are used to train the model. This process is then repeated k times, with each subset being used as the test set exactly once. *Stratified* k fold sampling involved selecting the k folds such that each subset contains the same proportion of the two class labels, AEP and Non-AEP. Thus, a more accurate performance of the model can be obtained by taking averages of the performance statistics over these k subsets.

According to the stratified k-fold method, cross validation of the trained model from k = 4 to 10 has been performed. The Table 4.1 shows how each subset has been designed, and the number of AEP and Non-AEP samples in each subset.

4.5 Results

Convergence and convergence plots

Before testing the HMM on the test set, it must be ensured that the model has been properly trained, and has achieved convergence. For HMMs, the model is trained by repeatedly adjusting the parameters of the model to increase the likelihood probability of the observed sequence. This is done by the reestimating procedure described here. For a given HMM ($\lambda = (A, B, \pi)$), we define

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i \tag{4.2a}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from } S_i \text{ to } S_j \tag{4.2b}$$

No of folds (k)	Subset No	No. Of AEP	No. Of Non-AEP	Ratio of AEP/Non-AEP	Total
4	1	23	35	1.52	58
	2	22	36	1.63	58
	3	22	36	1.63	58
	4	23	36	1.56	59
5	1	17	30	1.76	47
	2	17	30	1.76	47
	3	17	30	1.76	47
	4	17	30	1.76	47
	5	18	29	1.61	47
6	1	14	25	1.78	39
	2	14	25	1.78	39
	3	14	25	1.78	39
	4	15	24	1.6	39
	5	14	25	1.78	39
	6	15	25	1.66	40
7	1	12	21	1.75	33
	2	12	21	1.75	33
	3	12	21	1.75	33
	4	12	22	1.83	34
	5	12	22	1.83	34
	6	13	21	1.61	34
	7	13	21	1.61	34
8	1	10	19	1.9	29
	2	10	19	1.9	29
	3	11	18	1.63	29
	4	11	18	1.63	29
	5	11	18	1.63	29
	6	11	19	1.72	30
	7	11	19	1.72	30
	8	11	19	1.72	30
9	1	9	17	1.88	26
	2	9	17	1.88	26
	3	9	17	1.88	26
	4	9	17	1.88	26
	5	9	17	1.88	26
	6	10	16	1.6	26
	7	10	16	1.6	26
	8	10	16	1.6	26
	9	10	16	1.6	27
10	1	8	15	1.875	23
	2	8	15	1.875	23
	3	8	15	1.875	23
	4	8	15	1.875	23
	5	9	14	1.55	23
	6	9	15	1.66	24
	7	9	15	1.66	24
	8	9	15	1.66	24
	9	9	15	1.66	24
	10	9	15	1.66	24

Table 4.1: Showing the Division of Samples from the Dataset using Stratified Sampling

A set of reestimation formulae for $\pi, A,$ and B are

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time (t=1)} = \gamma_1(i) \tag{4.3a}$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{expected number of transitions from } S_i \text{ to } S_j}{\text{expected number of transitions from } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \tag{4.3b}$$

$$\begin{aligned} \bar{b}_j(k) &= \frac{\text{expected number of times in state j and observing symbol } v_k}{\text{expected number of times in state j}} \\ &= \frac{\sum\limits_{t=1 \text{ s.t. } O_t=v_k}^{T-1} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)} \end{aligned} \tag{4.3c}$$

The right hand sides of (4.3a)-(4.3c) are computed using the current model ($\lambda = (A, B, \pi)$). The left hand sides are (4.3a)-(4.3c) are then estimated and called as ($\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$) It has been proven by Baum that [3] either 1)the initial model λ defines a critical point of the likelihood function or 2) model $\bar{\lambda}$ is more likely to have produced the given observation sequence in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$. This means we have found a new model ($\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$), which has a higher probability of producing the given observation sequence.

The convergence plots for all the cross-validation trials have been shown, and convergence can be seen as the increase in log probability between current iteration and the previous iteration is less than 0.001, which we have set as our tolerance parameter. As is typical of HMMs, the two HMMs may require different number of iterations to achieve convergence. In all these figures (4.3)-(4.9), HMM1 stands for the HMM trained of AEP samples, and HMM2 stands for the HMM trained on Non-AEP samples.

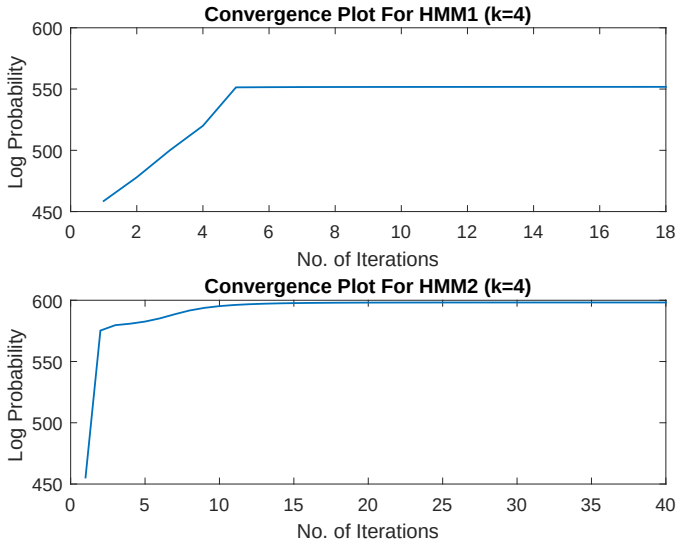

Figure 4.3: Convergence plot for k = 4

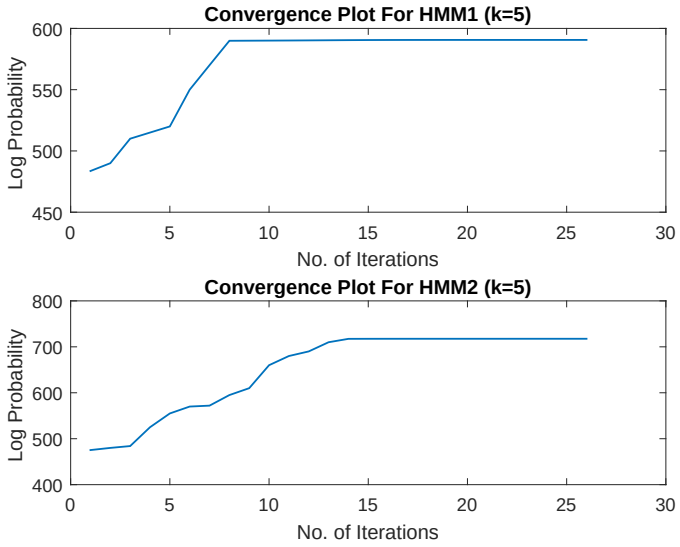

Figure 4.4: Convergence plot for k = 5

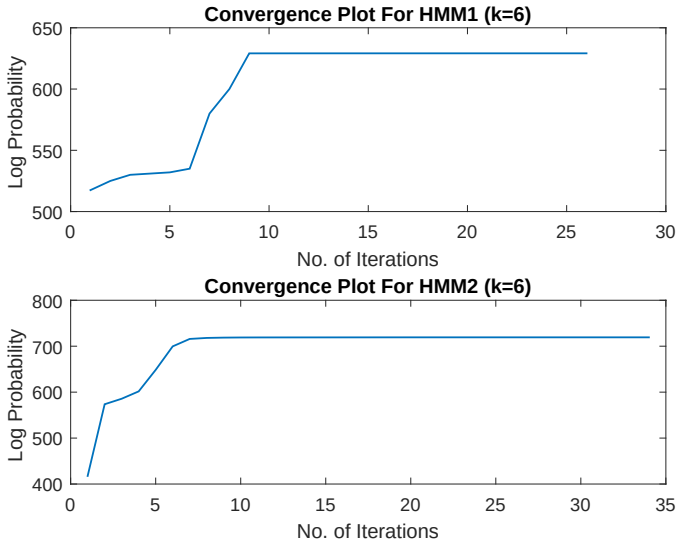

Figure 4.5: Convergence plot for k = 6

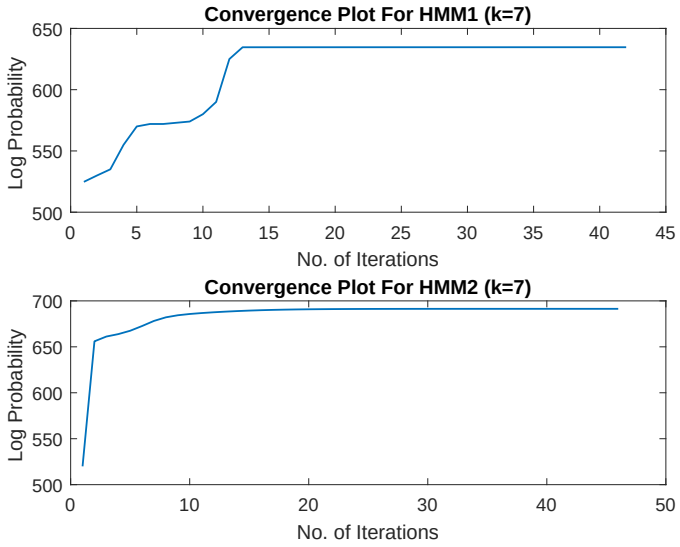

Figure 4.6: Convergence plot for k = 7

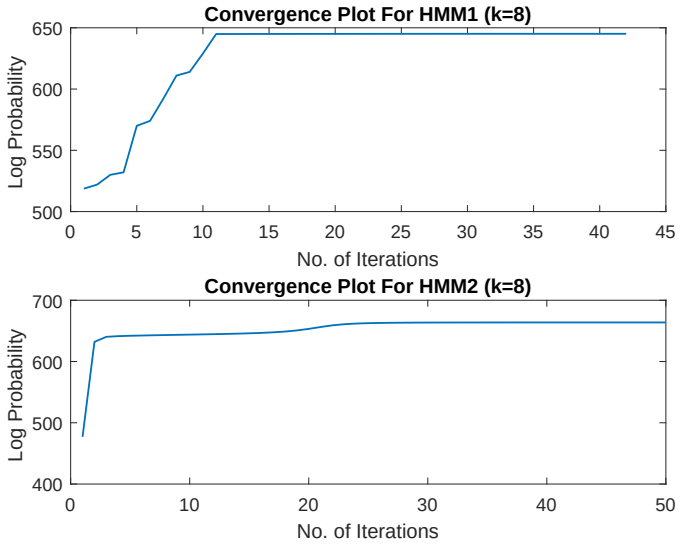

Figure 4.7: Convergence plot for k = 8

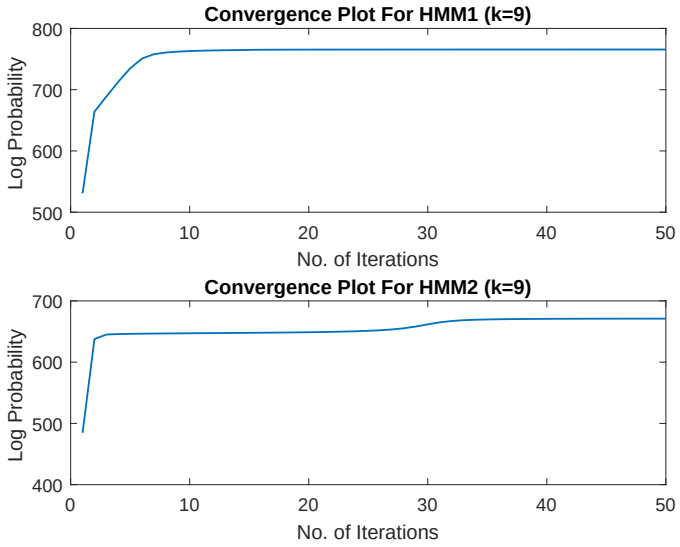

Figure 4.8: Convergence plot for k = 9

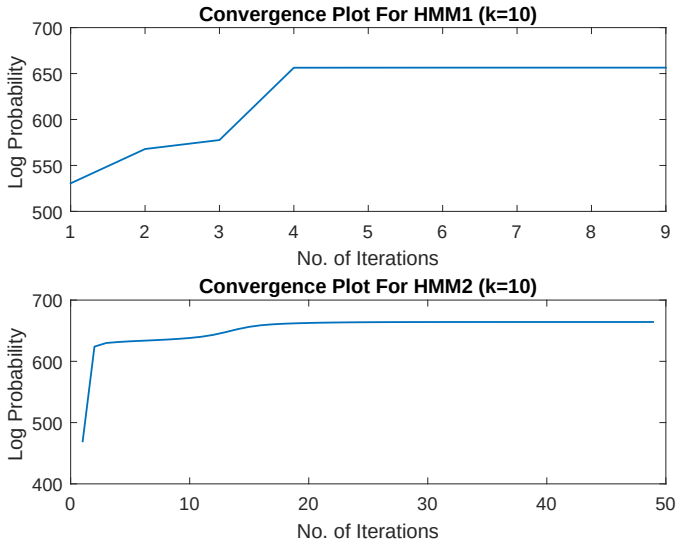

Figure 4.9: Convergence plot for k = 10

Confusion matrix and performance parameters

For studying the performance of any classifier, in the field of predictive analysis, a confusion matrix is always better than metrics like accuracy, which can yield misleading results. Accuracy measures the percentage of misclassified samples. In the case of an unbalanced data set, accuracy measures would favor the classifiers which tend to classify all samples as belonging to the majority class. so, as an alternative, we have a confusion matrix, which is a table of 2 rows and 2 columns, and is shown in Fig 4.10. The different terms in the confusion matrix are all explained:

- A True Positive (TP) is obtained when the predictor classifies a sample as positive, when the actual ground truth is also positive. It is equivalent to a 'hit'.
- A True Negative (TN) is obtained when the predictor classifies a sample as negative, when the actual ground truth is also negative. It is also called as a 'correct rejection'.
- A False Positive (TP) or False Alarm is obtained when the predictor classifies a sample as positive, when the actual ground truth is negative. It is equivalent to the Type I error.
- A False Negative (TP) is obtained when the predictor classifies a sample as negative, when the actual ground truth is positive. It is equivalent to a 'miss' or the Type II error.

From the confusion matrix, 3 performance metrics, namely sensitivity, specificity and precision have been calculated. These are defined as

- Sensitivity, or the True Positive Rate (TPR) is the proportion of positives identified correctly from the total number of positives. Sensitivity quantifies the avoiding of false negatives.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{Total No. of Positives}} \tag{4.4}$$

- Specificity, or the True Negative Rate (TNR) is the proportion of negatives identified correctly from the total number of negatives. Sensitivity quantifies the avoiding of false positives.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{Total No. of Negatives}} \tag{4.5}$$

- Precision, or the Positive Predictive Value (PPV) is the proportion of positives identified correctly, and it gives the accuracy rate for the positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{Positive Outputs}} \tag{4.6}$$

The Table 4.2 gives the above mentioned metrics for each trial of cross-validation. The next Table 4.3 gives us the average values of the performance metrics over all the trials for a particular value of k. These are the most reliable metrics of the performance of the model. A plot of these 3 metrics in percentage, versus the k values is shown in Fig 4.11.

		Ground Truth	
		Positive	Negative
Prediction	Positive	True Positives	False Positives
	Negative	False Negatives	True Negatives

Figure 4.10: A 2X2 Contingency table, also called as a Confusion Matrix

Table 4.2: The Performance Metrics for Each Trial of Cross-Validation

k value	Trial No.	Sensitivity	Specificity	Precision
4	1	0.87	0.91	0.87
	2	0.83	0.91	0.86
	3	0.83	0.91	0.86
	4	0.83	0.91	0.87
5	1	0.82	0.90	0.82
	2	0.87	0.88	0.76
	3	0.87	0.88	0.76
	4	0.81	0.87	0.76
	5	0.78	0.86	0.78
6	1	0.80	0.92	0.86
	2	0.85	0.88	0.79
	3	0.80	0.92	0.86
	4	0.79	0.84	0.73
	5	0.83	0.85	0.71
	6	0.92	0.86	0.73
7	1	0.77	0.90	0.83
	2	0.71	0.89	0.83
	3	0.69	0.85	0.75
	4	0.69	0.86	0.75
	5	0.77	0.90	0.83
	6	0.83	0.86	0.77
	7	0.79	0.90	0.85
8	1	0.89	0.90	0.80
	2	0.90	0.95	0.90
	3	0.73	0.83	0.73
	4	0.73	0.83	0.73
	5	0.82	0.89	0.82
	6	0.82	0.89	0.82
	7	0.90	0.90	0.82
	8	0.89	0.86	0.73
9	1	0.89	0.94	0.89
	2	0.80	0.94	0.89
	3	0.78	0.88	0.78
	4	0.88	0.89	0.78
	5	0.89	0.94	0.89
	6	0.89	0.88	0.80
	7	0.80	0.88	0.80
	8	0.82	0.93	0.90
	9	0.90	0.94	0.90
10	1	0.75	0.87	0.75
	2	0.86	0.88	0.75
	3	0.86	0.88	0.75
	4	0.88	0.93	0.88
	5	0.88	0.87	0.78
	6	0.88	0.88	0.78
	7	0.89	0.93	0.89
	8	0.78	0.87	0.78
	9	0.75	0.81	0.67
	10	0.89	0.93	0.89

Table 4.3: Average Values of the Performance Metrics

k value	Sensitivity	Specificity	Precision
4	0.84	0.91	0.87
5	0.83	0.88	0.78
6	0.83	0.88	0.78
7	0.75	0.88	0.80
8	0.83	0.88	0.79
9	0.85	0.91	0.85
10	0.84	0.88	0.79

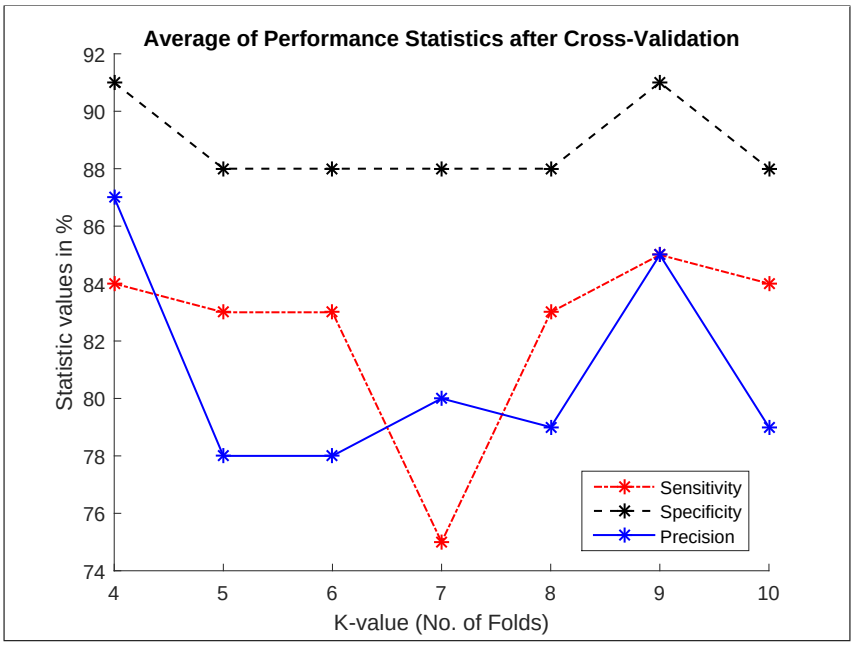

Figure 4.11: A Plot of the 3 Performance Metrics Over All Values of k (4-10) in Cross-Validation

Conclusion

From Table 4.3, it can be seen that k=9 provides for the best results in terms of sensitivity and specificity. From Fig 4.11, we can see that the specificity is always higher then the sensitivity. This means that HMM2 (Non-AEP) is slightly dominating over HMM1 (AEP), which is because of a higher number of training samples available for that class. Also, the convergence plots show that HMM2 takes more iterations to train, but also has higher accuracy than HMM1. Overall, the performance can be improved by increasing the number of training samples. Also, multi-resolution analysis, which involves sampling the EEG at higher rates ($>$ 1028Hz) is shown to have provided an improved performance in speech recognition applications, and could be applied to EEG signals.

Some speech recognition systems have used hybrid classifiers, where the multilayer perceptron (MLP) class-posterior probability have been appended to the features of an HMM. [21]

Chapter 5

Detection of Epileptic Transients from EEG

In Chapter 2, the methodology of obtaining the data of EEG signals, and the scoring process by the experts has been explained. The experts have to monitor hours of EEG to detect the presence of any Epileptiform transients (ETs). The goal of this chapter is to be able to automatically detect the location of ETs, irrespective of whether it is an AEP or a Non-AEP. So, we are trying to automate the process of annotating the data. An advantage of using an HMM based classifier is that the same model used for classification can be reused for detection, but the training data would be different.

5.1 The Detection Data

Having proper data to train the classifier is of paramount importance. The data obtained from the experts defines where and on what channel an ET is present. This data, as described previously, has been saved as a MATLAB struct. The 2nd class would be the Non-ET data, and the methodology of obtaining it is described. The total data available to us is in the form of 200 EEG files, each file belonging to a separate patient. The Figure 5.1 shows a histogram of the number of annotations in each file. Out the 200 files, there are 99 files which contain one or more Yellow Boxes (YBs) in them. The other 101 files have been monitored by all 7 experts and do not have any ETs in

them, and have not been annotated. Fig 5.1 shows that many files contain multiple annotations, with some files contain more than 5 annotations. For designing the training data, a simple rule followed is 'Everything that is not annotated can be used as training data for the Non-ET class'. But, to design a good training data, some precautions have been taken, and can be seen in Fig 5.2. The experts view the EEG signals on some specific channels (A channel is the difference of 2 electrodes). In our data, the experts have chosen 59 different channels to identify all the 235 annotations. So, all of the 101 files, when viewed on any of these 59 channels, should provide Non-ET data. Most of this is the normal brain activity, and does not contain any artifacts or activities. Since this data is so different from the ETs, it is not sufficient to train a good detector.

In the other 99 EEG files, apart from the portion marked as ETs, everything still falls under the Non-ET class and has been included in the dataset. Some files which contain a large number of ETs have been avoided.

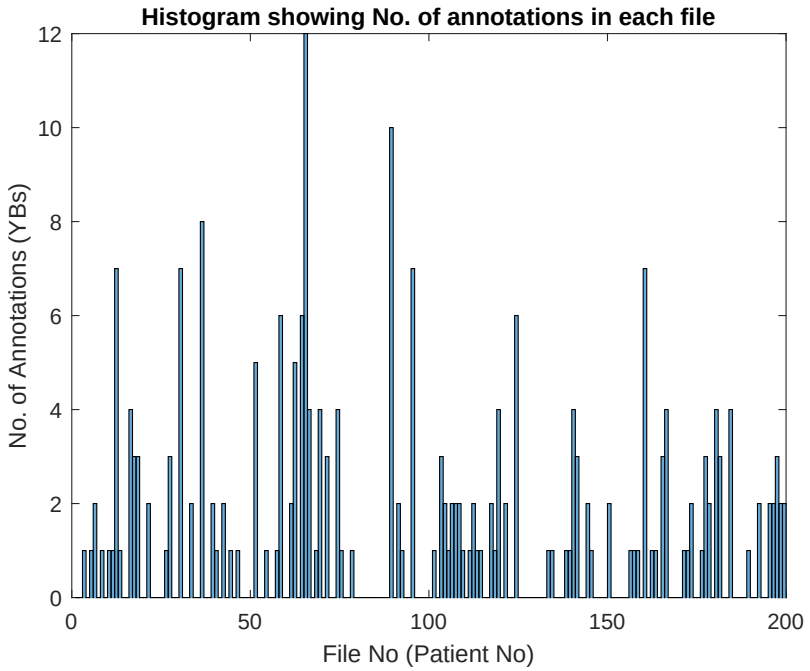

Figure 5.1: A Histogram showing the number of annotations marked in each file by the experts

5.2 Duration of the Non-ET

The 235 ET annotations have a huge variability in their duration,shown in Table 5.1. Window length for data from Non-ET has also been chosen to follow a similar distribution.
So, for the Non-ET data, parts of the signals ranging from 17 samples to 110 samples have been chosen from the 200 EEG files on the 59 different channel combinations. With all the combinations

Mean	Median	Std. Dev	Minimum	Maximum	5^{th} percentile	95^{th} percentile
48	38	27	11	315	17	110

Table 5.1: Statistics for the Duration of the 235 Annotations

mentioned above, 12000 Non-ET data has been extracted. The fig 5.2 shows a complete picture of the data extraction process. After that, LPC features similar to the classification method are derived from the data and stored in a MATLAB struct. This struct has the exact structure as the one described in the Chapter 2.

5.3 Contextual Information

The presence of an ET, detected by visual inspection, is done by comparison with the surrounding data. So, it is necessary to inculcate that information into our dataset. So, the existing MATLAB struct has been modified to include 50 samples before and 50 samples after the data. Since the average duration of the ETs is around 48 samples, this window is deemed sufficient for comparison of the annotated part with it's surrounding. This data has been stored in the variables `overall_db.precontext` and `overall_db.postcontext`. So, one complete sample can be accessed as `[overall_db.precontext overall_db.spike overall_db.postcontext]`.

5.4 Balancing of Data

HMMs are extremely sensitive to imbalanced data, i.e data which has unequal number of samples of each of it's classes. As described above, 12000 Non-ET features have been extracted, whereas only 235 annotations have been used. This represents a hugely imbalanced data. In literature, many data balancing techniques have been proposed such as upsampling, selective sampling, random sampling, etc. In this study, an informed undersampling of the Non_ET samples has been

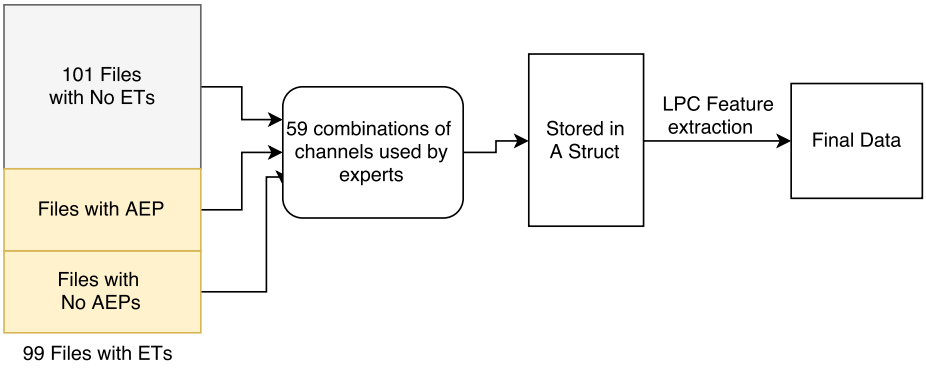

Figure 5.2: Block Diagram Showing Steps in Extraction of Non-ET Data

performed.Some techniques which rely on maximizing a distance measure between the 2 classes have also been suggested, but have been avoided because they do not represent the reality of many non-epileptiform transients actually looking very similar to epileptiform transients. [9] So, to provide for the Non-ET data, 100 samples have been chosen from files which do not have a single ET. 65 samples from files which do have an AEP transients and 70 samples which have only Non-AEP transients have been selected. Now, the dataset is balanced with 235 annotations of each class.

5.5 Cross-Validation

Since we have equal number of samples for each labels, cross validation is easier to perform. k-fold validation, with k values from (5-10) has been performed. The Table 5.2 shows how the folds have been formed, and the performance for each trial.

5.6 Implementation of HMM-based Detector

- The model type for the two detector HMMs is the left-right model (Baki's) having $a_{ij} = 0$ for $j > i$. The covariance matrix is then a diagonal matrix.
- N, the number of states: Since the contextual information is also being used, a higher number of states are required. N is set to be 5.
- M, the number of Gaussian mixtures : M is chosen to be 4, so there are 4 Gaussian mixtures per state.

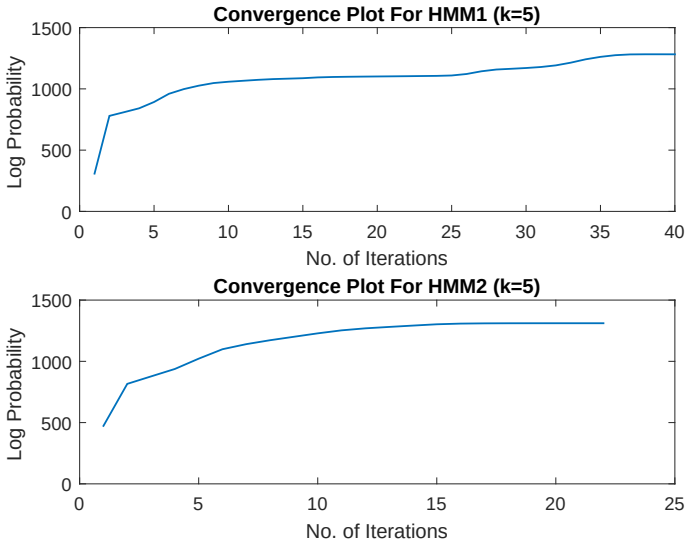

Figure 5.3: Convergence plot for k = 5

- Minimum covariance (0.001), minimum limit of parameter estimates (0.001), tolerance level(0.0001) and the algorithm used are all the same as in the classification problem.
- Convergence : Since there are a higher number of samples to train from, it is observed that the HMM takes a lesser number of iterations to converge. The limit is still set at 50, or the tolerance of 0.0001, whichever is reached earlier.

5.7 Results

Convergence Plots

Two HMMs, one for each class, have been trained on the dataset designed above. To check for convergence before testing, the log probability at each iteration is plotted. The convergence is achieved by repeatedly reestimating the model parameters of the HMMs. Fig (5.3)-(5.8) show the convergence plots for the two HMMs. In these figures, HMM1 stands for the ETs, and the HMM2 stands for the Non-ETs. Also, some plots show a sudden increase in convergence after a few iterations. This is indicative of the algortihm approaching a local minima, which is a problem the HMMs always suffer from. Also, it cannot be considered as a converged state, as the difference in the sum of log probabilities is still greater than the tolerance level that has been set. (0.001)

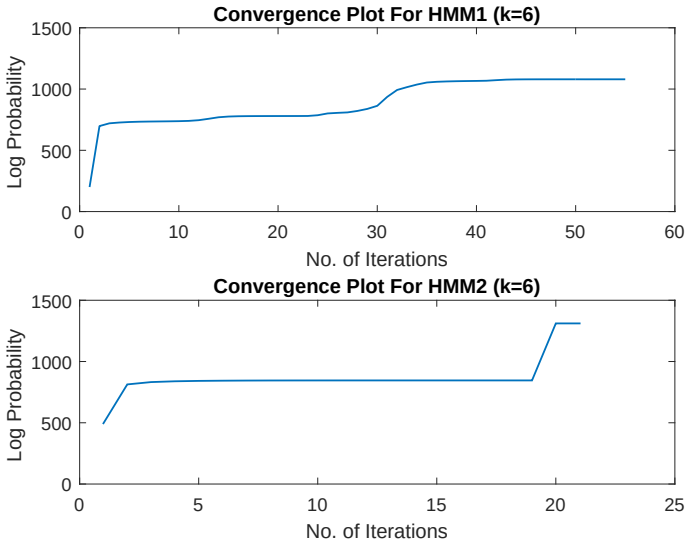

Figure 5.4: Convergence plot for k = 6

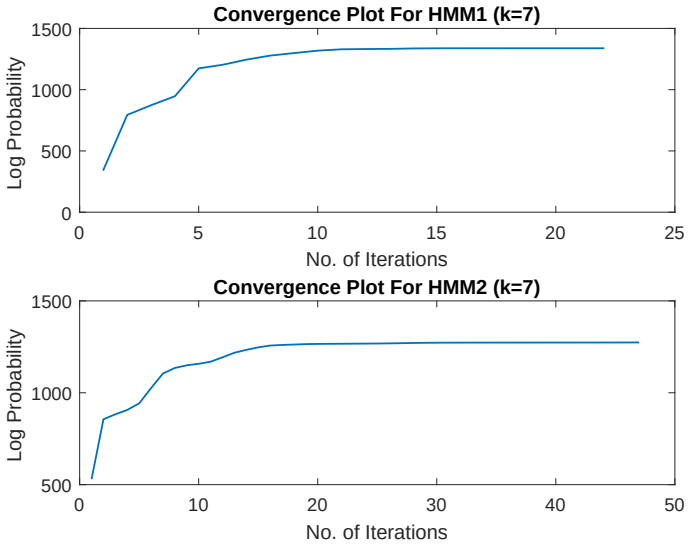

Figure 5.5: Convergence plot for k = 7

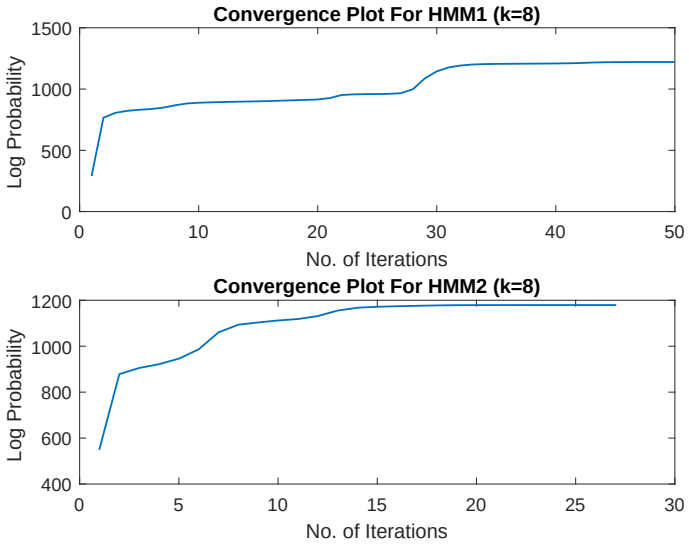

Figure 5.6: Convergence plot for k = 8

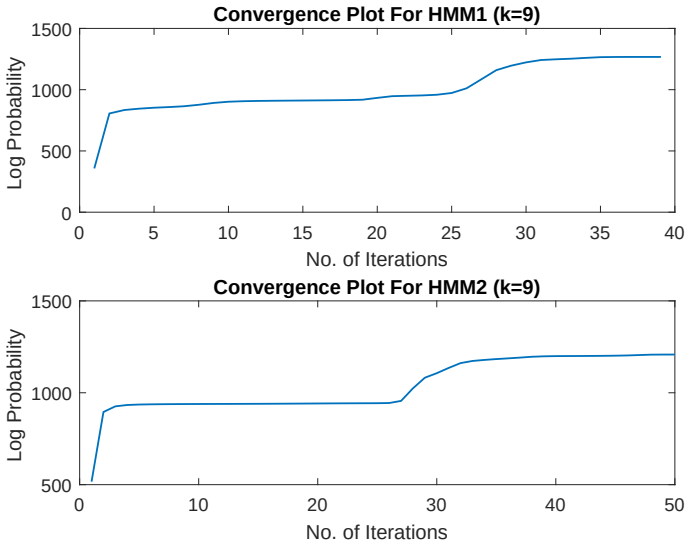

Figure 5.7: Convergence plot for k = 9

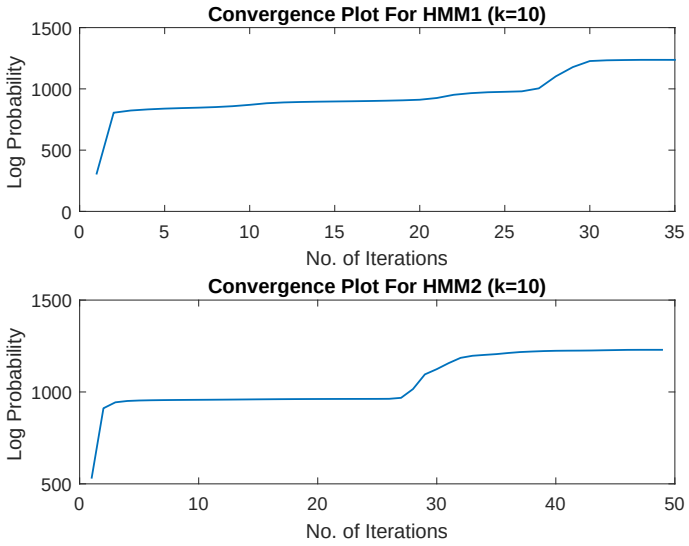

Figure 5.8: Convergence plot for k = 10

Performance for each trial of cross-validation

Cross validation by k-folds, where k is from $5-10$, has been performed to ensure the validity of results, and provide more accurate measure of performance of the detector. The Table 5.2 shows how each fold in the validation has been formed, and the performance of the detector on each fold. The average performance measures for each value of k has been shown in Table 5.3 and plotted in Fig 5.9 .

5.7.1 Discussion of results

As can be seen from the performance measures, there is a significant improvement in the detection of YBs as compared to classification. This is because of a higher number of samples available for training the HMMs. Also, since the lengths of the individual sequences are higher, and the LPC models are a better fit for characterizing the Non-ET and ET data. Better models can be built by using hybrid forms of HMMs, where multilayer perceptrons or clustering methods are used to get the Non-ET data. We can see that performance gets better with increase in number of folds, as the training samples increases.

K Value	Subsets	Class ET	Class Non-ET	Total	Sensitivity	Specificity	Precision
5	1	47	47	94	0.87	0.94	0.93
	2	47	47	94	0.86	0.91	0.9
	3	47	47	94	0.86	0.91	0.93
	4	47	47	94	0.87	0.94	0.9
	5	47	47	94	0.88	0.95	0.93
6	1	39	40	79	0.9	0.93	0.91
	2	39	39	78	0.85	0.91	0.87
	3	39	39	78	0.72	0.8	0.79
	4	39	39	78	0.87	0.9	0.89
	5	39	39	78	0.82	0.95	0.89
	6	40	39	79	0.9	0.93	0.91
7	1	33	34	67	0.82	0.94	0.89
	2	33	34	67	0.82	0.91	0.87
	3	33	34	67	0.74	0.88	0.82
	4	34	34	68	0.85	0.91	0.88
	5	34	33	67	0.82	0.88	0.85
	6	34	33	67	0.88	0.88	0.88
	7	34	33	67	0.88	0.91	0.9
8	1	29	30	59	0.87	0.93	0.89
	2	29	30	59	0.86	0.93	0.87
	3	29	30	59	0.93	0.86	0.87
	4	29	29	58	0.85	0.86	0.88
	5	29	29	58	0.92	0.96	0.87
	6	30	29	59	0.93	0.9	0.9
	7	30	29	59	0.86	0.9	0.83
	8	30	29	59	0.83	0.86	0.91
9	1	26	27	53	0.96	0.83	0.95
	2	26	26	52	0.83	0.96	0.83
	3	26	26	52	0.9	0.94	0.86
	4	26	26	52	0.88	0.91	0.92
	5	26	26	52	0.91	0.88	0.94
	6	26	26	52	0.93	0.9	0.91
	7	26	26	52	0.89	0.92	0.88
	8	26	26	52	0.91	0.91	0.93
	9	27	26	53	0.9	0.91	0.94
10	1	23	24	47	0.92	0.88	0.9
	2	23	24	47	0.89	0.94	0.88
	3	23	24	47	0.95	0.96	0.87
	4	23	24	47	0.96	0.95	0.93
	5	23	24	47	0.88	0.93	0.96
	6	24	23	47	0.95	0.84	0.94
	7	24	23	47	0.96	0.94	0.88
	8	24	23	47	0.88	0.93	0.91
	9	24	23	47	0.92	0.93	0.93
	10	24	23	47	0.89	0.9	0.92

Table 5.2: Table showing how each fold for cross validation has been formed.The performance measures on each fold has also been given

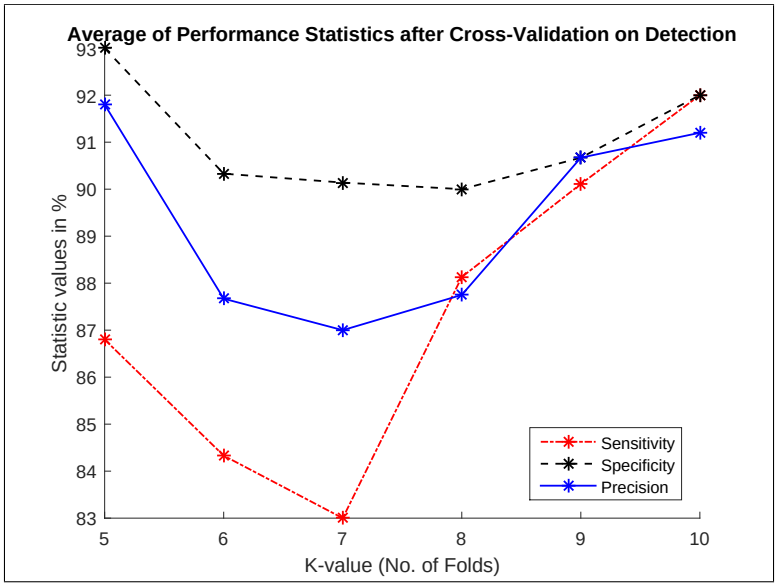

Figure 5.9: Plot of Average of Performance Statistics Over All k

	Sensitivity	Specificity	Precision
5	86.80	93.00	91.80
6	84.33	90.33	87.67
7	83.00	90.14	87.00
8	88.13	90.00	87.75
9	90.11	90.67	90.67
10	92.00	92.00	91.20

Table 5.3: Table of Average of Performance Statistics

Chapter 6

Validation of the Detector on Actual EEGs

After designing the detector and classifier HMMs individually, the next step is actual validation on the EEG files that we have. No more data processing or feature extraction is needed at this point. The trained HMMs from the previous chapters have been used for this purpose. The fig 6.7 shows the complete process of validation of the performance of the HMM based detection and classification system.

In order to check whether the above described HMM system works properly, only signals which already have some annotations in them are considered. This is so that we can compare the perfor-

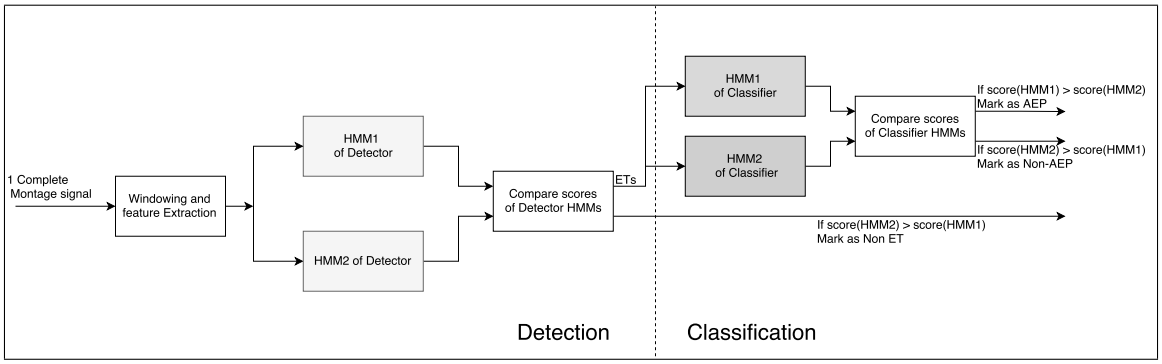

Figure 6.1: The Complete Process of Detection and Classification of ETs from 1 Montage Signal

	Actual Epileptiform Transients		
File No	AEP	Non-AEP	Total
89	10	0	10
65	12	0	12
12	4	3	7
160	3	4	7
36	8	0	8
95	4	3	7
124	0	6	6
23	No Epileptiform Transients		
84	No Epileptiform Transients		

Table 6.1: A Table Showing the Files used in Validation and the No. of ETs and Non-ETs in them.

mance of our detector with the experts diagnosis. The Table 6.1 shows which are the files selected to test the performance, and the number of ETs in each file.

For each file, there may be multiple annotations, but they are mostly annotated on separate channels. So, the channels having the highest number of annotations were chosen to run through the detector. A window is slid across the EEG montage, and LPC features are extracted from it. Then, these features are fed to the detector, and checked if it is an ET. If identified as being of class 2, i.e the Non-ET class, then the signal is marked as Non-ET and the window moves ahead. If not, i.e, if the window is identified as an ET, it needs to be passed through the classifier and checked which class of ETs it belongs to. After passing through the classifier, each window is labeled as an AEP or a Non-AEP appropriately. The fig show some examples of the above mentioned algorithm running on some of the signals mentioned above. All boxes represent that the detector has identified that part of the window as an ET. Red boxes are windows which have been identified correctly as belonging to the ET class, and black boxes are false positives.

The window length for HMMs has been a topic of debate over the years. The method of sliding a fixed length window across the signal always introduces noise or unwanted data. So, a fixed length sliding window has been avoided, and instead, a window having lengths similar to the annotations has been used. A huge reduction in false positives is observed because of this. Also, though there is a large variability in the lengths of the annotations, this variability is mostly introduced because of Non-AEP transients. Around 90% of the AEPs have a length in the range of $24 - 58$ samples. So, keeping the length of pre and post contextual window of 60 effectively covers all the AEP annotations, but may leave out some Non-AEP and other information. Both, the classifier and the detector are the fully trained HMMs, with the classifier HMMs trained by k=4 cross validation and

detector HMMs trained on k=9 cross validation.

6.1 Performance Measures:

Each montage is a 30 sec EEG signal, sampled at 256Hz, so has a length of 7680 samples. As this is a huge number compared to the window length and length of annotations, performance measures like sensitivity and specificity would not be accurate. In order to come up with a measure of performance for this system, two quantities are defined below:

- Marking ratio (M): The Marking ratio, is the ratio of the length of the signal marked, to the total length of signal that could have been marked. This measure tells us how much portion of the signal is marked, and a good recognition system would have it 0 for EEGs from healthy patients.

$$M = \frac{\sum \text{Length of each box}}{\sum \text{Total length of the EEG montage}} \tag{6.1}$$

- Marking Hit ratio (H) : The Marking hit ratio is the ratio of the length of all portions of signal correctly identified as ET, to the total length of the marked signal. This should ideally be 1, but due to the presence of many false positives, this value tends to be much lower.

$$H = \frac{\sum \text{Length of correctly identified ET segments}}{\sum \text{Total length of all marked segments}} \tag{6.2}$$

To summarize, a higher value of M and a lower value of H indicates huge number of false positive. A lower value of M and higher value of H indicates a good performance.

File No	Total Boxes	Red Boxes	M (%)	H(%)
89	29	10	8.28	37.73
65	31	12	10.26	27.41
12	27	4	12.13	10.30
160	25	3	10.98	11.37
36	35	8	14.21	15.38
95	23	4	9.01	17.34
124	40	0	17.91	10.46

Table 6.2: The table gives the Marking Ratio (M) and Marking Hit Ratio (H) for the validation

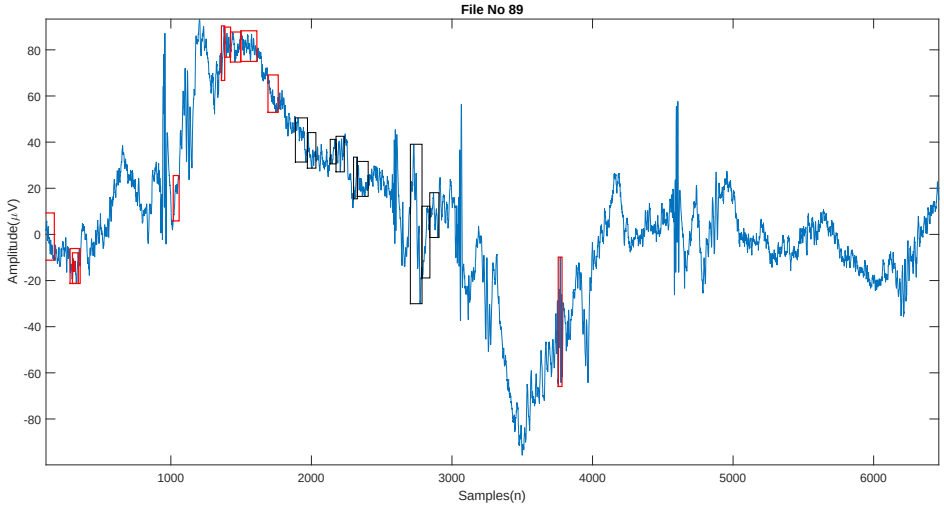

Figure 6.2: The validation of File No 89, which is the difference of the 2 Electrodes Fp1 - F3. It can be clearly seen that there are many false positives. All the ETs belonging to AEP class have been properly detected.

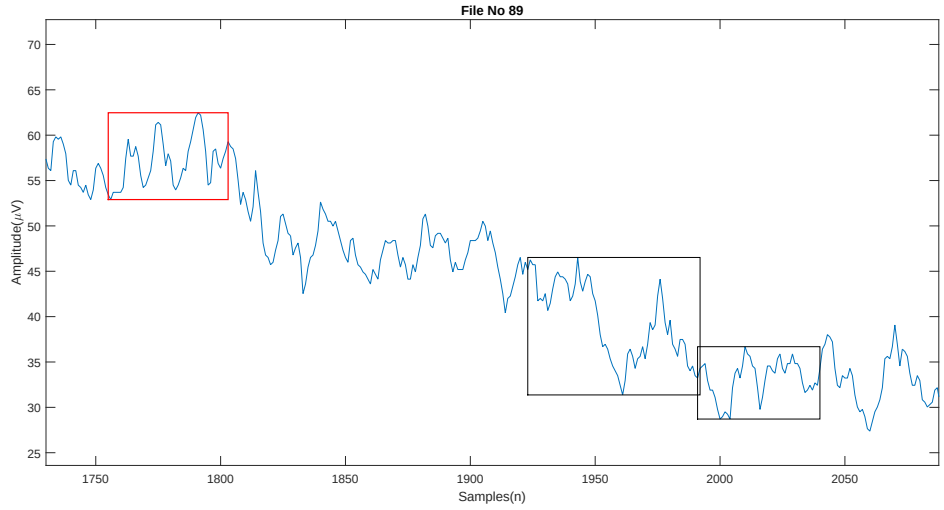

Figure 6.3: A closer look, showing that the red box, which is a correctly marked AEP (205), has included pre and post contextual data as well.

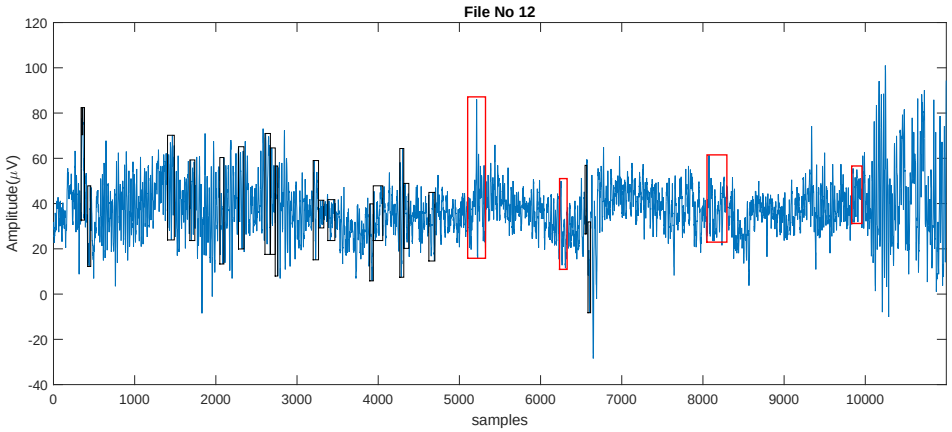

Figure 6.4: The validation of File No 12, which is the difference of the 2 Electrodes Fp1 - F7. Again, we see a huge number of False Positives. File 12 has 4 AEP and 3 Non-AEP. This detector has failed to identify the Non-AEP transients.

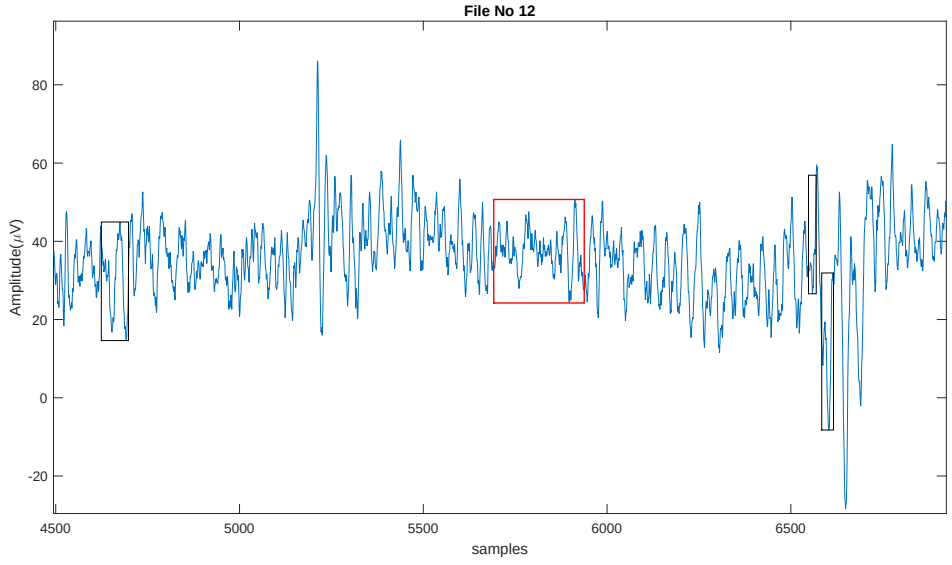

Figure 6.5: A closer look at file 12 shows us that the red box is an AEP (204).

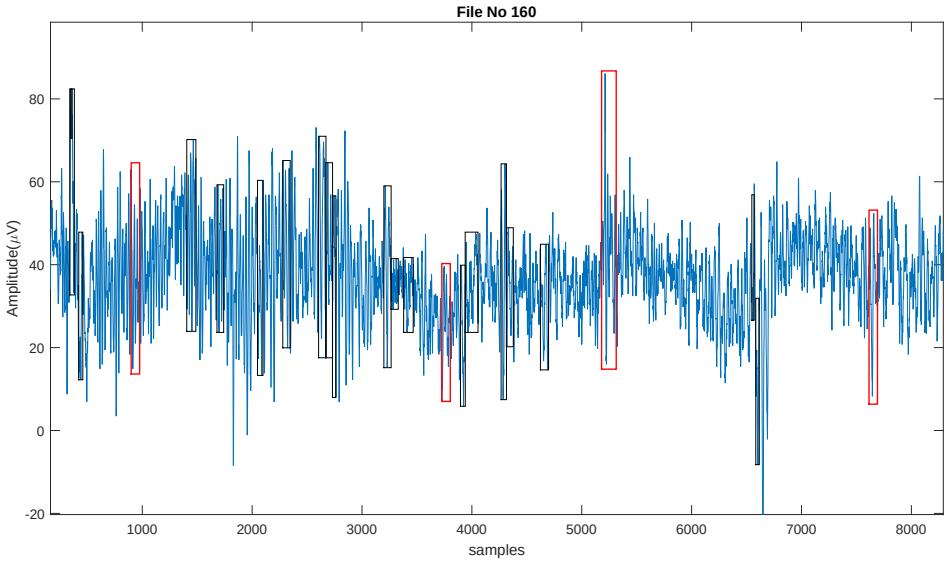

Figure 6.6: The validation of File No 160, which is the difference of the 2 Electrodes O2 - O1. Again, we see a huge number of False Positives. File 160 has 3 AEP and 4 NonAEP. This detector has failed to identify 3 of the Non-AEP transients

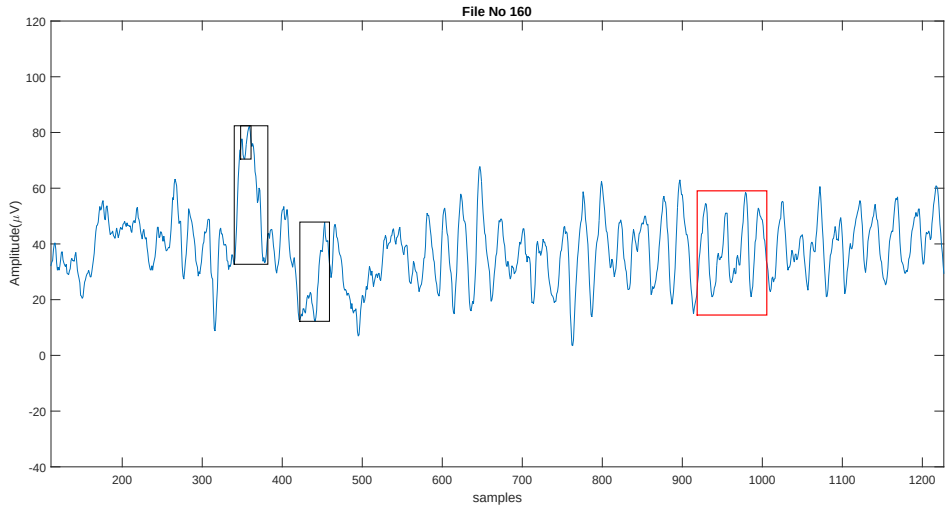

Figure 6.7: A closer look at File 160 shows us that the red box is a very long one, and it has been identified as a NonAEP. We can also see 2 black boxes overlapping each other.

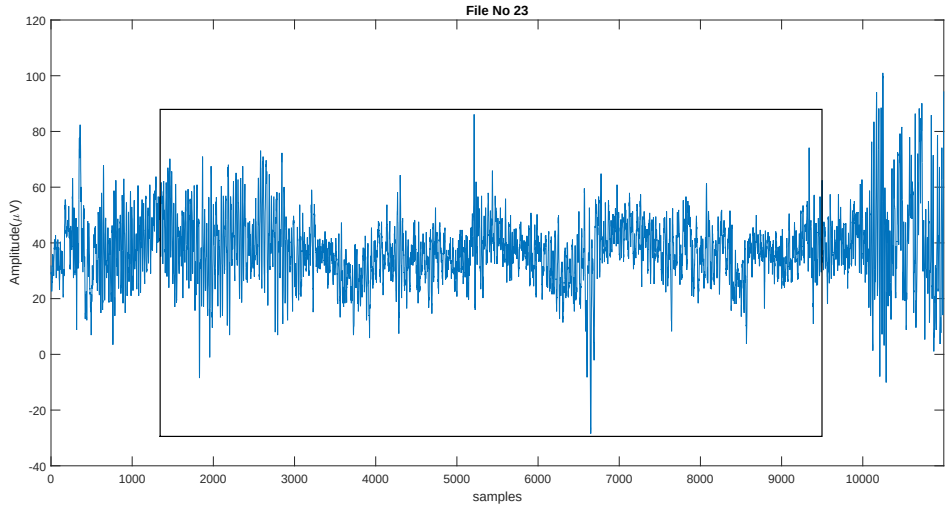

Figure 6.8: This image shows that not a single box was marked, and has perfect detection. Most of the files having 0 ETs are having good detection performance.

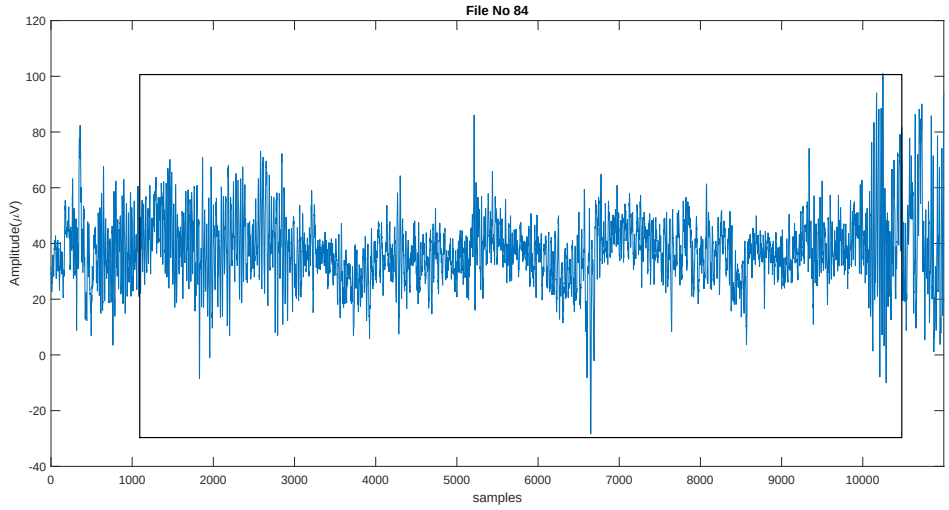

Figure 6.9: Another example having no ETs

6.2 Conclusion

We can see that the performance on detection data is very promising, but the same models when validated by individual EEG signals, give a lot of errors in the form of False Positives. A probable cause of error is the windowing technique applied to a signal. Also, many black boxes were connected (adjacent) to each other. The behaviour of LPC features, which are weighted coefficients in succession, might be the reason behind this.

Other techniques, like thresholding the EEGs, or using ANNs in combination with HMMs are being studied and may provide substantially better results. Once an ET has been properly detected, we can see that the model is doing a good job of accurately classifying it as an AEP or Non-AEP.

Chapter 7

Future Research

In this research, HMMs serve as the main model for classification and detection, which are trained on features derived from the complete data. Substantial research in speech recognition is being done in cascading HMMs with more HMMs, or other classifiers like SVM or MLP. These techniques would provide with good estimates of initialization, starting probabilities and other randomized parameters to the HMM. There is a huge potential for these techniques to provide more accurate results.

Currently, there are many different montages and channels being used to view the EEG. Also, the sampling frequency used in the industry is different in different places. The power (amplitude in μV) of different EEG monitors is also different, and can be adjusted by the individual preferences of the expert. These are the quantities which have an impact on the extracted features, and should be standardized across the industry. Many time series features other than LPC are being explored, and might be used.

A better definition of the exact shape, length and other properties of the EEG spike can be used to extract more dependable features. Also, a standardized database for all EEGs would be very helpful in evaluating all these algorithms and features in combination with one another. [11] Also, for the Non-ET data, expert provided exemplars of signals would be very helpful. An example dataset, having signals or artefacts which are recognized by the experts to be very similar to AEPs, but are not AEPs, need to be developed. Training models on such exemplars would surely provide better results. Apart from HMMs, other techniques being explored involve the use of Wavelets or Power Spectral Density to extract features from the EEG. Morphological similarities between an AEP spike

and the mother wavelet of DB4 are being exploited to characterize the spike. Also, techniques like Radial Basis Functions (RBF), SVM, and Deep Learning are being explored and may lead to more accurate results.

Bibliography

[1] N. Acir, I. Oztura, M. Kuntalp, B. Baklan, and C. Guzelis. Automatic detection of epileptiform events in eeg by a three-stage procedure based on artificial neural networks. *IEEE Transactions on Biomedical Engineering*, 52(1):30–40, Jan 2005.

[2] Bishnu S Atal and Suzanne L Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50(2B):637–655, 1971.

[3] Leonard E Baum and George Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968.

[4] Silvia Chiappa and Samy Bengio. HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. Technical report, IDIAP, 2003.

[5] Practical Cryptography. Linear prediction tutorial. `http://practicalcryptography.com/miscellaneous/machine-learning/linear-prediction-tutorial/`.

[6] R. S. Fisher, W. v. E Boas, W. Blume, C. Elger, P. Genton, P. Lee, and J. Engel. Epileptic seizures and epilepsy: Definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE) - fisher - 2005 - epilepsia - wiley online library. April 2005.

[7] Institute for Defense Analyses. Communications Research Division and John D Ferguson. *Symposium on the Application of Hidden Markov Models to Text and Speech*. Institute for Defense Analyses, Communications Research Division, 1980.

[8] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[9] Ana Isabel García-Moral, Rubén Solera-Ureña, Carmen Peláez-Moreno, and Fernando Díaz-de María. Data balancing for efficient training of hybrid ann/hmm automatic speech recognition systems. *IEEE Transactions on audio, speech, and language processing*, 19(3):468–481, 2011.

[10] John R Glover, N Raghaven, Periklis Y Ktonas, and JD Frost. Context-based automated detection of epileptogenic sharp transients in the EEG: elimination of false positives. *IEEE Transactions on Biomedical Engineering*, 36(5):519–527, 1989.

[11] Jonathan J Halford. Computerized epileptiform transient detection in the scalp electroencephalogram: Obstacles to progress and the example of computerized ecg interpretation. *Clinical Neurophysiology*, 120(11):1909–1915, 2009.

[12] 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.

[13] Leland B Jackson. *Digital Filters and Signal Processing: With MATLAB® Exercises*. Springer Science & Business Media, 2013.

[14] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1, 2007.

[15] John Makhoul. Linear prediction: A tutorial review. *The Journal of the Acoustical Society of America*, 62(S1), 1977.

[16] Schafer Patricia, O and Shirven Joseph, I. Epilepsy statistics — Epilepsy Foundation, October 2013.

[17] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. Eeg-based discrimination between imagination of right and left hand movement, December 1997.

[18] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[19] L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series: Advanced monographs. PTR Prentice Hall, 1993.

[20] EJ Smith. Introduction to EEG, 2014.

[21] C. M. Ting, S. King, S. H. Salleh, and A. K. Ariff. Discriminative tandem features for hmm-based eeg classification. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3957–3960, July 2013.

[22] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[23] Brian C. Waters, Chad G.and Dean and Jonathan J. MD Halford. EEGnet: A web platform for collaborative EEG research. In *Graduate Research and Discovery Symposium (GRADS)*, number 54. 2013.

[24] Jing Zhou. A study of automatic detection and classification of EEG Epileptiform Transients. `http://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=2276&context=all_dissertations`, August 2014.