1. The first part of the document is a list of names and titles, including "The Hon. Mr. Justice" and "The Hon. Mr. Justice".

Resolution test chart showing patterns of vertical and horizontal lines with numerical labels: 1.0, 1.1, 1.25, 1.4, 1.6, 1.8, 2.0, 2.2, 2.5, 2.8, 3.2, 3.6, 4.0, 4.5, 5.0, 5.6, 6.3, 7.1, 8.0, 9.0, 10.

PIONEERS IN METHYLENE BLUE TESTING SINCE 1974

18000 COUNTY ROAD 9, SUDBURY, ON N3B 2G7, CAN
TEL 513 426 7007 FAX 513 426 7007 Tlx 540000000

**National Library
of Canada**

**Bibliothèque nationale
du Canada**

Canadian Theses Service

Service des thèses canadiennes

**Ottawa, Canada
K1A 0N4**

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Neural Network Based Detection of EEG Abnormalities

by

Leon L. Lipoth, B.A.Sc.

**A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Engineering**

**Department of Systems and Computer Engineering,
Carleton University,
Ottawa, Ontario, Canada**

December 17, 1991

© Leon L. Lipoth 1991

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-76013-3

Canada

The undersigned hereby recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

Neural Network Based Detection of EEG Abnormalities

submitted by Leon L. Lipoth, B.A.Sc., in partial fulfillment of the requirements for the degree of Master of Engineering.

Chair,
Department of Systems and
Computer Engineering

Thesis Supervisor

Thesis Supervisor

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
December 17, 1991

Abstract

This thesis examines a neural network approach to detecting abnormalities in the Electroencephalogram (EEG). The approach is novel in its application of neural network processing to the quantified evaluation of EEGs. A Probabilistic Neural Network (PNN) architecture is emphasized. The advantages of the PNN include its quick training and easy modification of classification decision boundaries.

The PNN classifier is developed as described in Section 4.3. In addition, the tools and procedures of Sections 3.3 and 3.4 for the EEG data processing are developed. An improvement to the spectral averaging process is made by introducing multiple, overlapped windows to the sampled EEG data. This technique allows for a tradeoff between spectral variance and frequency resolution.

In the data processing, a normative reference set of EEGs is established by the examination of the time and frequency domain characteristics of the EEGs. The normative reference set consists of EEGs for which no abnormalities are detected. The remaining EEGs are considered abnormal. Using frequency domain feature vectors from six standard frequency bands ($\delta, \theta, \alpha, \beta_1, \beta_2$, and β_3), the PNN is trained on examples of both normal and abnormal EEGs. The most successful classification achieved is with the alpha band relative asymmetrical power feature vector. A 95% sensitivity yields a 40% specificity. These results are based on the 'leave-one-out' method of Section 5.1.2.

A need for the reformulation of the criteria used to build the normative reference set is identified. Tighter controls on the admission of an EEG to the normative reference set would result in better detection of EEG abnormalities. In addition, the need for more EEGs in the data base, less allowable artifact in the EEG epochs, and more exploration of features is identified. The PNN provides a powerful tool for the exploration of new features.

This thesis provides ideas for future research work with an emphasis on neural network applications in Section 5.5. The ideas include neural network based artifact recognition, feature definition, and Statistical Probability Mapping (SPM).

Acknowledgements

I'll begin my thanks at the National Defense Medical Center (NDMC). Thank you to Dr. Chris Skinner (AKA 'Doc') who made this thesis possible. His interest and knowledge were essential to the completion of this work. Another thank you to Norm Chow who spent many hours generating and organizing mountains of EEGs and brain maps for this project. A thank you also goes to Jason Rusmisl who developed a topographic map viewing program. The courtesies extended to me by the folks at NDMC are much appreciated.

My thanks next extend to Carleton University. Thank you to my supervisors, professors Hafez and Goubran. Professor Hafez sparked my interest in neural networks. The highest compliment I can pay him is that he reminds me of my father. Rafik Goubran gave me encouragement and confidence to proceed with this work. He is meticulous, courteous, and a pleasure to know. Thanks for all the help!

Also at Carleton are numerous others deserving of thanks. C.B. for rides, G.B. (Mr. Latex) for consulting, J.B. for chit-chats, E.D. for distractions, F.L. for nifty software, D.M. for jokes (bad ones!), J.R. for sharing resources, and G.K. for helping H. Of course, I'll not forget D.M., D.M., and V.G. for always being kind to me when I needed something from them. Although he didn't go to Carleton, I'd like to thank A.E. for telling me that "imagination is more important than knowledge." For anyone I've missed, I'd like to thank *.*..

Very helpful in my return to school was the funding from NSERC and from the Department of Systems and Computer Engineering. Thank you, very much.

People not at Carleton, but deserving of thanks include Janet and Russell Davis, Al and Mary Braun, Ron and Dusty Scott, my roommate John, and my other roommate, John. Thanks for making the past two years an enjoyable stay in Ottawa.

One special person who was at Carleton, is not now, is part of the reason I'm in Ottawa, and has made my stay in Ottawa enjoyable is H (AKA Hazel). She helped me indirectly with moral support and directly by criticizing (and can she be critical!) this document (with the exception of these acknowledgments). I value her efforts on all counts and am singling her out for special thanks. I consider myself very lucky (and persistent).

Finally, I would like to thank my family and friends from home for their support. The list is much too big to fit, even if only by initials. You are all a part of me, and I used many memories to get through the past two years. The list of people who came to visit include Jim and Tim, and my Mom and Dad. My Mom and Dad are always there to let me stand on their shoulders so that I may see farther. Thanks for the support.

I dedicate this work to my grandmother Mary Pilchak whose passing from this earth marked the beginning of my work at Carleton.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Overview	3
2	The Routine EEG	5
2.1	The International 10–20 Electrode Placement	5
2.2	The Reference Electrode and Montages	6
2.3	The Clinical EEG Recording Session	11
2.4	Physiological Artifacts	13
2.5	EEG Interpretation and Abnormalities	15
3	Quantitative EEG Analysis	17
3.1	EEG Frequency Domain Analysis	17
3.1.1	Spectral Analysis and Features of the EEG	18
3.1.2	Topographic Mapping and the Detection of Abnormalities	21
3.1.3	Related Research	24
3.2	Data Acquisition	27
3.3	Data Processing	30
3.4	Estimation of Background Brain Activity	31
3.4.1	BrainLab’s Technique	32
3.4.2	The Welch Method of Spectral Estimation	33
3.4.3	The Final PSD Estimate	38
4	Neural Networks as Pattern Classifiers	40
4.1	Overview	40
4.2	The Basics of Neural Networks	43
4.2.1	Processing Elements	43
4.2.2	Network Topology	44
4.2.3	Memory and Learning	45
4.2.4	The Backpropagation Neural Network	47
4.3	The Probabilistic Neural Network	50
4.4	Related Research	57

5	The PNN Based Detection of EEG Abnormalities	59
5.1	The PNN Classifier	59
5.1.1	Training Files and Feature Selection	60
5.1.2	Assessing the Classification Performance	60
5.2	The PNN Apparent Performance Measures	65
5.2.1	Simulation Set 1	65
5.2.2	Simulation Sets 2 and 3	69
5.3	The PNN True Performance Measures	70
5.4	The PNN Classification Results	78
5.5	Improvements to the Research Methodology	81
5.5.1	EEG Data Processing	81
5.5.2	EEG Feature Selection	82
5.5.3	Improving the PNN Classifier	85
6	Conclusions	86
6.1	Research Summary	86
6.2	Further Research	91
A	EEG Artifact Identification.	98
B	EEG Data Processing - <i>M-files</i> and <i>Turbo C</i> programs.	106
C	The PNN <i>Turbo C</i> Program.	120
D	Simulation Set 1 - Apparent Performance Graphs	138
E	Simulation Set 2 - Apparent Performance Graphs	144

List of Figures

2.1	The International 10-20 Electrode Placement.	7
2.2	A Referential Montage.	9
2.3	A Transverse or Coronal Montage.	9
2.4	The Double Banana Montage.	10
3.1	An EEG Recording and its Spectral Density.	20
3.2	A Topographical Brain Map.	22
3.3	Statistical Probability Mapping.	23
3.4	Data Collection Epochs.	28
3.5	Epochs 14 and 15 as Recorded from a Typical EEG.	28
3.6	Data Sampling Block Diagram.	29
3.7	A Brain Activity PSD Estimate as Calculated by BrainLab.	33
3.8	Examples of Kaiser Windows.	34
3.9	Multiple Kaiser Windows in an Epoch.	36
3.10	Spectral Estimates Based on Varying the Number of Windows.	37
4.1	Classification using a Neural Network.	42
4.2	A Training Example with Two Classes.	43
4.3	A Generic Processing Element.	44
4.4	Transfer Functions used in Processing Elements.	45
4.5	A Simple Neural Network.	46
4.6	The Basic Backpropagation Neural Network (BPNN).	47
4.7	The BPNN as a Classifier.	49
4.8	A PDF Estimation Example, $\sigma = 0.1$	53
4.9	A PDF Estimation Example, $\sigma = 0.5$	53
4.10	A PDF Estimation Example, $\sigma = 3.0$	54
4.11	A Probabilistic Neural Network.	55
5.1	An 8-45-2 PNN Topology.	61
5.2	Delta Band Relative Asymmetrical Power Feature Vector.	62
5.3	Construction of Delta Band Training Files.	63
5.4	Classification Results for an 8-45-2 PNN.	64

5.5	Apparent Performance Graph - PNN with $G = 10$, $F = 80$	66
5.6	Apparent Performance Graph - Effect of Increasing F	67
5.7	Apparent Performance Graph - Effect of Decreasing G	67
5.8	The Effect of Varying the Smoothing Parameter.	68
5.9	The Effect of Increasing the Number of Training Vectors.	71
5.10	PNN Trained on All Available Feature Vectors.	72
5.11	The PNN Delta Band Classification.	73
5.12	The Optimal Delta Band Smoothing Parameter.	73
5.13	The PNN Delta Band Classification Results.	75
5.14	The PNN Theta Band Classification Results.	75
5.15	The PNN Alpha Band Classification Results.	76
5.16	The PNN Beta1 Band Classification Results.	76
5.17	The PNN Beta2 Band Classification Results.	77
5.18	The PNN Beta3 Band Classification Results.	77
5.19	PNN Classifier for a Single Electrode.	84
D.1	Apparent Performance Graph - Training File T1.	139
D.2	Apparent Performance Graph - Training File T2.	139
D.3	Apparent Performance Graph - Training File T3.	140
D.4	Apparent Performance Graph - Training File T4.	140
D.5	Apparent Performance Graph - Training File T5.	141
D.6	Apparent Performance Graph - Training File T6.	141
D.7	Apparent Performance Graph - Training File T7.	142
D.8	Apparent Performance Graph - Training File T8.	142
D.9	Apparent Performance Graph - Training File T9.	143
E.1	Apparent Performance Graph - Training File T1B.	145
E.2	Apparent Performance Graph - Training File T2B.	145
E.3	Apparent Performance Graph - Training File T3B.	146
E.4	Apparent Performance Graph - Training File T4B.	146
E.5	Apparent Performance Graph - Training File T5B.	147
E.6	Apparent Performance Graph - Training File T6B.	147
E.7	Apparent Performance Graph - Training File T7B.	148
E.8	Apparent Performance Graph - Training File T8B.	148
E.9	Apparent Performance Graph - Training File T9B.	149

List of Tables

2.1	Outline of a Routine EEG Procedure.	12
3.1	Standard EEG Frequency Bands.	18
5.1	A Summary of PNN Classifier Results.	78

Glossary

ANN Artificial Neural Network.
ANS Artificial Neural System.
BEAM Brain Electrical Activity Mapping.
BPNN Backpropagation Neural Network.
CT Computed Tomography.
DFT Discrete Fourier Transform.
ECG Electrocardiogram.
EEG Electroencephalogram.
EMI Electromagnetic Interference.
FFT Fast Fourier Transform.
MRI Magnetic Resonance Imaging.
NDMC National Defense Medical Center.
PE Processing Element.
PET Positron Emission Topography.
PNN Probabilistic Neural Network.
PSD Power Spectral Density.
qEEG Quantified EEG.
SPM Statistical Probability Mapping.

Chapter 1

Introduction

1.1 Motivation

The *electroencephalogram* (EEG) is an important diagnostic tool in the detection and identification of neurological disorders. Traditionally, the clinical electroencephalographer examines the multiple waveforms in search of abnormalities indicative of cerebral disease. Because the EEG provides a wealth of information, it is a challenging problem, even for the experienced encephalographer, to extract the diagnostically relevant waveforms [1, 12, 20, 34, 35]. Thus, a significant area of research is the computer aided evaluation of EEGs.

One increasingly popular computerized tool and area of flourishing research is known as *Brain Electrical Activity Mapping* (BEAM) or, simply, *brain mapping* [12, 20, 38]. Brain mapping represents a *quantified EEG* (qEEG) as a topographic map projected on to the scalp. The mapped feature may be an averaged amplitude, a spectral power level, or a statistical measure. The topographic map presents a picture of the EEG feature over a given interval of time. Ideally, the clinician can use these maps to differentiate between normal and abnormal brain electrical activity. As well, once an abnormality has been detected, the maps may be further useful in the diagnosis of the specific pathological condition causing the abnormality.

Promising results have been achieved in distinguishing between normal and pathological activity by utilizing various brain mapping techniques [9, 26, 34, 35,

37, 40]. However, the analysis of these brain maps is based on visual inspection by the electroencephalographer. This leaves a high degree of subjectivity in the map interpretation. Thus, to reduce the subjectivity in map interpretation and to increase the uniformity of assessment of EEGs, an automated solution is desirable.

One possible solution to the computer based analysis of EEGs is the application of an *Artificial Neural Network* (ANN). Recently, much work has concentrated on applying neural networks as pattern classifiers in real world problems including medical diagnosis [4, 8, 17, 23, 24, 48, 49]. In essence, a neural network attempts to mechanize human information processing capabilities [50]. A neural network approach to the analysis of brain maps would strive to mimic the ability of electroencephalographers to interpret maps.

It is important to note that both brain mapping and artificial neural systems are, at present, highly exploratory areas. Thus, the merging of these two paradigms has only recently become possible. Each has identifiable problems which are discussed where necessary throughout the ensuing chapters.

1.2 Objectives

As mentioned in the previous section, the first step in the clinical assessment of EEGs is the detection of abnormal cerebral activity. Restated, this first step involves the gross classification of an EEG as being either 'normal' or 'abnormal'.¹ The main purpose of this thesis is to apply an appropriate neural network to this classification problem.

In applying a neural network to the gross classification of electroencephalograms, it is necessary to draw on experience from the fields of qEEG and brain mapping. Thus, the objectives of this thesis are threefold:

1. To investigate the use and techniques of qEEG and brain mapping in the detection of abnormal cerebral activity. Furthermore, to identify and, if possible, implement improvements to these techniques.

¹An abnormal EEG is one which suggests the possibility of cerebral disease [29].

2. To select and develop a neural network architecture to be used in the detection of EEG abnormalities. Furthermore, to evaluate the performance of the network in the detection of abnormalities.
3. To identify future areas of research including important issues with qEEG and brain mapping, improvements to the selected neural network architecture, and other possible uses for neural networks in clinical EEG assessment.

The purpose of these objectives is to seek an understanding of the issues involved in quantifying and automating the detection of abnormal cerebral activity in the clinical EEG. In particular, emphasis is placed on a neural network based solution to this problem. A successful solution will provide two important benefits. Firstly, it will act as a filter allowing the clinician to concentrate on abnormal EEGs. Secondly, it will lay the foundation for further automated evaluation of the clinical EEG.

Outside the scope of these objectives is the problem of discriminating among pathological conditions once the abnormality has been detected. Rather, this work is intended to introduce the power of neural signal processing to the quantified evaluation of EEGs. As such, it proceeds by drawing from established research practices in both brain mapping and neural network systems.

1.3 Overview

Chapter 2 provides an overview of the routine EEG. The chapter provides an introduction to the routine EEG along with associated nomenclature. Electrode placement and EEG recording procedures in a clinical environment are described. As well, important physiological artifacts which interfere with the recording of brain electrical activity are introduced. Chapter 2 concludes with common terminology used in the interpretation and detection of abnormalities in the EEG.

The purpose of Chapter 3 is twofold: to introduce research in the analysis of the quantified EEG and to describe the EEG data available for this thesis. The first section investigates the methodology of quantitative EEG analysis for the detection of abnormalities and provides a review of related research. The second section

describes the EEG data collection facilities and data available for this study. As well, the methodology used in the processing of the EEG data collected for this study is discussed in the final two sections.

Chapter 4 describes neural networks used as pattern classifiers. The first section begins with an overview and example of the classification of EEGs. The second section describes the basic structure and elements of a neural network. The third section details the theory behind the Probabilistic Neural Network (PNN). This network is used extensively in Chapter 5. The fourth section presents examples of research using neural networks as pattern classifiers.

Chapter 5 summarizes the PNN classification of EEGs based on abnormalities detected from EEG features. The first section provides the background needed to understand and discuss the performance of the PNN classifier. The second and third sections present research results from various PNN topologies. The fourth section is an interpretation of these results. The last section discusses improvements to the research methodology with an emphasis on neural network applications.

Chapter 6 concludes with a research summary and recommendations.

Chapter 2

The Routine EEG

This chapter provides background information on the routine EEG as recorded in a clinical environment. The first two sections describe the standard placement of electrodes and introduce associated nomenclature. The third section provides a typical example of the procedure for recording an EEG. This is followed by a description of important electrophysiological processes which interfere with the EEG recording. Finally, the clinical analysis of the EEG is introduced along with examples of abnormalities which can be detected.

2.1 The International 10–20 Electrode Placement

The purpose of an EEG is to measure the electrical activity of the brain. In clinical electroencephalography, this measurement is taken by the application of electrodes to the surface of the head. In general, the placement of these electrodes is according to a standard known as the international 10-20 system [25]. This system attempts to place electrodes over particular brain areas (frontal, parietal, etc.) independent of skull size. The 10-20 system was adopted by the IFSECN¹ in 1958 to facilitate the comparison of EEG records taken in different laboratories [7].

The electrode placement for the 10-20 system is shown in Figure 2.1. In practice, anterior to posterior locations are found by measuring from the nasion to the inion

¹International Federation of Societies for EEG and Clinical Neurophysiology.

through the vertex along the central sagittal plane. Along this line, five points are determined: the frontal pole (Fp), the frontal (Fz), the central (Cz), the parietal (Pz), and the occipital (O). The 10-20 system gets its name from the steps used to derive the electrode locations. Point Fp is situated above the nasion at 10% of the nasion-inion distance with the remaining points Fz, Cz, Pz, and O at 20% increments. Note that only three electrodes are situated at Fz, Cz, and Pz. Points Fp and O are for reference only.

Lateral electrode locations are found by measuring from the left to right preauricular (at the base of the ear) points through the vertex along the central coronal plane. The left and right temporal points, T3 and T4, are 10% of this distance above the preauricular point (on the corresponding side). Similarly, the central points, C3 and C4, are marked an additional 20% of the distance above the temporal points.

The anterior to posterior electrode locations from the frontal to occipital poles, Fp to O, via the T3 (left hemisphere) and T4 (right hemisphere) locations use the same 10%-20% pattern. Using Fp as reference, the frontal poles (Fp1, Fp2) and the inferior frontal points (F7, F8) are determined. Similarly, using O as reference, the occipital poles (O1, O2) and the posterior temporal positions (T5, T6) are found.

The remaining mid-frontal (F3, F4) electrodes are placed midway along the coronal line from F7 to Fz and from F8 to Fz. Similarly, the mid-parietal (P3, P4) electrodes are located. With the two auricular electrodes (A1, A2), this provides for a total of 21 standard electrode positions. Additional positions include the naso-pharyngeal electrodes (Pg1, Pg2).

Finally, it is worth observing that left and right hemisphere electrodes have odd and even numbers as subscripts, respectively.² As well, the numbers are selected to allow for intermediate positions (ie. F2, F6, F1, F5, etc.).

2.2 The Reference Electrode and Montages

The EEG is a record of the changing electrical field over the surface of the scalp. In practice, the output of a differential amplifier is used to record the difference

²In the case of the 3 midline electrodes Fz, Cz, and Pz the 'z' subscript is for zero.

Figure 2.1: The International 10-20 Electrode Placement.

in potential between two input electrodes. The voltage recorded depends on the locations of both electrodes. In general, electrode pairs in close proximity are said to be *bipolar*; while electrode pairs at relatively distant proximity are said to be *monopolar*. The practical distinction is that a closely spaced pair is sensitive to location and orientation of nearby electrical generators; while a widely separated pair is relatively insensitive. However, in actual fact there is no location on the

body that can be regarded as being at electrical infinity for all possible generators [45]. Thus, the EEG waveform depends on the choice of reference location.

There are three main choices for reference selection:

1. A *common reference* where the potential at each electrode is measured with respect to a common reference point.
2. A *common average reference* where the common reference consists of the average of a group of electrodes.
3. A *bipolar derivation* where there is no common reference. Instead, only local potential differences between pairs of adjacent electrodes are measured.

Each of these methods occasionally emphasizes specific features of the EEG in a more obvious manner [7]. Thus, a routine EEG generally consists of multiple combinations of sequences of electrodes and references. Each such combination is referred to as a *montage*.

Using the 10-20 system described in the previous section, many rational montages exist. Three popular montages are shown in Figures 2.2, 2.3, and 2.4. Figure 2.2 shows a *referential montage* which has a common reference (the ipsilateral ear) for each hemisphere. The midline electrodes are arbitrarily referenced to the left ear.

Figure 2.3 shows a *transverse* or *coronal montage* which utilizes a bipolar derivation. Also using a bipolar derivation is the *double banana montage* of Figure 2.4.

Figure 2.2: A Referential Montage.

Figure 2.3: A Transverse or Coronal Montage.

Figure 2.4: The Double Banana Montage.

2.3 The Clinical EEG Recording Session

Among many factors, the electrical activity of the brain is highly dependent on two major variables: [21]

1. The level of alertness of the individual.
2. The presence of external stimuli (ie. auditory, visual, etc.).

As such, these variables must be carefully controlled during the recording of an EEG.

The environment for a routine EEG typically consists of a subject comfortably reclined in a dimly lit, specially constructed, soundproof, EMI³ shielded room. The traditional EEG recorder, a multichannel electromechanical pen and paper based system, along with the facilities to digitally record the EEG are located in a separate room. The subject is instructed to minimize body movement, especially eye and head motion. This is to reduce the amount of eye and muscle *artifact* (refer to section 2.4). Ideally, the subject is motionless (except for respiration) and in a relaxed but alert state.

In all EEG departments, there is typically a routine recording procedure which defines the sequence of montages, the use of eye closure and opening, visual stimulation, auditory stimulation, and activation techniques used in the EEG recording session. A routine procedure is necessary for making comparisons with other patients as well as with subsequent recordings of the same patient. An example of a routine procedure is as depicted in Table 2.1.⁴

Referring to Table 2.1, a recording procedure begins with a calibration run of the equipment. A check of amplifier gains, filter settings, noise levels, pen alignment, and paper speed is performed. Any corrective action (ie. replacing a bad electrode contact) is taken at this time. After successful calibration, a standard sequence of montages is followed. In the first montage, the objective is to obtain an overall view

³Electromagnetic Interference.

⁴This is the procedure used in the Neurophysiology Department at the National Defence Medical Center, NDMC, in Ottawa, Canada.

Montage	Eyes	Stimulus	Duration
Calibration	Open	None	1 min
Referential	Closed/Open	None	30s/15s
	Closed/Open	None	30s/15s
	Closed/Open	None	30s/15s
	Closed	None	3.5 min
Coronal	Closed/Open	None	30s/15s
	Closed/Open	None	30s/15s
	Closed/Open	None	30s/15s
	Closed	None	3 min
Double Banana	Closed	Hyperventilation	3.5-5 min
	Closed	None (Post HV)	3 min
	Closed	6 Hz Photic/None	10s/10s
	Closed	8 Hz Photic/None	10s/10s
	Closed	10Hz Photic/None	10s/10s
	Closed	⋮	
	Closed	20Hz Photic/None	10s/10s
Calibration	Open	None	1 min

Table 2.1: Outline of a Routine EEG Procedure.

of the type of activity present. For this purpose a common reference montage, such as the referential montage of Figure 2.2, is utilized. A common reference is used over a bipolar derivation because it provides a better overall view of brain activity [7]. Note that the subject is instructed (by the supervising technician) to open or close his/her eyes at predetermined time intervals.

Subsequent montages are usually of bipolar derivation to isolate the location of any abnormalities. The second montage in the procedure of Table 2.1 is a coronal run in which the eyes are once again open/closed for specified intervals. The final montage, double banana, is used in conjunction with two activation procedures: *hyperventilation* and *photic stimulation*.

In hyperventilation activation the subject is instructed to overbreathe for a period of 3–5 minutes. The objective is to reduce the blood carbon-dioxide level in order to cause blood vessel constriction and reduce cerebral blood flow. After hyperventilation ceases, normal breathing is resumed and the recording is continued for an additional 3 minutes.

With photic stimulation, a stroboscope is used in an attempt to evoke a response at the flash frequency. For the procedure in Table 2.1, the flash frequency is varied from 6–20 Hz in increments of 2 Hz with a 10 s break between stimuli. Note that even though the eyes are closed, the flash has sufficient intensity (around 0.5 J) to evoke a response.

There are many variations in the recording of an EEG. For example, much work exists in the sleep recording of EEG [42, 3, 13]. Sleep records often reveal abnormalities not detected in alert state recording. As well, drug induced activations is an area of significant study called *pharmac-EEG*.

2.4 Physiological Artifacts

The EEG records the summed activity of multiple physiological sources. The electrical potentials generated by these sources follow the principle of superposition [7]. Potentials from physiological sources other than the brain are termed *artifacts* because they obscure the true brain activity. Four major contributors to artifact include: eye movement, cardiac activity, muscle activity, and sleep.

Eye movements commonly produce frontal low frequency activity which can easily be an order of magnitude greater than cerebral activity. When the eyes are stationary, there is a standing potential difference (5–10 mV) between the cornea and retina. When the eyes or eyelids move, a new potential gradient is induced. With

the international 10-20 system of electrode placement, the frontal polar electrodes (Fp1 and Fp2) and the frontal electrodes (F7, F3, Fz, F4, and F8) are most severely affected.

Cardiac activity includes pulse artifact and direct pick-up of the *electrocardiogram* (ECG). Pulse artifact is caused by variation of electrode resistance in the neighborhood of a pulsating artery. ECG pick-up appears as a repetitive spike in synchronization with the heart beat. As there is always some ECG artifact, it is common to record the electrocardiogram on a spare EEG channel. This provides a reference to aid in the detection of ECG artifact.

Muscle activity generates surface potentials in the scalp and neck. Muscle activity includes voluntary movements, jaw clenching, and respiration. Artifact first appears temporally (T3, T5, T4, and T6) but can affect any electrode location [30]. Another common practice is to record both the respiration rate and depth on a spare channel. This is not only useful for artifact detection, but also for monitoring hyperventilation activation procedures.

Sleep can be considered an artifact when the objective is to study the alert state brain activity. Sleep produces diffuse changes in brain activity. If drowsiness is not recognized, it may be confused with some form of pathological condition. As mentioned in the previous section, various levels of sleep are occasionally induced in activation procedures. At such times, sleep is not an artifact, but a desired state of consciousness.

In the interpretation of the EEG, the identification of the various physiological artifacts is extremely important [10, 28, 33, 46, 54]. To assist in artifact recognition, observations on the subject are recorded throughout the EEG. For example, obvious movements such as swallowing, coughing, sneezing, or talking are recorded on the paper EEG at the time of occurrence. As well, minimization of physiological artifacts is attempted by ensuring that the subject is comfortable and relaxed. The goal is to obtain a high proportion of artifact free EEG for interpretation of the underlying brain activity.

2.5 EEG Interpretation and Abnormalities

The interpretation of EEGs involves the recognition of normal patterns and the detection of abnormal cerebral activity. Even in the absence of artifact, this process is complicated by brain activity variations due to age, metabolism, mood, drugs, and a host of other determinants [21, 31]. It is under these conditions that the clinician must search for abnormalities in the EEG.

From a clinical viewpoint, an abnormal EEG is one which suggests the possibility of cerebral disease [29]. In practice, abnormalities are detected by the visual comparison of brain activity in healthy subjects of similar age and in a similar state of alertness. Abnormalities can be classified temporally as *paroxysmal* or *non-paroxysmal*, and topographically as *focal* or *generalized*.

A paroxysmal event occurs with an abrupt onset, has rapid attainment of maximal activity, and terminates suddenly. A non-paroxysmal event is more continuous in nature; it does not begin abruptly nor terminate suddenly. A focal abnormality is one which occurs in a localized space, for example over 3–4 adjacent electrode locations in the international 10-20 system; whereas a generalized abnormality is spread over both hemispheres. Furthermore, generalized activity may be *asynchronous* or *bilaterally synchronous*.⁵

Examples of non-paroxysmal focalized abnormalities can include lesions, tumors, contusions, or a lack of local blood supply [22]. Such conditions can reduce the brain activity in the surrounding area. This forms a *depression* which, when clearly confined to a restricted area, is a sign of a focal pathology. As well, the same conditions can cause an increase in activity around a given area. When such activity has a frequency content below the *alpha rhythm* (8–12 Hz), it is called *focal slowing*. In general, the lower the frequency at which slowing occurs, the more severe is the disturbance [22].

The ability of a focal abnormality in the EEG to isolate the problem area is known as *localization*. Successful localization is not assured, but has improved with the use of topographical mapping [40, 9], the topic of the next chapter.

⁵The term 'synchrony' in EEG is defined as two regions of activity with nearly the same potential change as a function of time and are nearly in phase or nearly 180° out of phase [16].

For paroxysmal focal activity the pathological basis is, in general, epilepsy. Epileptic disturbances consist of excessive discharges of neural populations. Such an event is known as *ictal* activity and is commonly referred to as a seizure. A focal ictal discharge provides a fair localizing ability [22].

Epileptic disturbances not only need be confined to focalized areas, but also can be manifest as generalized bilateral activity. In such cases, the activity usually occurs over both hemispheres in homologous areas and is synchronous [16]. However, if the activity is asynchronous, the activity must be distinguished from multiple focal paroxysmal disturbances. This is usually a difficult problem but, in either case, both are examples of foundation for an abnormal EEG.

Finally, generalized non-paroxysmal activity is perhaps the least well understood in terms of being able to associate any functional or cerebral diseases with the abnormality [29]. For example, Disturbed metabolism can cause generalized non-paroxysmal activity, but can not account for all forms of such abnormalities [29].

Chapter 3

Quantitative EEG Analysis

This chapter is divided into four sections. The first section presents an introduction to spectral analysis of the EEG, *topographic brain activity mapping*, and related research. This section investigates the methodology of quantitative EEG analysis. The second section describes the EEG data collection facilities and the data available for this study. The third section describes the process of determining two categories of subjects: NORMAL and ABNORMAL. These categories are formed based on abnormalities detected in the EEG. The last section details the procedure of estimating *background brain activity* as introduced in Section 3.1.1. Existing procedures are described and improvements are suggested and adopted in this thesis.

3.1 EEG Frequency Domain Analysis

This section begins with an introduction to the spectral analysis of the EEG. Customary EEG frequency bands are defined and examples of common spectral features are given. Section 3.1.2 introduces *topographic brain activity mapping* and describes its role in the detection of abnormalities in the EEG. Using the background provided by these first two sections, the Section 3.1.3 describes research related to the detection of EEG abnormalities.

3.1.1 Spectral Analysis and Features of the EEG

The traditional first step in the visual analysis of the raw EEG (the polygraphic voltage *vs* time recording) is to search for data discontinuities such as ictal activity and artifacts [10]. After such activity has been examined, the next step is to inspect the *background brain activity*. This process involves an estimation of the frequency content of the EEG waveforms. The frequency content is, in general, banded into customary bands of *delta*, *theta*, *alpha*, and *beta* as depicted in Table 3.1. Frequencies below the alpha range are referred to as *slow waves*, while frequencies above the alpha range are referred to as *fast waves*.

In the traditional first step, the visual extraction of data discontinuities from the raw EEG has been very successful in the detection of abnormalities caused by epileptic activities; however, in the next stage, the visual estimation of background activity frequency components has led to limited successes in the detection of other abnormalities [10, 18]. This is primarily due to the subjective nature of visual inspection and the complexity of the EEG. Thus, the limited analytic ability of unaided visual inspection has led to the quantification of the EEG. To improve on the extraction of frequency components in EEG background activity, spectral analysis is employed. The most extensively employed spectral analysis technique is

EEG Frequency Bands			
Band	Symbol	Range (Hz)	Comments
Delta	δ	0 - 4	Slow waves
Theta	θ	4 - 8	
Alpha	α	8 - 12	
Beta1	β_1	12 - 16	Fast waves
Beta2	β_2	16 - 20	
Beta3	β_3	20 - 24	

Table 3.1: Standard EEG Frequency Bands.

based on the *fast Fourier transform*, FFT [5]. The FFT is an efficient algorithm for the computation of the *discrete Fourier transform*, DFT. The DFT of a finite data sequence, $x[n]$, of length N is

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi kn/N)} \quad k = 0,1,2,3 \dots N-1.$$

From the DFT coefficients an estimate of the *power spectral density*¹, PSD can be formed by using the squared magnitude of each coefficient, $X[k]$.

Figure 3.1 illustrates the EEG data for one electrode and the corresponding PSD as derived from the FFT computation. This technique summarizes the frequency content of the data over the collection time period or *epoch*. The resolution of the FFT and the corresponding PSD is determined by the inverse of the epoch length. In this example, the epoch length is 4 seconds; therefore, the resolution is 0.25 Hz.

The PSD is the basis for determining various frequency domain features of the EEG. For example, by summing the spectral density values from 0 to 4 Hz the absolute power in the delta band is obtained. Similarly, the absolute power for other bands such as those in Table 3.1 can be calculated. Typically, the results from multiple epochs are averaged to represent a band specific spectral content. This averaging process yields commonly utilized features such as:

1. The *absolute power* in a specific band (ie. delta) is determined by averaging the power calculated in each epoch.
2. The *relative power* is the ratio formed by dividing the absolute power from a specific band by the total power from all bands.
3. The *instability of relative power* is the ratio of the standard deviation of the relative power to the mean relative power.
4. The *relative asymmetrical power* is the difference between the power at symmetrical electrodes (ie. Fp1 & Fp2) divided by the total power.

¹The term *power spectrum* is often used in place of the more accurate *power spectral density*. In either case, the units are $\mu V^2/Hz$ when referring to EEG power density levels.

Figure 3.1: An EEG Recording and its Spectral Density.

In addition to a host of other features, it is also possible to apply mathematical transformations (ie. logarithms, ratios, roots, etc.) to these spectral parameters [15]. Such transformations are usually done in conjunction with *statistical probability mapping*, SPM, discussed in the next section.

One of the problems with spectral analysis involves the stability or *stationarity* of the EEG. Changes in attention and levels of alertness cause shifts in the background EEG activity [28]. Because spectral analysis is sensitive to changes in stationarity, it is important to monitor the subject and be certain that behavioral instructions (ie. stay alert, eyes closed) are followed. At present, an open question is the total time segment of data needed to be averaged in order to obtain a representative sample of cerebral electrical activity [46]. However, most studies use epoch lengths of 2–5 seconds with a total of 30–120 seconds of averaged data for the final power spectral density estimate.

3.1.2 Topographic Mapping and the Detection of Abnormalities

After computing a spectral feature for each electrode, the natural next step is to present both spectral and spatial information as a single image. Figure 3.2 is an example of such a *brain map*. This map shows the distribution of the absolute power in the theta band. Also, the PSD for the T3 and T4 electrodes are shown. The theta band power² for the T3 electrode is $13.23 \mu V^2$ and for the T4 electrode is $58.31 \mu V^2$. The vertical scale represents the power from 0 to $81.85 \mu V^2$. The map demonstrates a severe right sided asymmetry indicative of a focal abnormality. The map is made from only 19 real data points corresponding to the electrode locations. The remaining points are calculated by *triangular interpolation*³ and *smoothing*⁴.

²The electrode 'power' is given as μV^2 . The real power can be computed by using the nominal reference value of $5 k\Omega$.

³Triangular interpolation uses the inverse of the distance from the three nearest electrodes to weight and sum the corresponding values.

⁴Smoothing replaces each picture element by the mean of its eight nearest neighbours and the value at the picture element itself.

Figure 3.2: A Topographical Brain Map.

The method of interpolation as well as the number of grey (or color) scale levels remain open to further research [54, 33, 10, 1].

In essence, the clinical application of topographic mapping is concerned with the detection and localization of abnormalities. There are three main approaches:

1. Unaided Visual Inspection.
2. Significance Probability Mapping.
3. Neurometrics.

Unaided visual inspection of topographical brain maps is a logical and necessary extension of classic EEG interpretation. It is logical in that the analytic power of the human visual system is already used in the interpretation of the raw EEG; it is necessary in that the clinician must learn how time domain activity maps to the frequency domain. For example, recognition of artifact [30] and state changes [3, 13] is as important in the frequency domain as in the time domain.

Figure 3.3: Statistical Probability Mapping.

The main problem with unaided visual analysis is in discriminating between normal and abnormal variations. For example, symmetrical brain activity is normal, but some asymmetry is acceptable. Deciding when a perceived asymmetry is abnormal is often a difficult task [10]. Thus, to assist in this decision, *significance probability mapping* [11], SPM, has been developed.

Figure 3.3 illustrates the SPM transformation from a topographical feature map to a *Z-Statistic* map. This transformation replaces, on an electrode by electrode basis, the original feature with the number of standard deviations, z , by which the observation differs from a *normative reference* or *control* group. The clinician can then search the map for significant deviations from the normative reference. Because this technique assumes a *Gaussian* distribution of the feature (at each electrode), it is often necessary to apply suitable mathematical transformations [15] to achieve a Gaussian shape. The Gaussian distribution is defined in Section 4.3.

The success of the process depends on the formation of a normative control group to be used as the reference. Two key issues in the selection of a normative

control include age and exclusionary criteria [10]. It is well established that cerebral activity varies dramatically as a function of age. As such, it is unreasonable to compare the brain activity of a child to that of an infant, or an adult to that of a child. Therefore, age appropriate samples must be available.

Exclusionary criteria refers to the level of control over admission of a subject as a 'normal'. For example, one criterion may define a 'normal' as a subject having no evidence of neurological or physiological disorders based on screening examinations; whereas another criteria may additionally require that there are no physical (ie. abnormal vision) disorders.

Related closely to exclusionary criteria are the classification performance measures of *sensitivity* and *specificity*. These measures are used to evaluate the performance of a classifier. Sensitivity refers to the ratio of individuals correctly classified as abnormal to the true number of abnormals. Specificity refers to the ratio of individuals correctly classified as normal to the true number of normals. Thus, if exclusionary criteria are too tight, then specificity is lost. Contrarily, if exclusionary criteria are too loose, then sensitivity is lost.

The third approach in the detection of abnormalities is a methodology known as *neurometrics* [27]. Neurometrics attempts to automate the clinical interpretation of EEGs. A *discriminate function* is derived and used to place the individual into one of a set number of diagnostic categories. Such a function is always intended to classify between very specific disorders and a normal population. The main problem with this approach is that there are virtually an infinite number of clinical diagnoses, but a limited number of proven discriminant functions [10].

3.1.3 Related Research

With the available technology, computerized topographical mapping of brain activity began to flourish in the late 1970s. In 1979, Duffy et al. [12] published their first article on the *Brain Electrical Activity Mapping*, BEAM, technique. Early work was primarily concerned with interpolation algorithms and display formats. In 1982 Coppola, Buchbaum, and Rigal [6] were quick to emphasize the ease of visualizing EEG data. A year later, Persson and Hjorth [43] described the topographical map

as a supplement to the conventional EEG and as beneficial in communicating EEG findings.

In an effort to demonstrate the clinical utility of topographical maps, Significance Probability Mapping, SPM, was adapted to BEAM by Duffy et al. [11]. In [11] 11 of 12 subjects with brain tumors were correctly identified from a normative control group size of 18 subjects. The one tumor not identified by SPM was a small parietal lesion in a patient with a normal EEG. Thus, SPM was demonstrated to be of use in the detection of physical lesions.

Similar studies illustrating the clinical utility of brain mapping soon followed. In [35] Morstyn et al. discovered significant differences in the spectral energy distribution between schizophrenic and control subjects. Differences included increased spectral energy in delta, theta, and beta3 energy bands. These results were consistent with investigations using other methods such as *positron emission topography* (PET) scans and *cerebral blood flow studies*. Such techniques are invasive to the subject and, in general, more expensive than an EEG. Thus the primary advantage of topographical mapping over these techniques is that it is a non-invasive procedure which provides quantitative measures of brain function.

In [37] Nuwer used topographical mapping in the detection of epileptic abnormalities. He showed improved sensitivity of mapping where other tests indicated 'normal' results. Other tests included PET scans, *computed tomography* (CT) scans, and *magnetic resonance imaging* (MRI). An introduction to these techniques can be found in [45].

Panet-Raymond and Gotman [40] also evaluated the usefulness of EEG background activity in localizing epileptic brain regions. Using SPM, in 20 of 22 subjects there was significant ($|z| > 2$) delta asymmetry. Furthermore, in 17 of 20 cases the asymmetry correctly lateralized the hemisphere with the abnormality. The results of this study suggest that background activity can be used as a method of detecting and localizing epileptiform abnormalities.

A criticism of all the studies discussed so far is that they only address the sensitivity of the topographic SPM technique but not the specificity. That is, differences between the control group and the subjects with pathologies are clearly demon-

strated, but no comparison is made between other 'normal' individuals and the control group. Thus, the aforementioned studies fail to address the 'normal' interindividual variation which affects both sensitivity and specificity.

Most neurometric studies based on the qEEG do attempt to measure both sensitivity and specificity. In [27] John et al. summarized results obtained in the analysis of cerebrovascular disease and psychiatric disorders. There were four categories of cerebrovascular disease which were detected with an overall sensitivity of approximately 90%. These same discriminate functions had an overall specificity of 97% with a control size of 64 subjects. These results were not verified by applying the discriminant functions to an independent set of subjects. Visual evaluation of the raw EEG waveforms by two experienced electroencephalographers yielded only a 57% detection of abnormalities. The four way discriminant classification of psychiatric disorders had an overall sensitivity of approximately 79% and a specificity of approximately 77%. Both of these results were verified with an independent set of patients.

Dolisi et al. [9] applied neurometrics to detect abnormalities in patients with intracranial tumors. The tumors were confirmed by CT scans and, in most cases, surgical examinations. Using asymmetrical *univariate*⁵ features, a sensitivity of 60% and a specificity of 95% was achieved. This was improved by using asymmetrical *multivariate*⁶ features to achieve a sensitivity of 95% and a specificity of 98%. These results were not verified by applying the discriminant function to an independent set of subjects.

⁵A univariate feature is a single variable (ie. absolute δ power) as taken from each electrode.

⁶A multivariate feature combines 2 or more univariate features.

3.2 Data Acquisition

The facilities available to collect EEG data for this thesis are those located at the National Defence Medical Center, NDMC, in Ottawa, Canada. These facilities include a Nicolet Instruments model 1A98 EEG recorder, an 80386 processor based computer, and digital data collection and analysis software called *BrainLab* as supplied by Nicolet.

The model 1A98 recorder provides all of the hardware, including a 21 channel pen and paper tracing system, necessary to record the EEG. This recorder is programmable by the BrainLab software which can control amplifier settings, filter settings, and montage selection. The routine EEG data collection procedure at NDMC is as outlined previously in Table 2.1. The sampled data is recorded during the eyes closed, alert state in the last 3.5 minutes of the referential montage. In total, 80 seconds (s) of 19 channel EEG data is sampled and recorded as illustrated in Figure 3.4. The 80s of data consists of twenty 4s epochs separated by approximately 1.2s each. Figure 3.5 shows epochs number 14 and 15 as sampled and recorded from a 19 channel EEG.

Referring to Figure 3.6, each EEG channel has an antialiasing lowpass filter at 30 Hz with a total of approximately 55 dB/decade attenuation. BrainLab collects 8-bit quantized data with a sample rate of 256 samples/ s . The data is then digitally low pass filtered and down sampled by a factor of four to achieve a final sample rate of 64 samples/ s for data storage. Thus, for each 4s epoch, 19 channels consisting of 256 samples each are captured. A total of twenty such epochs are available from each subject for analysis.

771
78
68
70
61
77
78
75
772
74
64
74
62
70
74
78
72
62
72

X: 1 SEC/IN Y: 4794/IN
SENSITIVITY: 750/IN ANTIFOLDLEVEL: 2000

Figure 3.5: Epochs 14 and 15 as Recorded from a Typical EEG.

Figure 3.6: Data Sampling Block Diagram.

3.3 Data Processing

The purpose of the data processing is twofold: to establish a normative control group from which the detection of abnormalities will be based, and to quantify the EEG background activity. At the time of this study, there was available the EEG data as described in the previous section from 122 individuals. The subjects are all members or retired members of the Canadian Forces. The group consists of a mixture of 107 males and 15 females with a minimum age of 19, a maximum of 79, and a mean of 39 years old. The subjects are divided into two categories: **NORMALS** and **ABNORMALS** based on neurological disorders detected by a neurologist⁷ interpreting the entire routine EEG and the frequency domain topographical maps of the sampled EEG. These maps included the power in the six bands ($\delta, \theta, \alpha, \beta_1, \beta_2, \beta_3$) for each epoch as well as a summary of each power band over all of the epochs. If any abnormality is detected, the subject is placed in the **ABNORMAL** category; if no abnormalities are detected, the subject is placed in the **NORMAL** category.

In order to quantify the EEG background brain activity using spectral analysis, it is first necessary to remove epochs contaminated with artifact. Recall from the previous section that twenty epochs of 4s each are recorded for each subject. Each epoch was examined in the time domain tracing for the presence of artifact. The results are summarized in Appendix A. The most common artifacts were electrocardiographic and eye movement. In order to reduce the number of epochs lost due to eye artifact, a threshold function was defined to allow some presence of eye movement. The threshold function used is

$$(\delta_{Fp1} + \delta_{Fp2})/(\delta_{O1} + \delta_{O2}) > T$$

where: δ_{Fp1} is the delta power of electrode Fp1,
 δ_{Fp2} is the delta power of electrode Fp2,
 δ_{O1} is the delta power of electrode O1,
 δ_{O2} is the delta power of electrode O2,
 T is the threshold, a constant number.

⁷Dr. C. R. Skinner, Lieutenant-Colonel, of NDMC.

For this study, the threshold is set at 2.0. Thus, epochs where the eye movement exceeds this threshold function are discarded. Effectively, this implies that further analysis, such as the creation of a discriminant function, must be tolerant of some eye artifact.

To get a representative measure of the frequency domain background brain activity, a minimum of 7 with preferably 10 artifact free epochs are needed. This criteria is based on three factors. Firstly, as many epochs as possible are desired for spectral averaging and noise reduction; secondly, using fewer epochs mean a greater probability of the EEG signal remaining stationary; and, finally, using fewer epochs also means that a smaller number of individuals will be left out of the study. This last factor stems from the limited number, 20, of epochs available from each subject.

Using the above criteria, there are 32 subjects with too badly contaminated epochs to proceed with further analysis. The result is a **NORMAL** group size of 60 and an **ABNORMAL** group size of 30 subjects. In the **NORMAL** category the minimum age is 19, the maximum is 76, and the average is 36 years old. In the **ABNORMAL** category the minimum age is 19, the maximum is 79, and the mean is 44 years old.

For each subject the averaged spectral estimate of background brain activity is based on the first 10 artifact free epochs. For seven subjects, only 7, 8, or 9 epochs are used as there are not 10 artifact free epochs available. The process of deriving the spectral estimate of the background brain activity is detailed in the next section.

3.4 Estimation of Background Brain Activity

The following section investigates the spectral estimation of the frequency components in the background brain activity. The first section describes the estimation as performed by the BrainLab analysis software. The next section details improvements to this estimation process based on the work done by Welch [55]. The last section summarizes the final estimate of the power density spectrum, PSD, of the background brain activity.

3.4.1 BrainLab's Technique

In topographical brain mapping most estimates of the frequency content of background brain activity are based on *periodogram* analysis. This refers to the direct Fourier transformation of an entire epoch of the EEG signal as described previously in Section 3.1.1. Multiple transformations are then averaged to reduce random variations due to both noise and the estimation process itself [39]. This technique is known as *Bartlett's procedure*.

The Nicolet BrainLab analysis software is based on Bartlett's procedure. The software directly computes the FFT for each of the first 15 artifact free epochs and averages the results to obtain the final PSD estimate. Figure 3.7 illustrates the final PSD estimate for one electrode as calculated by BrainLab. Note that BrainLab zeros the spectral coefficients from 0–1 Hz inclusive [36]. This is to reduce the effect of *spectral leakage* from the 0 Hz or *dc* component of the EEG signal. That is, when the EEG signal has a non-zero mean, this is translated to an impulse at 0 Hz in the PSD. In turn, this obscures or *leaks* into the lower amplitude low frequency components of the PSD. When summing the power in the alpha band (0–4 Hz), BrainLab compensates for the zeroing of the 0–1 Hz coefficients by using a 4/3 multiplier for each coefficient from 1.25–3.75 Hz⁸.

In this thesis, an alternative technique which preserves the low frequency coefficients is used. This technique involves computing the mean of the sampled data, in this case 256 samples (an epoch) from one electrode, and subtracting the mean from the data before computing the FFT. Therefore, upon computation, the FFT coefficient for 0 Hz will be zero. In summary, this technique uses the sample mean as an estimate of the dc frequency component and subtracts it from the data to achieve better low frequency estimates [39]. The technique is used for all remaining FFT computations performed in this research.

Further examination and possible improvements to the BrainLab technique of estimating the PSD are important because the PSD is the basis for all frequency domain analysis of the EEG. As such the next section describes the use of the

⁸BrainLab has a 0.25 Hz frequency resolution as the FFT uses 256 samples over the entire 4s epoch.

Figure 3.7: A Brain Activity PSD Estimate as Calculated by BrainLab.

modified periodogram based on the work by Welch [55]. As will be seen, the Welch method of spectral estimation provides a tradeoff between frequency resolution and variance of the spectral estimate.

3.4.2 The Welch Method of Spectral Estimation

With the *modified periodogram* a *window* is applied to the data before it is Fourier transformed by the FFT. Figure 3.8 illustrates three forms of a *Kaiser* window. The Kaiser window is defined as

$$w[n] = \begin{cases} \frac{I_0[\beta(1-((n-\gamma)/\gamma)^2)^{1/2}]}{I_0(\beta)} & 0 \leq n \leq M, \\ 0 & \text{otherwise,} \end{cases}$$

where $\gamma = M/2$, ($L = M + 1$) is the length of the window, β is shape parameter, and $I_0(\cdot)$ is the zeroth-order modified Bessel function of the first kind. The Kaiser windows of Figure 3.8 have their shapes defined by the β parameters of 0, 3, and

Figure 3.8: Examples of Kaiser Windows.

8. Note that $\beta = 0$ corresponds to a *rectangular* window. The window length, L , is 256 which corresponds to an entire epoch ($4s \times 64 \text{ samples}/s$).

Application of a window to the data means that a point by point multiplication of the data within the window occurs

$$x_w[n] = w[n] \cdot x[n],$$

where $x_w[n]$ is the windowed data, $w[n]$ is the set of window coefficients, and $x[n]$ is the original sampled data. Any data which is outside of the window is ignored. For the modified periodogram the Fourier transform (via the FFT) is based on the windowed data.

By shortening the window length, L , and placing multiple windows over the epoch length, more windowed data segments are available for averaging the estimate of the PSD. Figure 3.9 has two such examples. In the first, three overlapping windows with $L = 128$ are placed in the epoch; the second has seven overlapping windows with $L = 64$. In each case, the length of the overlap is $L/2$. Thus, decreasing the window length, L , results in more data segments available for averaging which,

in turn, reduces the variability of the spectral estimate. The tradeoff, however, is in the reduction of frequency resolution. This reduction is in the same proportion as the reduction in window length. For example, with $L = 256$ (a 4s time span) the frequency resolution is 0.25 Hz; whereas, with $L = 128$ (a 2s time span) the frequency resolution is 0.5 Hz.

For this thesis, a comprehensive set of alternatives using the Welch method for spectral estimation was explored. The results can be concisely summarized by one example. The *Matlab* m-file ('psdex.m') used to generate the following example can be found in Appendix B. Figure 3.10 has the spectral estimate based on 1, 3, and 7 windows of an epoch of EEG data (256 samples) from one electrode. The positioning for the multiple windows is as shown in Figure 3.9. Referring to Figure 3.10, note the smoothing that occurs with the increase in the number of windows used for estimating the PSD. This corresponds to a reduction in the variance of the spectral estimate. Also, as the variance is reduced, the spectral peak near 10 Hz is reduced and broadened. This is caused by spectral leakage which worsens as the window size is decreased (as the number of windows is increased) [39].

Finally, the $L/2$ overlap is chosen because Welch showed it provides maximal reduction in the variability of the spectral estimate [55]. Greater overlap does not further reduce the variance because the windowed segments become less independent.

Figure 3.9: Multiple Kaiser Windows in an Epoch.

Figure 3.10: Spectral Estimates Based on Varying the Number of Windows.

3.4.3 The Final PSD Estimate

The final PSD estimate for each electrode consists of the average of the PSDs from the 7–10 artifact free epochs as described in Section 3.3. The PSD for each epoch is based on the modified periodogram using 3 Kaiser windows with $L = 128$, $\beta = 4$, and $L/2$ overlap. The window length of 128 samples results in a frequency resolution of 0.5 Hz for the spectral estimate. The modified periodogram with 3 Kaiser windows ($\beta = 4$) was chosen because it provides a reduction in the variance of the spectral density estimate while retaining sufficient resolution to identify spectral peaks.

Each EEG has 19 final PSD estimates (one for each electrode in the 10-20 system) computed as described above. To arrive at the PSD estimates from an individual's EEG, the following steps were used. A *Turbo C* program ('extract.c') extracts the sampled raw (time domain) EEG data from an individual's data file ('eegid.eg0') as generated by BrainLab. The PSD estimates are then calculated using a *MatLab* version 3.5e software package program on an 80386 processor based computer. The extracted EEG data (from 'eegid.eg0') is transferred into a *MatLab* readable file ('eegid.raw'). This file ('eegid.raw') is used as input to the *MatLab* *m-file* program ('psd.m') which calculates the PSD for each electrode in each epoch. The resulting PSDs are stored in a *MatLab* readable file ('eegid.pow'). This file ('eegid.pow') is used as input to another *Turbo C* program ('estimate.c') which implements the artifact rejection and averaging process of Section 3.3 to obtain the final PSD estimates. The *Turbo C* programs and the *MatLab* *m-files* used can be found in Appendix B.

From the final PSD estimates, the frequency domain features for each EEG can be extracted. The final PSD estimates are used to calculate the averaged absolute power in each of the six standard frequency bands (δ , θ , α , β_1 , β_2 , and β_3) for each electrode. These averaged absolute powers are stored in a *MatLab* readable file ('eegid.est'). These features are to be used in the detection of abnormalities and the classification of the EEG into either the NORMAL or ABNORMAL categories as defined in Section 3.3. The method of classification investigated in this thesis is a neural network approach. In particular, a *Probabilistic Neural Network* is the main paradigm explored. Chapter 4 provides the background information on neural networks used as classifiers and describes the Probabilistic Neural Network. Chapter

5 presents the results obtained in applying the Probabilistic Neural Network to this EEG classification problem.

Chapter 4

Neural Networks as Pattern Classifiers

This chapter begins with an overview of neural networks used as pattern classifiers. The second section follows with a description of the basic structure and elements of a neural network. The process of network training and the *Backpropagation Neural Network* are introduced. The third section develops the theory behind the *Probabilistic Neural Network*. The Probabilistic Neural Network provides a general and powerful classification technique. Finally, the fourth section gives examples of research using neural networks as pattern classifiers.

4.1 Overview

Neural networks are mathematical models of theoretical mind and brain activity. Neural networks are also referred to as neural systems, artificial neural networks (ANNs), artificial neural systems (ANSs), connectionism, neurocomputers, and parallel distribution processors. Neural network models are characteristically based on the massively parallel processing and distributed representation of information that are believed to exist in the brain [50]. For example, the brain performs information transformations that operate in parallel on billions of highly interconnected neurons. As well, present theory suggests that information is stored in the

variable strength connections between neurons. As will be seen in the next section, these observations are incorporated in the mathematics of ANNs.

ANN theory is derived from multiple disciplines including biology, neuroscience, mathematics, psychology, physics, philosophy, engineering and computer science. The primary purpose of a neural system is to explore and reproduce human information processing tasks. In particular, neural systems are applicable in situations where only a few decisions are required from a massive amount of data and where a complex nonlinear mapping must be learned [50]. A short list of such applications includes data compression, system modelling, function approximation, near-optimal solutions to combinatorial optimization problems, and pattern recognition. The ability of a neural system to recognize or classify patterns is applied in this thesis.

Figure 4.1 outlines the general architecture for the use of a neural network as a classifier. Input or feature vectors are created from measurements made on objects of each class. For example, with an EEG (the object) the absolute power from four electrodes (ie. Fp1, Fp2, O1, and O2) can be used to form a 4-dimensional feature vector $\mathbf{x}$. This feature vector is then used as input to the neural network. The goal of the neural network is to classify the object into one of k classes. As shown in Figure 4.1, the class is indicated by a k -dimensional output vector $\mathbf{y}$. More specifically, if the classification decision is that the object belongs to class c ($1 \leq c \leq k$), then component y_c of the output vector $\mathbf{y}$ will be a logical true value (ie. a '1') and all other components of $\mathbf{y}$ will be a logical false value (ie. a '0').

In order for a neural network to make classification decisions, it must first be trained using feature vectors representative of each class. For example, consider the EEG classification problem in Figure 4.2. On the basis of a 4-dimensional feature vector, the EEG is to be classified into one of two categories: NORMAL or ABNORMAL. Representative examples of each class are used to create a NORMAL training set, T_n , with 10 members and an ABNORMAL training set, T_a , with 6 members. In the *training phase*, each of the training vectors are applied as inputs to the neural network and the corresponding desired output (the correct class) is also given. The neural network *learns* to associate the correct class with the training vector by utilizing a *learning algorithm*. The learning process is discussed in more detail in

Figure 4.1: Classification using a Neural Network.

Section 4.2.3.

Once the training phase has been completed, the knowledge attained by the neural network is tested. In the *testing phase*, feature vectors of known classification are applied as inputs to the network. The network then produces the output which it associates with the applied feature vector. This output, the neural network classification, is then compared with the true (known) classification. If the neural network classifier has a high percentage of correct classifications, then excellent *generalization* has been achieved. If there is a low percentage of correct classifications, then this is described as poor generalization. Poor generalization is usually caused by inadequate training examples. In addition, the testing phase can reveal other 'human qualities' (such as memorization and confusion) of neural network classifiers. Such qualities are demonstrated where appropriate in the next section.

Figure 4.2: A Training Example with Two Classes.

4.2 The Basics of Neural Networks

4.2.1 Processing Elements

Just as the biological neuron is the building block of the brain, the *processing element* (PE) is the building block of a neural network. PEs are also referred to as neurons, short-term memory, nodes, or threshold logic units. Figure 4.3 depicts a generic processing element. The input signals form an input vector $u = (u_1, u_2, \dots, u_n)$. Each component of the input vector is connected to the PE by an adjustable value called a *weight*. Weights are also referred to as connection strengths, interconnects, or long-term memory. Sometimes there is a threshold value t , associated with the j^{th} PE. The weights form a vector $w_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ which has the dot product $u \cdot w_j$ taken to determine the scalar input value to the PE. The resulting scalar is simply a weighted sum less the threshold value as shown in Figure 4.3. This value is then passed through a *transfer function* to produce the output V_j . The output V_j branches ('fans out') into as many connections as desired, each branch carrying

Figure 4.3: A Generic Processing Element.

the same processing element output signal.

Four possible transfer functions for the PE are shown in Figure 4.4. All of these functions perform a nonlinear mapping of the input to the output. These functions include the ramp, the step, the sigmoid, and the exponential. The most commonly used is the sigmoid transfer function. It has the desirable properties of being monotonic, non-decreasing, differentiable, and bounded for all real inputs. As well, its shape can be varied to closely approximate either a ramp or step function.

4.2.2 Network Topology

Neural networks are formed by organizing PEs into layers and joining them with weighted interconnections. Figure 4.5 shows a simple neural network with three layers: the *input layer* which receives input signals from the environment, the *output layer* which emits signals to the environment, and the *hidden layer* which has no direct contact with the environment.

The three primary interconnection schemes are *intra-layer*, *inter-layer*, and *re-*

Figure 4.4: Transfer Functions used in Processing Elements.

current. Intra-layer connections are between PEs in the same layer; inter-layer connections are between PEs in different layers; and recurrent connections loop back to connect to the same PE. Inter-layer connections allow information to propagate in one of two ways: *feedforward* or *feedback*. Feedforward topologies allow information to flow among PEs only in the forward direction. That is, the output from a PE in a given layer can only be connected to the input of PEs in succeeding layers. Feedback topologies allow information to flow among PEs in any direction. For example, the output of a PE in a given layer may be connected to the input of a PE in a preceeding layer. The neural network of Figure 4.5 has a three layer, fully connected feedforward topology.

4.2.3 Memory and Learning

As mentioned in Section 4.2.1, the long-term memory of a neural network is stored in the interconnect weights. A neural network acts as either *content addressable memory* or *associative memory*. Content addressable memory is provided when

Figure 4.5: A Simple Neural Network.

the network maps data to addresses by storing the data at stable states in its memory matrix $\mathbf{W}$. Associative memory is provided when the network maps data to data. There are two primary associative mapping mechanisms: *autoassociative* and *heteroassociative*. A network is autoassociative if its memory matrix $\mathbf{W}$ stores n vectors (patterns) $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$. A network is heteroassociative if $\mathbf{W}$ stores n pattern pairs as $(\mathbf{p}_1, \mathbf{q}_1), (\mathbf{p}_2, \mathbf{q}_2), \dots, (\mathbf{p}_n, \mathbf{q}_n)$.

In order to act as content addressable or associative memory, the network must first *learn* the desired mapping. Learning is defined mathematically as $d\mathbf{W}/dt \neq 0$. Learning methods are either *supervised* or *unsupervised*. Supervised learning depends on an external teacher to teach associations by example. The teacher must determine how long and how often to present each association for training, what performance error information is used, and when the training is complete. Unsupervised learning has no external teacher and relies on only internal control and local

Figure 4.6: The Basic Backpropagation Neural Network (BPNN).

information. This technique is also known as self-organization. Supervised learning is used in this study and an example is given in the next section.

4.2.4 The Backpropagation Neural Network

To understand how a neural network can 'learn', an example is needed. One of the most commonly used neural systems is the *backpropagation neural network* (BPNN), as popularized by Rumelhart et al [47]. Figure 4.6 shows the basic three layer, fully connected feedforward topology of a BPNN. When a BPNN is used as a classifier, it operates in the heteroassociative mode with supervised learning.

Figure 4.7 depicts the classification problem as introduced in Section 4.1. This BPNN has a 4-dimensional input vector x and a 2-dimensional output vector y . The components of both x and y are real numbers. The output vector y is to distinguish between the categories of **NORMAL** and **ABNORMAL** as shown. The number

of hidden layer PEs is fixed at l . As shown in Figure 4.7, there are a total of 16 training vectors available for the training phase.

The learning process begins with each of the interconnection weights initialized to a random value in the interval $[-1,1]$. Thus, initially the memory matrix $\mathbf{W}$ contains no knowledge. The network is now trained from one and only one training vector at a time. A training vector is selected at random from the entire training set (NORMAL and ABNORMAL). The vector components are presented as inputs to the network. Simultaneously, the network is shown the desired output (the class to which the training vector belongs). The network then computes its outputs¹ according to the knowledge (contained in its weights) that it thus far has accumulated. For each output layer PE an error is formed (error = desired - computed) and is used to adjust all of the incoming interconnection weights from the l hidden layer PEs. Each weight is adjusted² by a small amount so as to reduce the output error. Finally, the error for each output PE is backpropagated³ to each hidden layer PE. For each hidden layer PE the backpropagated error is then used to adjust all of the incoming interconnection weights from the four input layer PEs. This completes one cycle of learning and is the essence of the backpropagation learning algorithm. The next cycle begins with the random selection of another training vector. After thousands of cycles, the memory matrix $\mathbf{W}$ gains its knowledge from the training vectors. The network is then ready for the testing phase as described in Section 4.1.

The choice of the number, l , of hidden layer PEs is open to investigation. In general, choosing too many hidden layer PEs allows too many degrees of freedom for the network. This is manifest in the testing phase as poor classification of independent test vectors, but excellent classification of training vectors. Thus, with too many hidden layer PEs, the network may effectively memorize the training set and have poor generalization. However, choosing too few hidden layer PEs limits

¹This is accomplished by having each PE compute its output value as described in Section 4.2.1. The BPNN typically uses a sigmoid transfer function as in Figure 4.4c.

²Various adjustment schemes are available as described in [50].

³For example, the error at processing element $b1$ is $e_{b1} = w_{11}e_{c1} + w_{12}e_{c2}$ where e_p is the error at processing element p .

Figure 4.7: The BPNN as a Classifier.

the network capacity and can also yield poor classification results. The choice of the number of hidden layer processing elements in a BPNN must strike a balance between these positions.

Finally, deviations from the basic three layer, fully connected topology of the BPNN can be explored. For example, other BPNN topologies may add more hidden layers and/or prune some interconnections.

4.3 The Probabilistic Neural Network

The *Probabilistic Neural Network*, PNN, was originally developed by Specht [51] and is a neural network implementation of an established statistical method. The PNN is based on the *Bayes minimum error decision rule* which is defined as follows: suppose that a classification problem has k classes represented by $C_1, C_2, \dots, C_k$ and that the relative frequencies or *a priori probabilities* of each of these classes is represented by $P_1, P_2, \dots, P_k$. Furthermore, let the *probability density function*, PDF, for each class be represented by $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})$. Finally, if the observations on which the classification decisions are based is represented by an N -dimensional feature vector $\mathbf{x}$ where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix},$$

then the Bayes minimum error decision rule is to choose the class corresponding to the maximum of k values

$$P_1 f_1(\mathbf{x}), P_2 f_2(\mathbf{x}), \dots, P_k f_k(\mathbf{x}). \quad (4.1)$$

For two classes the Bayes minimum error decision rule can be expressed in the *likelihood ratio* form

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{P_2}{P_1} \Rightarrow \mathbf{x} \in \begin{cases} C_1 \\ C_2. \end{cases} \quad (4.2)$$

The symbol $\in$ means 'is an element of' and signifies that $\mathbf{x}$ belongs to class C_1 if the left hand ratio is greater than the right hand ratio, and that $\mathbf{x}$ belongs to C_2 if the left hand ratio is less than the right hand ratio. This rule assumes that the penalty due to misclassifying a class C_1 member as a class C_2 member is the same as vice-versa; however, this is not always the case. For example, consider a preliminary test where misclassifying a patient with a brain tumor as being healthy is much more serious than the other way around. This concept is formalized by using a *cost function* R_{ij} , which is the cost of misclassifying an object from class C_i as belonging to class C_j ⁴. Taking into account the overall cost or *risk*, it can be shown [19] that the *Bayes minimum risk decision rule* for two classes is

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \gtrless \frac{R_{21} P_2}{R_{12} P_1} \Rightarrow \mathbf{x} \in \begin{cases} C_1 \\ C_2. \end{cases} \quad (4.3)$$

Once again, with either Equation 4.2 or 4.3 the decision rule for the two class case is of the form

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \gtrless \text{constant} \Rightarrow \mathbf{x} \in \begin{cases} C_1 \\ C_2. \end{cases} \quad (4.4)$$

Thus, the classification problem is reduced to determining the conditional probability density functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$. The PDFs are unknown and must be estimated from samples of known classification. The PNN uses a *non-parametric* method of estimating the PDFs as developed by Parzen [41]. The Parzen estimator (of the PDF) is non-parametric in that it makes no simplifying assumptions about the nature of the PDF distribution. The Parzen estimator is a *kernel* based estimator in that it consists of a summation of a set of smaller parametric functions. For example, the Parzen estimator used in the PNN has the form

$$f_k(\mathbf{x}) = \frac{1.0}{2\sqrt{\pi}\sigma} \cdot \frac{1.0}{n_k} \sum_{j=1}^{n_k} e^{-(\mathbf{x} - \mathbf{x}_{kj})^T(\mathbf{x} - \mathbf{x}_{kj})/2\sigma^2} \quad (4.5)$$

where $\mathbf{x}_{kj}$ is the j^{th} training vector in class C_k , n_k is the total number of training vectors available for class C_k , and σ is a *spread* or *smoothing* parameter. Essentially,

⁴Note that by definition $R_{ii} = 0$ for all i . This is interpreted as no cost for a correct classification.

each training vector is used to position a small Gaussian curve and these curves are added together to estimate the PDF for each class. The Gaussian distribution is defined as

$$g(x) = \frac{e^{-(x-m)^2/2\sigma_v^2}}{\sqrt{2\pi}\sigma_v}$$

where m and σ_v^2 are the mean and variance of x , respectively.

The role of the smoothing parameter is illustrated in Figures 4.8, 4.9 and 4.10. In each figure the PDF estimate is for the 1-dimensional vector (a scalar) case of five training vectors (samples) taken from a zero mean Gaussian (normal) distribution (shown as a dashed line). Note that the y -axis scale for Figures 4.9 and 4.10 is 1/2 that of Figure 4.8.

For a given kernel shape, σ determines how much each training sample contributes to any point, x . If σ is too small, then the estimate is undersmoothed and is simply a collection of sharp peaks located at each sample point as seen in Figure 4.8. If σ is too large, then the estimate is oversmoothed and smeared into an almost uniform PDF as seen in Figure 4.10. For this example, a $\sigma = 0.5$ gives the closest estimate of the underlying distribution. The estimate consists of a summation of five Gaussian curves according to Equation 4.5. In this example, the underlying distribution happens to be of Gaussian form; however, the underlying distribution can be of any form.

In determining the smoothing parameter, it is also apparent that σ must depend on the sample size, n_k . With very few training vectors more smoothing (a larger σ) is needed to prevent a series of peaks located around the training vectors. Conversely, with a large number of training vectors less smoothing (a smaller σ) can be used. Thus, σ is required to be a function of n_k such that $\lim_{n_k \rightarrow \infty} \sigma(n_k) = 0$. Furthermore, as described in [41] for the estimate of $f_k(\mathbf{x})$ to be consistent, it is required that $\lim_{n_k \rightarrow \infty} n_k \sigma(n_k) = \infty$. One way to satisfy these two requirements is to define the smoothing parameter as follows:

$$\sigma(n_k) = \frac{a}{n_k^b} \quad \text{where: } 0 < b < 1. \quad (4.6)$$

The smoothing parameter as defined above allows for easy modification of the *decision boundaries*. A decision boundary lies between the regions Ω_i , $i = 1, 2, \dots, k$

Figure 4.8: A PDF Estimation Example, $\sigma = 0.1$.

Figure 4.9: A PDF Estimation Example, $\sigma = 0.5$.

Figure 4.10: A PDF Estimation Example, $\sigma = 3.0$.

of the k classes into which the decision rule classifies the N -dimensional feature vector, $\mathbf{x}$. As an example, consider the decision rule for two classes (Equation 4.4). The decision boundaries (Ω_1, Ω_2) are determined by the estimates of the PDFs ($f_1(\mathbf{x}), f_2(\mathbf{x})$). In turn, these PDF estimates depend on the choice of the smoothing parameter. Thus, the decision boundaries of the PNN are modifiable by changing the a and b parameters of the smoothing function.

In general, the PNN is a four layer, partially connected, feedforward network. The topology of a PNN is demonstrated in Figure 4.11. This particular topology supports the two category classification problem as introduced in Section 4.1. The PNN has a 4-16-2 topology in that it has a 4-dimensional input vector $\mathbf{x}$, 16 *pattern layer* PEs, and a 2-dimensional output vector $\mathbf{y}$. The output vector $\mathbf{y}$ is used to distinguish between the NORMAL and ABNORMAL categories as shown.

The purpose of the PNN topology is to implement Equation 4.5 for each class (NORMAL and ABNORMAL) and to make a decision based on Equation 4.4. This is accomplished as follows. The input layer PEs have ramp transfer functions (refer to

Figure 4.11: A Probabilistic Neural Network.

Figure 4.4a) and simply present the input vector $\mathbf{x}$ to each of the *pattern layer* PEs. There is a one-to-one mapping of pattern layer PEs to training vectors. Specifically, pattern layer PEs b_1 - b_{10} are mapped to training vectors $\mathbf{x}_{n1}$ - $\mathbf{x}_{n10}$ and pattern layer PEs b_{11} - b_{16} are mapped to training vectors $\mathbf{x}_{a1}$ - $\mathbf{x}_{a6}$ respectively. Each pattern layer PE stores (in its incoming interconnection weights) the components of its associated training vector $\mathbf{x}_t$ ($t = n1, n2, \dots, n10, a1, a2, \dots, a6$). When an input vector $\mathbf{x}$ (to be classified) is presented, each pattern layer PE computes the Parzen kernel $s = (\mathbf{x} - \mathbf{x}_t)^T(\mathbf{x} - \mathbf{x}_t)$ where $\mathbf{x}_t$ is the associated training vector. The pattern layer PE then uses the exponential transfer function $e^{-\alpha s}$ with $\alpha = 1/2\sigma^2$ where σ is the smoothing parameter.

The *summation layer* simply groups the pattern layer PEs into the two categories and provides the $1/n_k$ ($k = 1, 2$) scaling factor for the summation of Equation 4.5. Finally, the output layer provides the decision rule of Equation 4.4 and indicates the result via vector $\mathbf{y}$ as shown in Figure 4.11.

The PNN training is quite different to that of the BPNN classifier of Section 4.2.4. The PNN training phase simply involves assigning a pattern layer PE to each training vector. As well, the pattern layer PE must have its output connected to the appropriate summation layer PE. This completes the PNN training. Thus, the PNN trains very quickly relative to the BPNN.

Another significant advantage of the PNN over the BPNN is the control over the decision boundaries. As described previously, the PNN can have its decision boundaries modified by changing either the smoothing parameter or the decision threshold constant of Equation 4.4. Thus, the degree of nonlinearity of decisions is modifiable. Such control is not easily achieved with a BPNN which acts somewhat as a 'black box' classifier.

A disadvantage of the PNN is that it typically requires a large number of training vectors to achieve good generalization. The number of training vectors required to achieve successful generalization depends on the complexity of the PDFs for each class. This is the penalty paid for using a non-parametric method (the Parzen estimator of Equation 4.5) to estimate the PDFs. However, given sufficient data of known classification, this method can closely approximate the true underlying

PDFs (whatever their form may be).

In summary, the PNN provides a very general and powerful classification technique. The PNN is a neural network implementation of a Bayes classifier where the class dependent PDFs are approximated using a Parzen estimator. The Parzen estimator is a non-parametric method of estimating the PDFs in that it makes no simplifying assumptions about the nature of any PDF distribution. This is accomplished by using a summation of Gaussian curves to form a PDF estimate. Once the PNN is trained, the final classification decision can be optimized to minimize the expected risk by simply modifying the decision threshold constant of Equation 4.4.

4.4 Related Research

Recently, much work has concentrated on applying neural networks as pattern classifiers in medical applications. The field of clinical medicine has an ever growing demand for classification techniques. Neural networks are becoming a highly explored alternative in a diverse cross section of projects.

Feature detection in medical imaging is one such project applying neural network classifiers. Cios et al. [4] attempt to detect cardiac diseases from two dimensional echocardiographic images. It is shown that a BPNN classifier is capable of detecting subtle differences between normal and diseased cardiac tissues. In [14], Garg and Floyd attempt to detect pulmonary nodules (lung growths) from a chest radiograph (x-ray). This requires feature detection in a complicated anatomical background. After successful network training, an analogy of the network's processing to that of human visual processing is made.

Another significant area for the application of neural network classifiers is that of evoked potentials. In [32], Lipoth et al. apply a BPNN classifier to Electroretinographical (retinal responses evoked from flash stimuli) waveforms in order to classify eye diseases. Highly successful classification (an overall 95% classification accuracy) is achieved with a compact network topology. Holdaway et al. [24] classify head injuries using EEG recorded responses to a somatosensory stimuli (a nerve excita-

tion). The preliminary results are quite promising towards the development of a head injury management system.

Other research related to the neural network processing of EEGs is also under way. Sreenivas et al. [52] use a neural network to explore the possibility of separating the EEG signal from artifacts. The network is used as a filter to reduce the noise (artifact) in the EEG. In terms of signal enhancement, a 5 dB improvement in the signal to noise ratio is achieved. In [44], Ramadan et al. apply a BPNN classifier to distinguish between EEGs in normal and dyslexic readers. Although limited EEG data was available, the preliminary results are encouraging (an overall 87% classification accuracy).

This thesis focuses on the PNN classifier based detection of abnormalities in electrical brain activity. In Chapter 5, the PNN classifier is applied to the detection of abnormalities in the EEG. As described in Section 3.3, the classification of the EEG is into either a NORMAL or ABNORMAL category. Presently, this research represents part of a new frontier in the application of neural networks. It is comparable to the neurometric studies discussed in Section 3.1.3.

Chapter 5

The PNN Based Detection of EEG Abnormalities

In this chapter, the research methodology combines the techniques of quantitative EEG analysis (Chapter 3) and neural networks used as pattern classifiers (Chapter 4). This chapter summarizes the PNN classification of EEGs based on abnormalities detected from EEG features. The first section provides the background needed to understand and discuss the performance of the PNN classifier. The second and third sections present research results from various PNN topologies. The fourth section is an interpretation of these results. Finally, the last section discusses improvements to the research methodology with an emphasis on neural network applications.

5.1 The PNN Classifier

This section describes the construction of the training files used for the PNN topology used in the first set of simulations. The training files consist of NORMAL and ABNORMAL examples of the relative asymmetrical power feature vectors. Once the PNN is trained, its classification performance must be quantified. Section 5.1.2 describes the performance measures which are used to assess the PNN classifier.

5.1.1 Training Files and Feature Selection

The first PNN topology explored for the detection of EEG abnormalities is that of Figure 5.1. The input consists of the delta band 8-dimensional relative asymmetrical power feature vector $\mathbf{x}$ as shown in Figure 5.2. The relative asymmetrical power feature vector is useful for the detection of a wide variety of cerebral abnormalities from epileptic activity to structurally induced (via tumors, lesions, etc.) activity [9, 40, 37, 10, 11]. The components of the relative asymmetrical power feature vector are arranged from anterior to posterior electrode locations as shown in Figure 5.2. For each electrode pair (homologous locations), a right-sided asymmetry is indicated by a positive value and a left-sided asymmetry is indicated by a negative value. For this feature vector all components are bounded to the interval of $[-1,1]$. Note that the central three electrode locations (Fz, Cz, and Pz) are not represented in this feature vector.

Referring to Figure 5.1, the training file T_1 consists of 30 NORMAL training vectors and 15 ABNORMAL training vectors. Figure 5.3 shows the construction of 9 training files from a population of 60 NORMALs and 25 delta band ABNORMALs. Note that of the original 30 ABNORMALs from Section 3.3, only 25 showed delta band abnormalities. The remaining 5 showed no delta band abnormalities and are considered 'normal ABNORMALs' for this band. The overall EEG is categorized as ABNORMAL due to abnormalities detected in one or more of the other bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3).

5.1.2 Assessing the Classification Performance

The assessment of the PNN classification performance is best described by example. Figure 5.4 demonstrates the classification results for the PNN classifier of Figure 5.1 trained with training file T_1 of Figure 5.3. The G and F parameters define the smoothing parameter as follows

$$\sigma(n_k) = \frac{G/100}{n_k^{-(F/100)}} \quad \text{where: } 0 < F < 100. \quad (5.1)$$

Note that this corresponds to Equation 4.6 with $a = G/100$ and $b = F/100$. Training file T_1 consists of 30 NORMAL and 15 ABNORMAL training vectors. After training,

Figure 5.1: An 8-45-2 PNN Topology.

Figure 5.2: Delta Band Relative Asymmetrical Power Feature Vector.

the entire set (60 NORMALs, 25 ABNORMALs, and 5 'normal ABNORMALs') of available vectors is used to test the PNN classification performance. In Figure 5.4 the solid line represents the fraction of NORMALs correctly classified as 'NORMAL'; the dashed line represents the fraction of ABNORMALs correctly classified as 'ABNORMAL'; the dash-dot line represents the fraction of 'normal ABNORMALs' correctly classified as 'NORMAL'. The horizontal axis in Figure 5.4 shows how the choice of the decision threshold constant of Equation 4.4 affects the classification results. P_n and P_a are the relative frequencies (probabilities) with which the EEG belongs to the NORMAL and ABNORMAL classes, respectively.

The dashed and solid lines are of primary importance in assessing the PNN classifier. The dashed line represents a measure of the *apparent sensitivity* of the classifier, while the solid line represents the *apparent specificity* of the classifier. The qualifier 'apparent' indicates that these are optimistic estimates of the sensitivity

Figure 5.3: Construction of Delta Band Training Files.

Figure 5.4: Classification Results for an 8-45-2 PNN.

and specificity, respectively. They are optimistic because the test set of vectors (all available vectors) includes the original training vectors. For Figure 5.4 the training vectors make up 50% of the NORMAL and 60% of the ABNORMAL test vectors. Vectors not included in the training set are known as *independent* test vectors and can be used to estimate the *true* sensitivity and specificity of the PNN classifier.

For an accurate estimate of the true performance measures (sensitivity and specificity), a large set of independent test vectors is needed. However, a large number of independent test vectors reduces the number of vectors which can be used as training vectors. To both maximize the number of training vectors and achieve the most accurate estimate of the true performance measures, the *leave-one-out* method [19] can be used. This method is an iterative approach as follows. Suppose that there are n feature vectors available for the training and test sets. To start, the PNN is trained on all but the first feature vector (the 'one-left-out'). This singular test vector is then classified by the PNN and the result is tabulated. The process

continues with each feature vector being classified in turn by the PNN classifier trained on the other $n - 1$ feature vectors. This involves many computations as n distinct PNN classifiers must be trained and tested. However, the procedure yields an asymptotically (with increasing n) accurate estimate of the true sensitivity and specificity of the PNN classifier.

5.2 The PNN Apparent Performance Measures

This section explores the characteristics of the PNN by examining the apparent performance measures of sensitivity and specificity. The performance measures are computed for the PNN trained on the delta band relative asymmetrical power feature vectors. The characteristics explored include the effects of varying the smoothing parameter and of increasing the size of the training set.

5.2.1 Simulation Set 1

The purpose of the first set of simulations is to vary the smoothing parameter, σ , and determine the effects on the classification performance. The classification categories of NORMAL and ABNORMAL are as originally defined in Section 3.3. This first set of simulations uses the delta band relative asymmetrical power feature vector and the 9 training files as described in Section 5.1.1.

The PNN classification results are assessed on the basis of the apparent sensitivity and specificity as described in Section 5.1.2. For each training file, the G and F parameters of Equation 5.1 are varied over a wide range ($1 \leq G \leq 200$ and $5 \leq F \leq 95$) to establish the characteristics of the PNN classifier. The simulations are run using a custom *Turbo C* program ('pnn.c') as found in Appendix C. This program has been verified by using independent results from the commercially available *NeuralWorks Professional II* software package. It was necessary to develop the *Turbo C* program because the *NeuralWork* package requires time consuming human supervision for the presentation of different training files. In addition, the *Turbo C* program collects statistics on the networks' performance and concisely summarizes the results. By comparison, the *NeuralWorks* result files are large and awkward to

Figure 5.5: Apparent Performance Graph - PNN with $G = 10$, $F = 80$.

process.

Figures 5.5 and 5.6 illustrate how varying the F parameter of the smoothing function affects the classification performance measures. In Figure 5.6 increasing the F parameter causes a decrease in the smoothing parameter and shifts the intersection point of the apparent sensitivity (dashed line) and apparent specificity (solid line) to the right. A similar effect is seen in Figure 5.7 where decreasing the gain G parameter also causes a decrease in the smoothing function. Thus, any change in F or G which decreases the smoothing parameter results in a rightward shift of the intersection point of the dashed and solid lines. An increase in the smoothing parameter causes a leftward shift in the intersection point.

Figure 5.8 uses the PNN trained on training file T_1 to show how varying the smoothing parameter affects the level of the intersection point. The F parameter is fixed ($F = 70$) and the gain is increased. Increasing the smoothing parameter by increasing the gain causes a decrease in the level of the intersection point. Recall from Section 4.3 that an excessively wide (large G) smoothing parameter produces

Figure 5.6: Apparent Performance Graph - Effect of Increasing F .

Figure 5.7: Apparent Performance Graph - Effect of Decreasing G .

Figure 5.8: The Effect of Varying the Smoothing Parameter.

an almost uniform PDF estimate which, in turn, degrades the classification performance. This is seen as a reduction in the level of the intersection point. Also, recall that an excessively narrow (small G) smoothing parameter produces a series of peaks as a PDF estimate which, in turn, causes the PNN to effectively 'memorize' the training vectors.

The optimal smoothing parameter corresponds to the highest fraction of correct classifications possible. Referring to Figure 5.8, the smoothing parameter appears to be optimal for $G = 0$ ¹. This is the case because the apparent sensitivity/specificity intersections are used. The curve for low values of G is acutely affected by the memorization of training vectors. Because the training vectors are also used as test vectors, the apparent sensitivity and specificity are optimistically inflated. The closer that the gain G moves towards zero, the more pronounced is the memorization and inflation. Thus, the true optimal value of the smoothing parameter can not be found from the apparent sensitivity/specificity intersection points. However, a

¹For $G = 0$, this is commonly known as *Nearest Neighbour Pattern Matching*.

reasonable range with $F = 70$ is $10 \leq G \leq 15$ for the PNN trained on T_1 . The lower bound of $G = 10$ is chosen conservatively knowing that memorization of training vectors has an increasing effect for lower values of G . The optimal value of the smoothing parameter can be found from the true sensitivity/specificity intersections. This is the approach used in Section 5.3.

Thus far, only the results for the PNN classifier trained on training file T_1 have been presented. Similar PNN classifiers are constructed for training files $T_2 - T_9$. The apparent sensitivity and specificity for each of these networks can be found in Appendix D. The results given are comparable as the networks use the same smoothing parameter ($G = 10, F = 70$) in each case.

5.2.2 Simulation Sets 2 and 3

The purpose of both the second and third set of simulations is to determine the effect of increasing the number of training vectors in the training files. The second set of simulations uses the same delta band relative asymmetrical power feature vector and the 9 training files as described in Section 5.1.1. However, the number of NORMAL training vectors are increased from 30 (simulation set 1) to 40. As well, the ABNORMAL training vectors are increased from 15 to 20. The resulting 9 training files are named $T_{1B} - T_{9B}$. Again, the simulations include a wide range of smoothing parameters ($1 \leq G \leq 200$ and $5 \leq F \leq 95$) and the results are assessed on the basis of the apparent performance measures.

Figure 5.9 illustrates the effect of the increase in the number of training vectors. Training file T_{1B} has an additional 10 NORMAL and 5 ABNORMAL training vectors relative to training file T_1 . With training file T_{1B} the fraction of correct NORMAL classifications increases at a faster rate than with training file T_1 . Similarly, the fraction of correct ABNORMAL classifications decreases at a slower rate. Thus, Figure 5.9 shows how the apparent performance measures improve. The intersection of the sensitivity/specificity is 0.83 for training file T_{1B} compared to 0.78 for training file T_1 . Similar improvements occur with training files $T_{2B} - T_{9B}$. The apparent sensitivity and specificity for each of these networks can be found in Appendix E. The results are comparable in each case to the corresponding training

file in Appendix D.

The third set of simulations uses all of the available feature vectors for training. That is, a training file T_{1C} consisting of 60 NORMAL and 25 ABNORMAL training vectors is used to train the PNN. Once again the simulations are run over a wide range of smoothing parameters and the results are assessed on the basis of the apparent performance measures.

The results for training file T_{1C} with a smoothing parameter defined by $G = 10$ and $F = 70$ are shown in Figure 5.10. The intersection of the sensitivity/specificity occurs at 0.97. This represents near perfect classification; however, this is an extremely optimistic result as all of the test vectors are also used as training vectors. To get a more accurate estimate of the classification ability of the PNN, the 'leave-one-out method' defined in Section 5.1.2 must be used. This is the approach utilized in the next section.

5.3 The PNN True Performance Measures

The performance measures of apparent sensitivity and specificity in the previous section provide optimistic results. They are useful only to demonstrate the characteristics of the PNN. The true performance of the PNN classifier is best estimated by using the 'leave-one-out' method of Section 5.1.2. The purpose of the simulations in this section is to use the 'leave-one-out' method to estimate the true performance of the PNN classifier.

The first set of simulations using the 'leave-one-out' method is based on the delta band relative asymmetrical power feature vector. Figure 5.11 summarizes the estimates of the true sensitivity (dotted line) and true specificity (solid line) for a delta band (relative asymmetrical power feature vector) trained PNN. A smoothing parameter with $G = 16$, $F = 50$ is used. The COMPOSITE training file in Figure 5.11 uses a total of 85 training files as follows. There are 85 (60 NORMAL and 25 ABNORMAL) delta band relative asymmetrical power feature vectors available. The first 60 training files consist of all but one of the NORMAL feature vectors (a different one for each file). The remaining 25 training files consist of all but one of

Figure 5.9: The Effect of Increasing the Number of Training Vectors.

Figure 5.10: PNN Trained on All Available Feature Vectors.

the ABNORMAL feature vectors (a different one for each file). The same 8-84-2 PNN² is then trained on each of the 85 resulting training files. In each case, the singular vector (the one left out) not used in training is used as a test vector. The result in each case is tabulated to obtain estimates of the true sensitivity and specificity. This is the PNN classifier implementation of the 'leave-one-out' method.

In Figure 5.11 the sensitivity/specificity intersection occurs at 0.58 (fraction of correct classifications). This is a near optimal smoothing parameter as shown in Figure 5.12. The smoothing parameter is varied by fixing the F parameter ($F = 50$) and increasing the gain G . The intersection of the true sensitivity/specificity is optimal at approximately 0.58 (fraction of correct classifications). Note that the near optimum range of the smoothing parameter is broad. Thus, a range of values for the smoothing parameter are available for near optimum performance.

The results of Figure 5.11 emphasize the importance of verifying the PNN classi-

²Recall from Section 4.3 that an 8-84-2 means an 8-dimensional input, 84 pattern layer, and 2 output PNN classifier

Figure 5.11: The PNN Delta Band Classification.

Figure 5.12: The Optimal Delta Band Smoothing Parameter.

fier's performance against an independent test set of EEG feature vectors. Comparing the performance in Figure 5.10 to that of Figure 5.11 demonstrates the optimism of the apparent performance measures. The PNN classifier of Figure 5.10 claims a 97% sensitivity and a 97% specificity. However, based on the independent test vector results (true estimates) of Figure 5.11, a 97% sensitivity yields only a 10% specificity. In a similar fashion, some of the results of John et al. [27] and Dolisi [9] are questionable because they only use the apparent sensitivity and apparent specificity performance measures (refer to Section 3.1.3). Those neurometric studies which are not verified with an independent test set of subjects are probably highly optimistic.

Thus far, only PNN classification results for the delta band feature vectors have been presented. However, similar relative asymmetrical power feature vectors are constructed in the remaining energy bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3). For each of these bands, there are 60 NORMAL feature vectors and a band dependent number of ABNORMAL feature vectors available for training. Recall from Section 5.1.1 that of the 30 ABNORMAL feature vectors available for training, only 25 are considered to have delta band abnormalities. The remaining 5 are considered 'normal ABNORMALS' in the delta band. Similarly, the other five bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3) have band specific ABNORMAL feature vectors (25, 20, 25, 24, and 25, respectively) with the remainder designated as 'normal ABNORMALS' for that band.

A final set of simulations using the 'leave-one-out' method for each of the remaining five bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3) was run. For each band, a graph similar to Figure 5.12 is used to determine the optimal range for the smoothing parameter. The results shown in Figures 5.13 - 5.18 are with smoothing parameters chosen within the optimal range. The solid line represents an estimate of the true specificity. The dashed line represents an estimate of the true sensitivity. The dash-dot line shows the fraction of correct classifications for 'normal ABNORMALS' and is presented for completeness. Table 5.1 summarizes the classification results of Figures 5.13 - 5.18 inclusive. These results are discussed in detail in the next section.

Figure 5.13: The PNN Delta Band Classification Results.

Figure 5.14: The PNN Theta Band Classification Results.

Figure 5.15: The PNN Alpha Band Classification Results.

Figure 5.16: The PNN Beta1 Band Classification Results.

Figure 5.17: The PNN Beta2 Band Classification Results.

Figure 5.18: The PNN Beta3 Band Classification Results.

Specificity for Given Sensitivity					
Band	Sensitivity:				
	95%	90%	80%	70%	65%
Delta	12%	17%	27%	43%	48%
Theta	1%	3%	39%	56%	63%
Alpha	40%	42%	54%	63%	65%
Beta1	4%	7%	13%	30%	39%
Beta2	14%	18%	23%	47%	48%
Beta3	24%	40%	50%	55%	59%
Based on Relative Asymmetrical Power Feature Vector.					

Table 5.1: A Summary of PNN Classifier Results.

5.4 The PNN Classification Results

The results of Table 5.1 represent true estimates of the PNN classification performance. By far, the alpha band relative asymmetrical power feature vector achieves the best classification. In ascending order, the next best classification occurs in the beta3, theta, delta, beta2, and beta1 bands.

As demonstrated in Section 5.3 with the delta band true performance measures, the PNN classifier does not perform as well as the optimistic results of Section 5.2. A similar decrease in performance occurs in all of the other bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3). The following are factors which decrease the performance of the network:

1. The high interindividual variability of the NORMAL population.
2. Training on NORMALs which may actually be ABNORMAL .

3. Training on ABNORMALs which may actually be NORMAL .
4. An insufficient number of training vectors.
5. The presence of too much artifact.
6. The lack of familiarity with the Beta (1.2 and 3) brain maps.
7. The fundamental approach of averaging epochs.
8. The finite discriminatory ability of the relative asymmetrical power feature vector.

The first three factors relate directly to the exclusionary criteria used in the formation of the normative population. Recall from Section 3.3 that the EEGs are divided into the NORMAL or ABNORMAL category based on neurological disorders detected by an expert interpreting the entire routine EEG and the frequency domain topographical maps of the sampled EEG. An EEG is placed in the NORMAL if no abnormality is detected. This formation of a normative population has a 'loose' exclusionary criteria when compared to other studies. For example, in [10] Duffy argues in favor of a *benchmark* approach to the establishment of a normative population. With the benchmark approach, subjects are allowed into the normative population on the basis of overall health. Potential NORMAL subjects are thoroughly examined by a general physician, a psychologist, and a neurologist. Subjects are excluded from the normative population if their examinations show any evidence of disease or abnormality from a comprehensive list (cancer, depression, epilepsy, etc.). The goal is to choose subjects who are "as healthy as we ourselves would like to be"[10]. The result is much tighter exclusionary criteria than used in this thesis. This tighter exclusionary criteria reduces the normal interindividual variability and allows easier separation of the NORMAL and ABNORMAL EEGs.

Because the exclusionary criteria used in this thesis is based on visual analysis, incorrect original classifications (in the formation of the normative reference set) are possible. For example, there is a non-zero probability that an ABNORMAL may be incorrectly classified as being NORMAL. Again, the benchmark approach would serve to reduce this occurrence.

The fourth factor refers to the relatively low number of ABNORMAL training vectors available. Figures 5.13, 5.14, 5.15, 5.16, 5.17, and 5.18 have only 25, 25, 20, 25, 24, and 25 ABNORMAL training vectors respectively. Increasing the number of ABNORMAL training vectors results in a better approximation of the underlying PDF. In turn, this allows the PNN classifier to more accurately distinguish between the NORMAL and ABNORMAL EEGs.

The fifth factor relates to the possible allowance of too much artifact in the quantified EEG features (the spectral analysis). The delta band is most prone to eye artifact. The results of Figure 5.13 may be showing the effects of allowing too much eye artifact. This conclusion is supported by examining the theta and alpha band results of Figures 5.14 and 5.15. Both bands show significantly better classification performance than for the delta band. To reduce the amount of eye artifact contaminating the spectral features, the eye artifact threshold constant of Section 3.3 can be lowered. This, however, reduces the number of ABNORMAL EEGs available (from the present pool of 30 subjects) for training the PNN.

The sixth factor refers to the relative lack of familiarity with the meaning of the beta (β_1, β_2 , and β_3) band maps as compared to the delta, theta, and alpha maps. That is, the neurologist who originally classified the EEGs has more experience with the lower frequency band maps (δ, θ , and α) than with the higher frequency beta band maps. This is reflected in Figures 5.16 and 5.17 where the sensitivity/specificity intersections are 0.52 and 0.56 respectively. In the β_1 band, the PNN classifications are not significantly better than guessing.

The seventh factor questions the estimation of background brain activity by averaging epochs. This technique may not provide features powerful enough to discriminate all types of cerebral abnormalities from a normative reference. By exhaustively exploring features based on the estimation of background brain activity, the limitations of this approach will eventually be revealed. Important questions such as the length and number of epochs (for averaging) still remain.

Finally, the eighth factor considers the limited discriminatory ability of the relative asymmetrical feature vector. It is not possible for the relative asymmetrical power to detect all abnormalities. For example, a focalized abnormality (such as a

tumor) located directly below the central electrode (Cz) may not be detectable as it may not produce an asymmetry. More generally, this eighth factor also recognizes the lack of exploration of other feature vectors in this thesis. The exploration of other features is an important issue and must be addressed in future work. The PNN classifier provides a powerful and general tool for such work. The next section discusses this issue along with other possible improvements to the EEG data processing.

5.5 Improvements to the Research Methodology

Three areas of possible improvement are identified: EEG data processing, EEG feature selection, and improving the PNN classifier. Work relating to the use of neural networks is emphasized.

5.5.1 EEG Data Processing

One of the most important areas open for improvement in the EEG data processing is that of artifact removal. As described in Section 3.3, the present artifact removal consists of the visual examination of the sampled time domain epochs (20 epochs per subject). An epoch with any artifact, except for eye movement artifact, is discarded. If an epoch has eye movement artifact, the threshold function of Section 3.3 is applied. Only epochs where the eye movement exceeds the threshold constant are discarded. The purpose of this function is to reduce the number of epochs lost due to eye movement.

In most cases, the procedure is necessary only because of the limited number (20) of epochs collected per subject. An available solution is to use the artifact rejection capabilities of BrainLab (the data acquisition software of Section 3.2). BrainLab offers a simple threshold rejection option whereby the entire epoch is rejected if any EEG channel exceeds an absolute voltage level. The first twenty non-contiguous, 4 second epochs which meet this criteria are recorded to disk. Such an approach has several problems. First, it works well only for relatively large artifact. The threshold constant can not be too low as valid brain activity may be rejected. Second, the

epochs are non-contiguous. This limits the options for analyzing the EEG on a time progressive basis. That is, the gaps between the epochs are not stored and can not be retrieved.

A better solution is to sample and store the continuous EEG data on a real time basis. Ideally, the entire routine EEG should be stored for offline analysis. For a 20 minute EEG with 20 channels and 8-bit samples with a sample rate of 64 samples/second, this represents approximately 1.5 Mbytes of storage per EEG. A system capable of sampling and storing the entire routine EEG is both desirable and feasible.

Even with the entire routine EEG available, the removal of artifacts presently relies on visual inspection. The subjective nature of visual inspection motivates the need to develop an automated artifact removal system. One possibility is a neural network, perhaps a BPNN or a PNN, based artifact recognition system. A properly trained neural network offers a consistent, standardized approach to the removal of artifact. This is an area in need of research.

Finally, another advantage of the digital storage of the entire routine EEG is that it allows for the definition of more features. For example, features (such as alpha band reactivity³) can be made which combine eyes closed and eyes open parameters. As well, features from activation procedures such as hyperventilation and photic stimulation can be extracted. Such features are useful in the detection of abnormalities.

5.5.2 EEG Feature Selection

In Section 3.4 efforts are made to improve the PSD estimates upon which all of the EEG frequency domain features are based. However, this thesis did not emphasize feature selection. Instead it develops the PNN classifier based on the relative asymmetrical power feature vector and carries this example to completion for each of the six standard frequency bands. There is a need for feature exploration and the PNN classifier is a valuable tool for the exploration. With sufficient feature vector

³The alpha band reactivity is $r_\alpha = \alpha_{EO}/\alpha_{EC}$ where α_{EO} and α_{EC} are the alpha band powers with eyes open and eyes closed, respectively.

examples, the PNN can provide estimates of the true sensitivity and specificity by using the 'leave-one-out' method.

New features need not be restricted by the standard six frequency bands as defined in Section 3.1.1. For example, new feature bands may be defined by searching the PSD for spectral peaks and centering the bands around these peaks. Thus, EEG dependent spectral bands could be defined and compared. As well, new features may also be defined by applying a neural network. For example, a BPNN could be trained to produce an *index of focality* feature. This index of focality could rate on a scale of 1-10 the overall severity of any focalities (a depression or a peak) found in a brain map. The index of focality could then become part of a feature vector used as input to a PNN classifier.

A slightly different application of the PNN classifier is in conjunction with SPM. Recall from Figure 3.3 that SPM replaces, on an electrode by electrode basis, the original feature with the number of standard deviations, z , by which the observation differs from the normative reference set. This technique assumes a Gaussian distribution of the feature at each electrode. To achieve this assumption, it is often necessary to apply suitable mathematical transformations [15]. Instead of using the z -statistic with these transformations, consider the use of the PNN classifier for the feature at each electrode as shown in Figure 5.19.

The PNN classifier consists of a single input, the feature at one of the electrodes. The PNN classifier has two possible categories (NORMAL and ABNORMAL) into which the input must be classified. The network is trained from both a normative reference group (the NORMALs) and an abnormal reference group (the ABNORMALs). The classification of the electrode feature x depends on the decision ratio P_n/P_a as given by Equation 4.2 in Section 4.3. The classification of the electrode feature x always begins as ABNORMAL for $P_n/P_a = 0$ as shown in Figure 5.19. As the ratio of P_n/P_a increases, the network becomes more biased towards the NORMAL category. Eventually, at $r = P_n/P_a$, the classification of the electrode feature x changes to the NORMAL category. The ratio r at which this change occurs is a direct measure of the degree of abnormality of the feature x . Thus, the ratio r could replace the z statistic. That is, the ratio r can be calculated for each electrode based on the

Figure 5.19: PNN Classifier for a Single Electrode.

electrode specific PNN classifier. The r ratios from each electrode can then be used to generate a topographical brain map.

The advantage of the r ratio SPM over the z statistic SPM is that a Gaussian distribution of the electrode feature is not assumed. The PNN eliminates the need for mathematical transformations to approximate a Gaussian PDF. However, the PNN requires both a normative and an abnormal reference set for training.

5.5.3 Improving the PNN Classifier

The PNN Classifier is based on estimating the PDF of each class according to Equation 4.5 in Section 4.3. As such, any improvement to this estimation should make a better classifier.

One possibility is to experiment with different kernels. All PNNs in this thesis use Gaussian kernels with a variable smoothing parameter. As long as the kernel function $k(x)$ is nonnegative and has $\int_{-\infty}^{+\infty} k(x) dx = 1$, then it can be used as a kernel.

A second possibility is to modify the summation of the kernels in the PDF estimate. Equation 4.5 implies a simple summation of Gaussian kernels. Each kernel is centered at a training vector. If the training vectors are assumed to have a nonzero noise component, then the centering of the kernels is inaccurate. Perhaps the training vectors could first be placed into groups which are 'close' to each other. The degree of closeness could be measured by Euclidean distance or the Mahalanobis distance [1, 53]. The center of each of these groups could then be used to place the kernel. Such an approach may improve the PDF estimates and the resulting classification performance of the PNN.

Figure D.3: Apparent Performance Graph - Training File T3.

Figure D.4: Apparent Performance Graph - Training File T4.

Chapter 6

Conclusions

6.1 Research Summary

In this thesis, the first significant effort is in the investigation of the techniques of qEEG and brain mapping. In Section 3.1 a review of these techniques demonstrated the application of the Fast Fourier Transform (FFT) to estimate the power spectral density (PSD) of the EEG. The PSD computed from FFT spectral analysis is the basis for all frequency domain features of the EEG. These frequency domain features can be used to generate topographic brain activity maps. Such maps are clinically useful for the detection and localization of abnormalities found in the EEG. Three main approaches to utilizing these maps are identified:

1. Unaided Visual Inspection.
2. Significance Probability Mapping.
3. Neurometrics.

Unaided visual inspection of topographical brain maps uses the analytic power of the human visual system. It is the logical extension of the present visual based clinical interpretation of the raw EEG. Visual analysis of brain maps allows the clinician to learn how time domain activity maps to the frequency domain.

The main problem with unaided visual analysis is in discriminating between normal and abnormal variations. To assist in this decision, significance probability

mapping (SPM) is used. SPM transforms a topographical feature map to a z -statistic map. This transformation replaces, on an electrode by electrode basis, the original feature with the number of standard deviations, z , by which the observation differs from a normative reference group. The clinician can then search the map for significant deviations from the normative reference group.

The success of the process depends on the formation of a normative reference group. The formation of the normative reference group depends on age and exclusionary criteria. Exclusionary criteria is the level of control for admission of a subject into the normative reference group. If there is strict control for admission, then the exclusionary criteria is 'tight'; if there is easy admission, then the exclusionary criteria is 'loose'.

The choice of exclusionary criteria affects the performance measures of sensitivity and specificity. These measures are used to evaluate the performance of a classifier. Sensitivity is the ratio of individuals correctly classified as abnormal to the true number of abnormals. Specificity is the ratio of individuals correctly classified as normal to the true number of normals. If exclusionary criteria are too loose, then interindividual variation of the normative reference group is high and sensitivity is lost. In contrast, if exclusionary criteria are too tight, then a 'supernormal' reference group is formed and specificity is lost.

The classification performance measures of sensitivity and specificity are used to evaluate neurometrics. Neurometrics attempts to automate the clinical interpretation of EEGs. Neurometric studies make extensive use of SPM topographic brain maps to explore for EEG features useful in discriminating between pathological conditions and a normative reference set.

The research methodology in this thesis is similar to the methodology of neurometrics with two important differences. Firstly, this thesis has a more global view on EEG abnormalities. That is, neurometric studies focus on EEG abnormalities caused by specific pathological conditions. Neurometric studies then proceed to discriminate between specific pathological conditions and a normative reference group. Thus, neurometrics attempts automated diagnosis of pathologies based on abnormalities found in the EEG. In contrast, this thesis is simply concerned with

detecting abnormalities in the EEG. No attempt is made to identify the pathological condition causing the abnormality. Secondly, this thesis introduces the use of neural networks to the detection of EEG abnormalities. This is a novel approach.

This study uses a normative reference group from which the detection of EEG abnormalities are made. Section 3.3 describes the formation of the normative reference group. The EEGs are divided into two categories: NORMAL and ABNORMAL. These categories are formed based on the detection of abnormalities in the EEG. The abnormalities are detected by a neurologist interpreting the entire routine EEG and the frequency domain topographical maps of the digitally sampled EEG. Thus, this study has relatively loose exclusionary criteria when compared to other studies. The ramifications of this are important and are discussed in Section 5.4 and later in this summary.

This study utilizes frequency domain features of the EEG to detect abnormal cerebral activity. The features are based on the fundamental technique of estimating background brain activity by averaging PSDs from artifact-free epochs. A contribution of this thesis is the improvement of the frequency domain estimation of the background brain activity. Section 3.4.2 summarizes the introduction of Welch's modified periodogram [55] to the FFT based computation of the PSD. As the PSD is the basis for all EEG frequency domain features, this improvement is important. It allows for a tradeoff between frequency resolution and spectral variance.

In an effort to detect abnormal cerebral activity from the EEG frequency domain features, a Probabilistic Neural Network (PNN) classifier is used. The PNN is a neural network implementation of a Bayes classifier where the class dependent Probability Density Functions (PDFs) are approximated using a Parzen estimator. The Parzen estimator is a non-parametric method of estimating the PDFs in that it makes no simplifying assumptions about the nature of any distribution. The advantages of the PNN include its quick training and the easy modification of decision boundaries. The nonlinearity of the classification decisions is easily controlled by changing the smoothing parameter. A disadvantage of the PNN is that it typically requires a large number of training vectors to achieve good generalization. A lack of sufficient training vectors for detecting abnormalities in the EEG is evidenced in

the PNN classification results of Section 5.4.

Using the delta band relative asymmetrical power feature vector, the classification results of the PNN classifier are thoroughly explored. The PNN classifier yields an apparent sensitivity of 97% and an apparent specificity of 97%. These results, however, are optimistic because a set of independent test vectors is not used. To get an accurate measurement of the true sensitivity and specificity, the 'leave-one-out' method [19] is used. The 'leave-one-out' method both maximizes the number of training vectors used for the PNN classifier and achieves the most accurate estimate of the true performance measures. In this case, a 97% sensitivity yields only a 10% specificity.

These results emphasize the importance of verifying the PNN classifier's performance against an independent test set of EEG feature vectors. In a similar fashion, some of the results of John et al. [27] and Dolisi [9] are questionable because they only use the apparent sensitivity and apparent specificity performance measures (refer to Section 3.1.3). Those neurometric studies which are not verified with an independent test set of subjects are probably highly optimistic.

The 'leave-one-out' method is used for the relative asymmetrical power feature vectors in the remaining five frequency bands ($\theta, \alpha, \beta_1, \beta_2$, and β_3). The most successful classifier is with the alpha band where a 95% sensitivity yields a 40% specificity. This is of limited clinical utility.

All of the feature vectors for the PNN classifiers are based on the fundamental technique of estimating the background brain activity by averaging the PSDs from artifact-free EEG epochs. Five important issues remain in this approach:

1. The exclusionary criteria used in the formation of the normative reference set.
2. The number of subjects available in the data base.
3. The further reduction of artifact.
4. The exploration of additional features.
5. The duration and number of epochs used for spectral averaging.

The exclusionary criteria used in this thesis is relatively loose when compared to other studies. The main effect of loose exclusionary criteria is to allow a high degree of interindividual variation in the normative reference set. Because of the loose exclusionary criteria used in this thesis, all of the classification results should be interpreted as a measure of agreement with one expert's opinion of EEG 'normalcy'. There are two weaknesses in this approach: firstly, opinions of EEG 'normalcy' vary from one expert to another and, secondly, the analysis is highly dependent on visual analysis of the quantified EEG. To reduce the effect on classification performance of both of these weaknesses, the exclusionary criteria can be tightened.

In Duffy's benchmark approach [10] an individual must pass a rigorous set of examinations to be allowed into the normative reference group. Tightening the exclusionary criteria by using a benchmark approach results in a lower interindividual variation in the normative reference set. This increases the separability of abnormalities from the normative reference set. The result is improved classification performance by the PNN classifier.

The second issue is the limited number (60 NORMAL and 20-25 ABNORMAL) of training vectors available in this study. The results of Section 5.4 exhibited, in particular, a lack of training vectors from the abnormal EEGs. If the number of training vectors is increased, the PNN classifier will obtain a better estimate of the PDF. This should make a better classifier. The number of training vectors can be increased by expanding the EEG data base.

The third issue is the amount of artifact allowed in this study. The delta band PNN classifier exhibited significantly reduced performance relative to the theta and alpha band PNN classifiers. This may be due to the presence of too much eye artifact which most severely affects the delta band. The eye artifact in each epoch can be reduced by decreasing the threshold constant of Section 3.3.

The fourth issue is the exploration of alternative features. This study did not emphasize feature selection. Instead it develops the PNN classifier based on the relative asymmetrical power feature vector and carries this example to completion for each of the six standard frequency bands. There is a need for additional feature exploration. The PNN classifier is a valuable tool for this exploration. With suffi-

cient feature vector examples, the PNN can provide accurate estimates of the true sensitivity and specificity.

The fifth issue is the duration and number of epochs to use in the spectral averaging of brain activity estimates. At present, an open question is the total time segment of data needed to be averaged in order to obtain a representative sample of brain activity.

6.2 Further Research

An important issue in the establishment of the normative reference set is the exclusionary criteria. Future work should include tightening the exclusionary criteria by utilizing the benchmark approach to reduce the interindividual variability of the normative reference set. Also, the number of training set vectors should be increased by expanding the EEG data base, and the amount of artifact allowed in epochs (used for spectral estimates in the PSD) should be reduced by lowering the eye artifact threshold constant in Section 3.3.

Other future work should include improvements to the EEG data processing, to the EEG feature selection, and to the PNN classifier. Recommended EEG data processing improvements are twofold. First, develop and use a system which digitally captures the entire routine EEG. This allows for feature extraction from the entire EEG record. Second, develop an automated system for the recognition and removal of artifact. This is an application ideally suited to a neural network, perhaps a BPNN topology. Such a system would provide a consistent and standardized removal of EEG artifact.

Further research to determine the most representative spectral estimate of the background brain activity is important because the PSD is the foundation of all frequency domain features. The number and duration of epochs used in the spectral averaging of the PSD should be explored.

The exploration of more EEG features (other than the relative asymmetrical power) is left open for research. New features need not be restricted by the standard six frequency bands as defined in Section 3.1.1. Use of the z -statistic SPM and the

PNN classifier can be made. In addition, a PNN based r ratio SPM methodology is defined in Section 5.5.2. Such a methodology should be developed and evaluated for its usefulness in the exploration of EEG features. It may also be possible to use neural networks to define new EEG features. For example, a BPNN based 'index of focality' should be implemented and explored for its clinical utility. The 'index of focality' is defined in Section 5.5.2.

Experimentation with possible improvements to the PNN classifier is also recommended. The PDF estimates should be the focus of improvement as they are the backbone of the PNN. One possibility is to vary the kernel functions used in the PDF estimation.

A final recommendation is that the PNN classification performance should be compared to other neural network and more traditional classification techniques. For example, a comparison with various topologies of a BPNN classifier would be useful.

References

- [1] H. Ashida, J. Tatsuno, J. Okamoto, and E. Maru, "Field Mapping of EEG by Unbiased Polynomial Interpolation," *Computers and Biomedical Research*, vol. 17, pp. 267-276, 1984.
- [2] P. H. Bartels and H. G. Bartels, "Classification Strategies for Topographic Mapping Data," *Topographic Mapping of Brain Electrical Activity*, pp. 225-253, 1986.
- [3] M. S. Buchbaum, W. B. Mendelson, and W. Duncan, "Topographic Cortical Mapping of EEG Sleep Stages During Daytime Naps in Normal Subjects," *Sleep*, vol. 5, no. 3, pp. 248-255, 1982.
- [4] K. J. Cios, K. Chen, and R. A. Langenderfer, "Use of Neural Networks in Detecting Cardiac Diseases from Echocardiographic Images," *IEEE Engineering in Medicine and Biology*, vol. 9, no. 3, pp. 58-60, 1990.
- [5] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Computation of Complex Fourier Series," *Mathematics of Computations*, vol. 19, pp. 297-301, 1965.
- [6] R. Coppola, M. Buchsbaum, and F. Rigal, "Computer Generation of Surface Distribution Maps of Measures of Brain Activity," *Computers in Biology and Medicine*, vol. 12, no. 2, pp. 191-199, 1982.
- [7] M. R. Delucchi, "Techniques and Methods of Data Acquisition of EEG and EMG," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 3, no. C, 1974.
- [8] C. Deng and S. Haykin, "A Multilayer Neural Network Classifier for Radar Clutter," *International Joint Conference on Neural Networks*, vol. 1, pp. 1241-1246, June, 1990.
- [9] C. Dolisi, G. Suisse, and E. Delpont, "Quantitative EEG Abnormalities and Asymmetries in Patients with Intracranial Tumors," *Electroencephalography and Clinical Neurophysiology*, vol. 76, pp. 13-18, 1990.

- [10] F. H. Duffy, "Topographic Mapping of Brain Electrical Activity: Clinical Applications and Issues," *Topographic Brain Mapping of EEG and Evoked Potentials*, pp. 19-52, 1989.
- [11] F. H. Duffy, P. H. Bartels, and J. L. Burchfiel, "Significance Probability Mapping: An Aid in the Topographic Analysis of Brain Electrical Activity," *Electroencephalography and Clinical Neurophysiology*, vol. 51, pp. 455-462, 1981.
- [12] F. H. Duffy, J. L. Burchfiel, and C. T. Lombroso, "Brain Electrical Activity Mapping," *Annals of Neurology*, vol. 5, pp. 309-321, 1979.
- [13] F. Findji, P. Catani, and C. Liard, "Topographical Distribution of Delta Rhythms During Sleep: Evolution with Age," *Electroencephalography and Clinical Neurophysiology*, vol. 51, pp. 659-665, 1981.
- [14] S. Garg and C. E. Floyd, "The Training of an Artificial Neural Network for a Detection Task in Medical Images," *IEEE Engineering in Medicine and Biology Conference Proceedings*, vol. 13, no. 3, pp. 1407-1408, 1991.
- [15] T. Gasser, P. Bacher, and J. Mocks, "Transformations Towards the Normal Distribution of Broad Band Spectral Parameters of the EEG," *Electroencephalography and Clinical Neurophysiology*, vol. 53, pp. 119-124, 1982.
- [16] P. Gloor, "Generalized and Widespread Bilateral Paroxysmal Activities," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 11, no. B, 1976.
- [17] R. P. Gorman and T. J. Sejnowski, "Learned Classification of Sonar Targets Using a Massively Parallel Network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1136-1140, 1988.
- [18] H. L. Hamburger, "A Battery Approach to Clinical Utilization of Topographic Brain Mapping," *Topographic Brain Mapping of EEG and Evoked Potentials*, pp. 167-184, 1989.
- [19] D. J. Hand, *Discrimination and Classification*. New York, New York: John Wiley and Sons, 1989.
- [20] R. N. Harner, "Clinical Application of Computed EEG Topography," *Topographic Mapping of Brain Electrical Activity*, pp. 347-356, 1986.
- [21] R. Hess, "General Factors Influencing the EEG," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 11, no. B, 1976.

- [22] R. Hess, "Localized Abnormalities," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 11, no. B, 1976.
- [23] A. Hiraiwa, K. Shimohara, and Y. Tokunaga, "EEG Topography Recognition by Neural Networks," *IEEE Engineering in Medicine and Biology*, vol. 9, no. 3, pp. 39-42, 1990.
- [24] R. M. Holdaway, M. W. White, and A. Marmarou, "Classification of Somatosensory Evoked Potentials Recorded from Patients with Severe Head Injuries," *IEEE Engineering in Medicine and Biology*, vol. 9, no. 3, pp. 43-49, 1990.
- [25] H. H. Jasper, "The Ten-twenty Electrode System of the International Federation," *Electroencephalography and Clinical Neurophysiology*, vol. 10, pp. 371-375, 1958.
- [26] S. Jerret and J. Corsak, "Clinical Utility of Topographic EEG Brain Mapping," *Clinical Electroencephalography*, vol. 19, no. 3, pp. 134-143, 1988.
- [27] E. R. John, L. S. Pritchep, J. Friedman, and P. Easton, "Neurometric Topographic Mapping of EEG and Evoked Potential Features: Application to Clinical Diagnosis and Cognitive Evaluation," *Topographic Brain Mapping of EEG and Evoked Potentials*, pp. 90-117, 1989.
- [28] E. Kahn, R. Weiner, R. Brenner, and R. Coppola, "Topographic Maps of Brain Electrical Activity - Pitfalls and Precautions," *Biological Psychiatry*, vol. 23, pp. 628-636, 1988.
- [29] J. Kugler, "Generalized Non-Paroxysmal Abnormalities," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 11, no. B, 1976.
- [30] S. Lee and M. Buchsbaum, "Topographic Mapping of EEG Artifacts," *Clinical Electroencephalography*, vol. 18, no. 2, pp. 61-67, 1987.
- [31] D. Lehmann, "From Mapping to the Analysis and Interpretation of EEG and Evoked Potentials," *Topographic Brain Mapping of EEG and Evoked Potentials*, pp. 53-75, 1989.
- [32] L. L. Lipoth, H. M. Hafez, and R. A. Goubran, "Electroretinographical (ERG) Based Classification of Eye Disease," *IEEE Engineering in Medicine and Biology Conference Proceedings*, vol. 13, no. 3, pp. 1417-1418, 1991.
- [33] F. H. LopesdaSilva, "A Critical Review of Clinical Applications of Topographic Mapping of Brain Potentials," *Journal of Clinical Neurophysiology*, vol. 7, no. 4, pp. 535-551, 1990.

- [34] J. Morihisa, F. Duffy, W. Mendelson, and R. Wyatt, "The Use of Brain Electrical Activity Mapping (BEAM) as an Exploratory Technique to Delineate Regional Differences Between Schizophrenic Patients and Control Subjects," *Laterality and Psychopathology*, pp. 339-347, 1983.
- [35] R. Morstyn, F. Duffy, and R. McCarley, "Altered Topography of EEG Spectral Content in Schizophrenia," *Electroencephalography and Clinical Neurophysiology*, vol. 56, pp. 263-271, 1983.
- [36] Nicolet, "Nicolet BrainLab - Insights," technical manual, Nicolet Instruments Corporation, Madison, Wisconsin, 1989.
- [37] M. R. Nuwer, "Frequency Analysis and Topographic Mapping of EEG and Evoked Potentials in Epilepsy," *Electroencephalography and Clinical Neurophysiology*, vol. 69, pp. 118-126, 1988.
- [38] M. R. Nuwer, "The Development of EEG Brain Mapping," *Journal of Clinical Neurophysiology*, vol. 7, no. 4, pp. 459-471, 1990.
- [39] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, New Jersey 07632: Prentice-Hall, 1989.
- [40] D. Panet-Raymond and J. Gotman, "Asymmetry in Delta Activity in Patients with Focal Epilepsy," *Electroencephalography and Clinical Neurophysiology*, vol. 75, pp. 474-481, 1990.
- [41] E. Parzen, "On Estimation of a Probability Density Function and Mode," *Annals of Mathematics and Statistics*, vol. 33, pp. 1065-1076, 1962.
- [42] P. Passouant, "EEG and Sleep," *Handbook of Electroencephalography and Clinical Neurophysiology*, vol. 7, no. A, 1975.
- [43] A. Persson and B. Hjorth, "EEG Topogram - An Aid in Describing EEG to the Clinician," *Electroencephalography and Clinical Neurophysiology*, vol. 56, pp. 399-405, 1983.
- [44] Z. Ramadan, K. Ropella, J. Myklebust, M. Goldstein, X. Feng, and J. Flynn, "A Neural Network to Discriminate between Dyslexic Subtypes," *IEEE Engineering in Medicine and Biology Conference Proceedings*, vol. 13, no. 3, pp. 1405-1406, 1991.
- [45] D. M. Regan, *Human Brain Electrophysiology; Evoked Potentials and Evoked Magnetic Fields in Science and Medicine*. Box 882, Madison Sq. St., New York 10159: Elsevier Science Publishing Co., Inc, 1989.

- [46] E. A. Rodin, "Some Problems in the Clinical Use of Topographic EEG Analysis," *Clinical Electroencephalography*, vol. 22, no. 1, pp. 23-29, 1991.
- [47] D. E. Rumelhart, G. Hinton, and R. Williams, "Learning Representations by Backpropagating Errors," *Nature*, no. 323, pp. 533-536, 1986.
- [48] C. N. Schizas, C. S. Pattichis, I. S. Schofield, P. Fawcett, and L. Middleton, "Artificial Neural Nets in Computer-Aided Macro Motor Unit Potential Classification," *IEEE Engineering in Medicine and Biology*, vol. 9, no. 3, pp. 31-38, 1990.
- [49] H. Shyu, J. Libert, and S. D. Mann, "Classifying Seismic Signals Via RCE Neural Network," *International Joint Conference on Neural Networks*, vol. 1, pp. 1101-1105, June, 1990.
- [50] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*. Maxwell House, Fairview Park, Elmsford, New York 10523: Pergamon Press, Inc, 1990.
- [51] D. Specht, "Probabilistic Neural Networks for Classification, Mapping or Associative Memory," *International Conference on Neural Networks-88 Proceedings*, vol. 1, pp. 525-532, 1988.
- [52] T. Sreenivas, V. Unnikrishnan, and D. Narayana-Dutt, "Pruned Neural Network for Artifact Reduction in EEG Signal," *IEEE Engineering in Medicine and Biology Conference Proceedings*, vol. 13, no. 3, pp. 1411-1412, 1991.
- [53] J. Tatsuno, H. Ashida, and A. Takao, "Objective Evaluation of Differences in Patterns of EEG Topographical Maps by Mahalanobis Distance," *Electroencephalography and Clinical Neurophysiology*, vol. 69, pp. 287-290, 1988.
- [54] D. Walter, P. Etevenon, B. Pidoux, D. Tortrat, and S. Guillou, "Computerized Topo-EEG Spectral Maps: Difficulties and Perspectives," *Neuropsychobiology*, vol. 11, pp. 264-272, 1984.
- [55] P. 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*, vol. AU-15, no. 2, pp. 70-73, 1967.

Appendix A

EEG Artifact Identification.

The following computer listing provides a record of the artifact identified in the epochs for each subject in the data base used for this thesis. Where the EEG was **not** recorded with a common reference montage, the record is preceded by an 'X'. These records are not used in this thesis. The legend for the artifact types is as follows:

- B** Bad electrode connection.
- E** Eye movement artifact.
- J** Jaw movement artifact.
- K** Electrocardiographic artifact.
- M** Muscle artifact.
- P** Electrode pop (disconnected).
- S** Swallowing artifact.
- X** Epoch unavailable (not recorded).
- ?** Other artifact.

'NORMALS' :

EEG I.D.	Epoch 1-10	Epoch 11-20
	1234567890	1234567890
eeg1	X	
eeg2	X	
eeg3	X	
eeg4	X	
eeg5	X	
eeg6		
eeg7	EEEEEEEEEE	EEEEEEEEEE
eeg8		
eeg9		
eeg10		

EEG I.D. Epoch 1-10 Epoch 11-20

		1234567890	1234567890
eeg11			.EE.....
eeg12		..S.E.E.E.	.E.E..EE..
eeg13			
eeg14			
eeg15	X		
eeg16		E...	.E.EE.....
eeg17		...E.E.EE.	.EE.E....E
eeg18			
eeg19		.EEKKKKKKK	KKKKKKKKKM
eeg20		..E..EE...	

		1234567890	1234567890
eeg21		EE....	
eeg22		.S.E.....	..EE.EE.EE
eeg23		..F..E....	E.....
eeg24			
eeg25		EE.	E....
eeg26	X		
eeg27		EE..X.....	.E...EE...
eeg28		KXE...EE..	E
eeg29	X		
eeg30	X		

		1234567890	1234567890
eeg31	X		
eeg32	X		
eeg33		E..	
eeg34			
eeg35		E..EE.	S...E.....
eeg36		EEEEKM	KEEEEKEKSK
eeg37			
eeg38		.E.....EE.	E.....E..
eeg39		KKKKKKKKKK	KKKKKKKKKK
eeg40		EEEEEE.EE.	E....E...S

EEG I.D. Epoch 1-10 Epoch 11-20

```

                                1234567890 1234567890
eeg41      .....
eeg42      ....SE.E.. .....E..
eeg43      X .....
eeg44      X .....
eeg45      .....E .....
eeg46      .....E ..E.....E.
eeg47      KKKKKKKKKK KKKKKKKKKK
eeg48      KEE.KEKKKK KKKKKKKKKK
eeg49      X .....
eeg50      X .....

```

```

                                1234567890 1234567890
eeg51      .....XX.X .E.XXEXXEE
eeg52      E.....
eeg53      X ...E...E.. .E.EE....
eeg54      ...EEEEXE. EEEEEEEEEEE
eeg55      .....E E.E.....
eeg56      ...SS.E.K. ....KKKKE
eeg57      E...EE.... .SS..E....
eeg58      KKKKKKKKKK KKKKKKKKKK
eeg59      ...EE...EE E..EEEEEE.
eeg60      S.E.E.S.E. EE...S.E..

```

```

                                1234567890 1234567890
eeg61      X .....
eeg62      X .....
eeg63      .E.....
eeg64      KKK.EK.K.. EEKKKKKKKK
eeg65      .EEE..K.E. ....EMSEE.
eeg66      EEEEEEEEEEE EEE E.S..EE
eeg67      E..KKK.KKE E.SSESEKK
eeg68      .....E. S.E....E.E
eeg69      X .....
eeg70      KKKKKKKKKK KKKKKKKKKK

```

EEG I.D. Epoch 1-10 Epoch 11-20

```

1234567890 1234567890
eeg71      EEE.EEE.EE EEE.EEEEEE
eeg72      KKKKKKKKKK KKKKKKKKKK
eeg73      .J..... ..K...E...
eeg74      .....S .S.....
eeg75      EESEEEEEEE EEEEEEEE..
eeg76      ..... ..EESEEEE
eeg77      .....KE .SEK.EEEEE
eeg78      ..... ..PP..P
eeg79      BBBBBBBBBB BBBBBBBBBB
eeg80      .....E E.EE.....

```

```

1234567890 1234567890
eeg81      KKKKKKKKKK KKKKKKKKKK
eeg82      KKKKKKKKKK KKKKKKKKKK
eeg83      P.....E... ..PP
eeg84      KKKKKKKKKK KKKKKKKKKK
eeg85      KKKKKKKKKK KKKKKKKKKK
eeg86      ..... ..EE.
eeg87      BBBBBBBBBB BBBBBBBBBB
eeg88      E..EEEEEE. E...EEEE
eeg89      KKKKKKKKKK KKKKKKKKKK
eeg90      .EEE.EE..E ..E...EEE

```

```

1234567890 1234567890
eeg91      KKKKKKKKKK KKKKKKKKKK
eeg92      X .....
eeg93      KKKKKKKKKK KKKKKKKKKK
eeg94      ...EE.....
eeg95      .K.KK..... KKKKKKKK..
eeg96      KKKKKKKKKK KKKKKKKKKK
eeg97      E..E.E.... ..EEEE.
eeg98      ..... ..E.....E
eeg99      E.K....K.. E.K.K.KE..
eeg100     .....

```

EEG I.D. Epoch 1-10 Epoch 11-20

```

1234567890 1234567890
eeg101      KKKKKKKKKK KKKKKKKKKK
eeg102      S..SJ..... J....
eeg103      EEEEE..... E.E
eeg104      ...KEEKEK. KKEEEEE...
```

'ABNORMALS':

EEG I.D. Epoch 1-10 Epoch 11-20

```

1234567890 1234567890
eeg1A       ..... EE.E...SS.
eeg2A       E.JJEKS.S. S.KKEEE.EE
eeg3A       ...EM.MM.M ..S...M..M
eeg4A       ..... EEMMM
eeg5A       .EEE..... E....EE
eeg6A       .....M.M. EEEE.....
eeg7A       SKEESS.JJJ SSSSEKSKK.
eeg8A       .....E... ?..?....?.
eeg9A X     .....
eeg10A      ..E..... .ME...E..E
```

```

1234567890 1234567890
eeg11A      EEEEE.E.EE EEEE..E.EE
eeg12A      .....S. ....E...
eeg13A      ?.E.....?. ....?..
eeg14A      K.E..KK.E. ....
eeg15A      KS?SKMMMM MMKMEKKEMM
eeg16A      EEEMEE.SE. .E...S..K.
eeg17A      .....EEEE E.E.EE....
eeg18A      .....?... .????..?..
eeg19A      E.E.E..E.. ....
eeg20A      .E.E..EEEE ..EEEE...
```

EEG I.D. Epoch 1-10 Epoch 11-20

	1234567890	1234567890
eeg21A	SEMX..PE.S	SEMEPPPPMM
eeg22A	EEEEEE.ES.	EEEE.E..EE
eeg23A	EEEEEEEEEE	EEEEEEEEEE
eeg24A	S....	E..
eeg25A	E..KKKKKKK	KKKKKKKKKK
eeg26A	KEEEEEEEEEE	EEEEEEEEEE
eeg27A	EEEEEEEEEE	EEEEEEEEEE
eeg28A	..E...EE..	EE.....ES.
eeg29A X		
eeg30A	EEEEEEEEEE	EEEEEEEEEE

	1234567890	1234567890
eeg31A	EEEEEEEEEE	EEEEEEEEEE
eeg32A	SSSSSE.E.M	MEMMM..MMM
eeg33A	EMMMMMMMM	MMMMMMMMMM
eeg34A	.ME..E.E..	.E.....S
eeg35A	E.EEEEEEE	EEEEEEEEEE
eeg36A	EEEEEEEEEE	EEEEEEEEEE
eeg37A X		
eeg38A X		

These are the EEGs with artifact levels acceptable (the threshold function constant = 2.0) to calculate an average topographic map for each frequency band:

Available for class 'NORMALS', the EEGs included are:

eeg6	eeg8	eeg9	eeg10	eeg11
eeg12	eeg13	eeg14	eeg16	eeg17
eeg18	eeg20	eeg21	eeg22	eeg23
eeg24	eeg25	eeg27	eeg28	eeg33
eeg34	eeg35	eeg37	eeg38	eeg40
eeg41	eeg42	eeg45	eeg46	eeg48
eeg51	eeg52	eeg53	eeg55	eeg56
eeg57	eeg59	eeg60	eeg63	eeg65
eeg67	eeg68	eeg71	eeg73	eeg74
eeg76	eeg77	eeg78	eeg80	eeg83
eeg86	eeg90	eeg94	eeg95	eeg97
eeg98	eeg99	eeg100	eeg102	eeg103

There are a total of 60 EEGs.

Available for class 'ABNORMALS', the EEGs included are:

eeg1a	eeg2a	eeg3a	eeg4a	eeg5a
eeg6a	eeg8a	eeg10a	eeg11a	eeg12a
eeg13a	eeg14a	eeg16a	eeg17a	eeg18a
eeg19a	eeg20a	eeg21a	eeg22a	eeg23a
eeg24a	eeg26a	eeg28a	eeg30a	eeg31a
eeg34a	eeg35a	eeg36a	eeg37a	eeg38a

There are a total of 30 EEGs.

NOTE: NOT all 'ABNORMALS' are abnormal in every band. That is, there are 'ABNORMALS' which may be normal in a specific band. Such a case is referred to as a 'Normal ABNORMAL' in the appropriate band.

Appendix B

EEG Data Processing - M-files and Turbo C programs.

The following is the MatLab M-file 'psd.m' for computing the power spectral density for each epoch of each channel:

```
function [Px] = psd(X,scale)
%
% EXAMPLE: p = psd(X,scale);
%
% Returns the 96 x 380 matrix for the power spectral densities of
% each channel for each epoch. Make sure X is column oriented.
%
% X - the 256 x 380 data matrix of Sampled EEG data for each
% subject. There are 256 data samples/epoch and 20 epochs
% per channel. There are 19 channels of data. The data
% is as extracted by the 'extract.c' program.
%
% scale - the vector of scale factors for each channel. The scale
% factors are as extracted by the 'extract.c' program.
%
%
i = 1:20;
i = i+19 - 18;
j=1;
for k = 1:19 % Calibrate the epochs for each channel.
    eeg(i,:) = eeg(i,+)/scale(j);
    j = j+1;
    i = i+1;
end

k = 3; % k = the number of windows
m = 128; % m = the window size
slide = (256-m)/(k-1); % slide = the width to slide the window

index = 1:m; % index to the sliding window
beta = 4; % beta for the kaiser window
w = kaiser(m,beta); % define coefficients for sliding window
normalize = k*norm(w)^2/2; % normalizing factor
Px = zeros(96,380); % set aside space for accumulated power

for i = 1:k
    for j = 1:380
        Xw = w .* detrend( X(index,j), 0); % Vector of detrended windowed data
        P = fft(Xw,256);
        P = abs( P(1:96)).^2; % unscaled power
        Px(:,j) = Px(:,j) + P; % accumulate
    end
    index = index + slide; % slide the window
end

Px = Px(1:96,:)/normalize; % normalize result
```

The following is the MatLab M-file 'psdex.m' used to generate the example of Figure 3.10:

```
function [q] = psdex(X,epoch,channel,beta,eegid)
%
% EXAMPLE: p = psdex( X, epoch, channel,beta,eegid,cid);
%
% Creates the Spectral estimates based on 1, 3, and 7 kaiser windows
% with L/2 overlap as in Figure 3.9.
%
% X - the 256 x 380 data matrix of Sampled EEG data for each
%      subject. There are 256 data samples/epoch and 20 epochs
%      per channel. There are 19 channels of data. The data
%      is as extracted by the 'extract.c' program.
%
% epoch - a scalar, the epoch number.
% channel - a scalar, the channel number.
% beta - a scalar for kaiser window shape.
% eegid - a text string containing the I.D. of the EEG.
% cid - a text string containing the channel I.D.

k = 1; % k = the number of windows
m = 256; % m = the window size

index = 1:m; % index to the sliding window
w = kaiser(m,beta); % define coefficients for sliding window
normalize = k*norm(w)^2/2; % normalizing factor
Px = zeros(96,380); % set aside space for accumulated power

for i = 1:k
    for j = 1:380
        Xw = w .* detrend( X(index,j), 0); % Vector of detrended windowed data
        P = fft(Xw,256);
        P = abs( P(1:96)).^2; % unscaled power
        Px(:,j) = Px(:,j) + P; % accumulate
    end
    index = index + slide; % slide the window
end

Px = Px(1:96,:)/normalize; % normalize result
psd1 = Px(:,19*(epoch-1)+channel); % PSD for single window
```

```

k = 3; % k = the number of windows
m = 128; % m = the window size
slide = (256-m)/(k-1); % slide = the width to slide the window

index = 1:m; % index to the sliding window
w = kaiser(m,beta); % define coefficients for sliding window
normalize = k*norm(w)^2/2; % normalizing factor
Px = zeros(96,380); % set aside space for accumulated power

for i = 1:k
    for j = 1:380
        Xw = w .* detrend( X(index,j), 0); % Vector of detrended windowed data
        P = fft(Xw,256);
        P = abs( P(1:96)).^2; % unscaled power
        Px(:,j) = Px(:,j) + P; % accumulate
    end
    index = index + slide; % slide the window
end

Px = Px(1:96,:)/normalize; % normalize result
psd3 = Px(:,19*(epoch-1)+channel); % PSD for 3 windows

k = 7; % k = the number of windows
m = 64; % m = the window size
slide = (256-m)/(k-1); % slide = the width to slide the window

index = 1:m; % index to the sliding window
w = kaiser(m,beta); % define coefficients for sliding window
normalize = k*norm(w)^2/2; % normalizing factor
Px = zeros(96,380); % set aside space for accumulated power

for i = 1:k
    for j = 1:380
        Xw = w .* detrend( X(index,j), 0); % Vector of detrended windowed data
        P = fft(Xw,256);
        P = abs( P(1:96)).^2; % unscaled power
        Px(:,j) = Px(:,j) + P; % accumulate
    end
    index = index + slide; % slide the window
end

Px = Px(1:96,:)/normalize; % normalize result
psd7 = Px(:,19*(epoch-1)+channel); % PSD for 7 windows

```

```

i = 1:96;
i = 24*i/96;
plot(i,psd1,i,psd3,i,psd7);          % plot the results
ylabel('Power Density (square uV/Hz)');
xlabel('Frequency (Hz)');
title(['PSD for 1,3,&7 Windows (1/2 Overlap, Beta = ', num2str(beta),')' ]);
text(.5,.75,['Solid Line: 1 Window (L=256)']);
text(.5,.70,['Dashed Line: 3 Windows (L=128)']);
text(.5,.65,['Dotted Line: 7 Windows (L=64)']);
text(.7,.45,['EEG id: ',eegid],'sc');
text(.7,.40,['Epoch ',int2str(epoch),' Channel ',cid],'sc');
grid

```

The following is the Turbo C program 'extract.c' used to extract the sampled EEG data from the BrainLab 'eegid.ego':

```

/* ***** */
/*
/*  EXTRACT.C -  Extracts the Patient I.D. and scale information */
/*               from a '.EGO' BrainLab file. The result is      */
/*               stored in a MatLab readable '.RAW' file.         */
/* ***** */

#include <stdio.h>
#include <io.h>
#include <math.h>
#include <ctype.h>

typedef struct{
    long    type;          /* MOPT type, p. 3-120 of MatLab Manual */
    long    mrows;         /* row dimension                        */
    long    ncols;         /* column dimension                    */
    long    imagf;         /* imaginary flag NOT set if real      */
    long    namlen;        /* length of the matrix name (patient) */
} Header;                  /*  ^-- must include a '\0'           */

/* ***** */
/*  Invocation:  'extract EEGID.EGO' */
/*
/*               Extracted data is stored in 'EEGID.RAW' */
/* ***** */

main(argc, argv)
int    argc;
char   *argv[];
{
    FILE    *fopen(), *fin, *fout;
    char    c, fname[30], text[20]="scale";
    int     epoch,i, j;

    /* allocate space for EEG data storage. */

    short int data[4864], ch1[255], ch2[255], ch3[255], ch4[255], ch5[255],
        ch6[255], ch7[255], ch8[255], ch9[255], ch10[255], ch11[255],
        ch12[255], ch13[255], ch14[255], ch15[255], ch16[255],
        ch17[255], ch18[255], ch19[255];
    float scale[19];

    Header head;          /* allocate space for matlab header */

```

```

if (argc == 2){
    strcpy( fname, argv[1]);
    for( i=0; i<9; i++){
        if( fname[i] == '.'){
            j = i;
            i = 9;
        }
    }
    fname[j] = '.';
    fname[j+1] = '\0';
    strcat( fname, "raw");
    printf("File '%s' will be overwritten. Proceed? (^C to exit)",
           fname);
    scanf("%c", &c);

    fin = fopen( argv[1], "r+b"); /* open 'EEGFILE.EGO' as input */
    fout = fopen( fname, "w+b"); /* open 'EEGFILE.RAW' as output*/

    /* ***** */
    /* Define Matlab header. REfer to pages 3-120 to 3-122      */
    /* inclusive of Matlab manual.                               */
    /* ***** */

    head.type = 0;          /* 8 byte fp --> p 3-121. MOPT      */
    head.mrows = 1;         /* 1 row               */
    head.ncols = 16;        /* 8 X 16 = 128 bytes  */
    head.imagf = 0;         /* real data only      */
    head.namlen = 4;        /* Patient I.D. information */

    fwrite( &head, sizeof(Header), 1, fout); /* output header */
    putc( 'P', fout);
    putc( 'I', fout); /* output vector name */
    putc( 'D', fout);
    putc( '\0', fout);

    fread( data, 1024, 1, fin); /* seek Patient I.D. location */
    fread( data, 128, 1, fin); /* read Patient I.D. */
    fwrite( data, 128, 1, fout); /* store Patient I.D. */
    /* note: now at scale info (1152) */

```

```

/* ***** */
/* Define Matlab header. REfer to pages 3-120 to 3-122 */
/* inclusive of MatLab manual. */
/* ***** */

head.type = 10;          /* 4 byte fp --> p 3-121. MOPT */
head.mrows = 1;          /* 1 row */
head.ncols = 19;          /* 19 gain settings */
head.imagf = 0;          /* real data only */
head.namlen = 6;          /* vector name = "scale" */

fwrite( &head, sizeof(Header), 1, fout); /* output header */
for( i=0; i<(head.namlen-1); i++){ /* output matrix name */
    putc( text[i], fout);
} /* terminated by a NULL */
putc( '\0', fout);

fread( scale,76,1,fin); /* read scale info */
for( i=0; i<19; i++) /* convert scale to divisor */
    scale[i] = (-1*scale[i]/6) + 124.29;
/* which yields data in uV */

fwrite( scale,76,1,fout); /* store scale info */

/* ***** */
/* Define Matlab header. REfer to pages 3-120 to 3-122 */
/* inclusive of matlab manual. */
/* ***** */

head.type = 130;          /* 2 byte int --> p 3-121. */
head.mrows = 380;          /* 20 epochs * 19 channels per */
head.ncols = 256;          /* 256 elements per channel */
head.imagf = 0;          /* real data only */
head.namlen = strlen( fname) -3; /* matric name length +1 */

fwrite( &head, sizeof(Header), 1, fout); /* output header */
for( i=0; i<(head.namlen-1); i++){ /* output matrix name */
    putc( fname[i], fout);
} /* terminated by a NULL */
putc( '\0', fout);

fread( data,436,1,fin); /* seek the beginning of the data */

```

```

for( epoch=0; epoch <20; epoch++){
  fread( data,9728,1,fin);
  for( i=j=0; i < 4864; i+=19){
    ch1[j] = data[i];          /* read the data in          */
    ch2[j] = data[i+1];
    ch3[j] = data[i+2];
    ch4[j] = data[i+3];
    ch5[j] = data[i+4];
    ch6[j] = data[i+5];
    ch7[j] = data[i+6];
    ch8[j] = data[i+7];
    ch9[j] = data[i+8];
    ch10[j] = data[i+9];
    ch11[j] = data[i+10];
    ch12[j] = data[i+11];
    ch13[j] = data[i+12];
    ch14[j] = data[i+13];
    ch15[j] = data[i+14];
    ch16[j] = data[i+15];
    ch17[j] = data[i+16];
    ch18[j] = data[i+17];
    ch19[j++] = data[i+18];
  }
  fwrite(ch1,512,1,fout); /* write the data out          */
  fwrite(ch2,512,1,fout);
  fwrite(ch3,512,1,fout);
  fwrite(ch4,512,1,fout);
  fwrite(ch5,512,1,fout);
  fwrite(ch6,512,1,fout);
  fwrite(ch7,512,1,fout);
  fwrite(ch8,512,1,fout);
  fwrite(ch9,512,1,fout);
  fwrite(ch10,512,1,fout);
  fwrite(ch11,512,1,fout);
  fwrite(ch12,512,1,fout);
  fwrite(ch13,512,1,fout);
  fwrite(ch14,512,1,fout);
  fwrite(ch15,512,1,fout);
  fwrite(ch16,512,1,fout);
  fwrite(ch17,512,1,fout);
  fwrite(ch18,512,1,fout);
  fwrite(ch19,512,1,fout);
}
fclose(fin);
fclose( fout);
}
else {
  printf( "\nUsage:  extract EEGID.EGO\n");
}

```


The following is the Turbo C program 'estimate.c' which implements the artifact rejection and does the averaging for the final PSD estimates:

```

/* ***** */
/* ESTIMATE.C - implements the artifact rejection and */
/* spectral averaging of the first 10 artifact */
/* free epochs for each channel. */
/* */
/* The spectral estimates are from the 'EEGID.POW' as */
/* saved after the execution of the MatLab M-file 'psd.m'. */
/* The artifact in the epochs for the EEG are looked up */
/* from the 'ART.TAB' file which is similar to APPENDIX A. */
/* The resulting final PSD estimate for each channel is */
/* stored in the 'EEGID.EST' file. */
/* ***** */

#include <stdio.h>
#include <io.h>
#include <math.h>
#include <ctype.h>

#define EPOCH 20
#define CHANNEL 19
#define THRESH 2.0

typedef struct{
    long type; /* MOPT type, p. 3-120 */
    long mrows; /* row dimension */
    long ncols; /* column dimension */
    long imagf; /* imaginary flag NOT set if real */
    long namlen; /* length of the matrix name (patient */
} Header; /* -- must include a '\0' */

typedef struct{
    char id[4]; /* patient (PID) id # (1a-38a, 1-102) */
    char flag; /* flag to check if data usable */
    char ep[80]; /* buffer for artifact rating */
} Artifact;

/* ***** */
/* Invocation: 'estimate EEGID.POW' */
/* */
/* Extracted data is stored in 'EEGID.EST' */
/* ***** */

```

```

main(argc, argv)
int    argc;
char   *argv[];
{
    FILE          *fopen(), *dfnames, *fin, *fout;
    static char    c[3], t[25], pid[20], comment[50], fname[30];
    int            i, j, k, ep, ch, ctr;
    static short float data[128], den, ratio,
    avg[CHANNEL][6], energy[CHANNEL][6];

    Header header;
    Artifact art;

    if (argc == 2){

        /* ***** */
        /* get the artifact record for the EEG */
        /* ***** */

        fin = fopen( "ART.TAB", "r");
        strcpy(fname, argv[1]);
        printf("%s ", fname);
        j=0;
        for(i=0; i < strlen(fname) ;i++){
            if( isdigit(fname[i])){
                t[j] = fname[i];
                j++;
            }
        }
        t[j] = '\0';
        k = atoi(t);
        t[0] = '\0';
        printf("%3d", k);
        for(i=0; i<k; i++)
            fgets(&art.id[0], 80, fin);
        fclose(fin);
        printf(" %-30s\n", &(art.id[0]));
    }
}

```

```

/* ***** */
/* get the spectral info from 'EEGID.POW' */
/* ***** */

fin = fopen( argv[1], "r+b");
strcpy( fname, argv[1]);
i=0;
j = 9;
while( i<9){
    if( fname[i] == '.'){
        j = i;
        i = 9;
    }
    i++;
}
fname[j] = '.';
fname[j+1] = '\0';
strcat( fname, "est");
fout = fopen(fname, "w+b");

fread( &header, sizeof(Header), 1, fin);
fread( data, header.namlen, 1, fin);
/* fwrite( &header, sizeof(Header),1,fout); /* echo the header */
/* fwrite( data, header.namlen, 1, fout); /* PID */
fread( data, 128,1,fin);
/* fwrite( data, 128,1,fout); /* PID info */

fread( &header, sizeof(Header), 1, fin); /*Peel off matrix name*/
fread( pid, header.namlen, 1, fin); /* pid = 'PEEG#' */
pid[0] = 'b'; /* Change to 'BEEG#' */
strcpy( comment, "! ");
strcat( comment, pid); /* '! BEEG#' */
strcat( comment, " "); /* '! BEEG#' */
strcpy( c, " ");

```

```

/* ***** */
/* Perform the averaging and artifact rejection. */
/* ***** */

ctr = 0; /* # of epochs in average */

for( i=0; i<EPOCH; i++){
    fread( energy, CHANNEL*6*4, 1, fin); /* Artifact Rejection */
    ratio = (energy[0][0]+energy[8][0])/(energy[4][0]+energy[12][0]);
    if( (art.flag == ' ') &&
        (art.ep[i] == '.' || (art.ep[i] == 'E' && ratio < THRESH))){
        ctr += 1;
        c[0] = '0' + (i+1)%10;
        strcat( t, c);
        if( ctr <= 10){
            for( j=0; j<CHANNEL; j++){
                for( k=0; k<6; k++){
                    avg[j][k] += energy[j][k];
                }
            }
        }
        itoa( ctr, c, 10);
        strcat(comment,c);
        strcat( comment, ": ");
        strcat( comment,t);
        if( ctr <= 10)
            den = ctr/1.0; /* This flags failure to */
        else /* find 10 artifact-free */
            den = 10.0; /* Epochs */
        if( ctr == 0)
            den = 1.0;

        for( j=0; j<CHANNEL; j++) /* Average has 10 Epochs */
            for( k=0; k<6; k++)
                avg[j][k] /= den;
    }
}

```

```

/* ***** */
/* Define Matlab header. REfer to pages 3-120 to 3-122 */
/* inclusive of Matlab manual. */
/* ***** */

header.type = 140;          /* 8 byte fp --> p 3-121. MOPT */
header.mrows = 25;          /* 25 rows */
header.ncols = 1;           /* 1 column */
header.imagf = 0;           /* real data only */
header.namlen = 2;          /* matrix name length +1 */

fwrite( &header, sizeof(Header), 1, fout); /* output header */
putc( 'C', fout);
putc( '\0', fout);
fwrite( comment, 50, 1, fout);

/* ***** */
/* Define Matlab header. REfer to pages 3-120 to 3-122 */
/* inclusive of Matlab manual. */
/* ***** */

header.type = 110;          /* Single prec. --> p 3-121. */
header.mrows = 19;          /* 1 Estimate * 19 channels per */
header.ncols = 6;           /* 6 energy bands per channel */
header.imagf = 0;           /* real data only */
header.namlen = strlen( pid) +1; /* matrix name length +1 */

fwrite( &header, sizeof(Header), 1, fout); /* output header */
for( i=0; i<(header.namlen-1); i++){ /* output matrix name */
    putc( pid[i], fout);
}                                     /* terminated by a NULL */
putc( '\0', fout);
fwrite( avg, CHANNEL*6*4, 1, fout);

fclose(fin);
fclose(fout);
}
else {
    printf(
        "\nUsage: 'estimate EEGID.POW' (file ART.TAB must exist)\n");
}
}

```

Appendix C

The PNN Turbo C Program.

```

/* ***** */
/* PNM.C - The Probabilistic Neural Network (based on Parzen Estimation). */
/*
/* This program implements a Parzen Estimation of the probability density */
/* function for two classes (Normal & Abnormal): */
/*
/*
/*          Nn
/*          ---
/*          \      T      T      T      2
/*          ( (2.0X Xn - X X - Xn Xn )/(2.0Sn ) )
/* fn(X) = (1.0/Nn)* / e      i      i      i
/*          ---
/*          i=1
/*
/*          Na
/*          ---
/*          \      T      T      T      2
/*          ( (2.0X Xa - X X - Xa Xa )/(2.0Sa ) )
/* fa(X) = (1.0/Na)* / e      i      i      i
/*          ---
/*          i=1
/*
/* where:
/*   fn(X) = the pdf for the Normal subjects
/*   fa(X) = the pdf for the Abnormal subjects
/*   Nn    = the number of Normals in the training file (SIZEOFABN)
/*   Na    = the number of Abnormals in the training file (SIZEOFNORM)
/*   Xn    = A Feature Vector from the Normal " " (xn[i])
/*   Xa    = A Feature Vector from the Abnormal " " (xa[i])
/*   Sn    = Standard Deviation for the Normal Vectors (sign)
/*   Sa    = Standard Deviation for the Abnormal Vectors (siga)
/*
/* note:          (-b)          (-b)
/*   Sa( Na) = a*Na    and   Sn( Na) = a*Nn
/*
/*   where:
/*           (1/2)
/*   a = a constant gain          (a = gain )
/*   b = the factor for dependency on N   (b = factor/2 )
/*
/* ***** */

```

```

#include <stdlib.h>
#include <stdio.h>
#include <io.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include <ctype.h>

#define NNORM      100          /* Matrix dimensioning sizes */
#define NABN       50
#define TOTAL      200

#define FEATURES   8           /* 8-dimensional feature vector */

#define SIZEOFNORM  30          /* # of Normals in training file */
#define SIZEOFABN   15          /* # of Abnormals in training file*/
#define TRAIN_CASES 9           /* number of training files */
#define NUM_PROBS   40          /* number of Pn/Pa ratios */

/* ***** */
/* Each EEG has the following record associated with it: */
/* ***** */

typedef struct{
    short float    x[FEATURES]; /* The feature vector */
    short float    scale;       /* scale factor for features */
    char           type;        /* 'N', 'A', or 'T' */
    short float    fa;          /* pdf evaluated at x for abn */
    short float    fn;          /* pdf evaluated at x for norm */
    char           class;       /* 'N' or 'A' */
    char           name[12];     /* Patient ID */
} Record;

static Record      d[TOTAL];    /* Stores 'all.nna' in mem */

static short float xn[NNORM][FEATURES], /* Normal Training Vectors */
                  xa[NABN][FEATURES], /* Abnormal Training Vectors */
                  in[NNORM],           /* Indices to Normals */
                  ia[NABN];            /* Indices to Abnormals */

```



```

static int  num_norm,          /* # of Normal Normals */
            num_abn,          /* # of Abnormal Abnormals */
            num_na,           /* # of Normal Abnormals */
            num_abn_train,    /* # of abn training vects */
            num_norm_train;   /* # of norm train vectors */

static int  class_nn,         /* Normals classed as norm */
            class_an,         /* Normals classed as abn */
            class_aa,         /* Abnormals classed as abn */
            class_na,         /* Abnormals classed as norm */
            class_nna,        /* Normal abnormals classed */
                                /* as normal */
            class_ana;        /* Normal abnormals classed */
                                /* as abnormal */

static float gain,            /* Gain for variance */
            factor,          /* Exponent for variance */
            siga,            /* Std. Dev. for abnormals */
            sign,            /* Std. Dev. for normals */
            vara,            /* Variance for abnormals */
            varn;            /* variance for normals */

static float probb[NUM_PROBS] =
    {0, 0.25, 0.5, 0.75,
     1, 1.25, 1.5, 1.75,
     2, 2.25, 2.5, 2.75,
     3, 3.25, 3.5, 3.75,
     4, 4.25, 4.5, 4.75,
     5, 5.25, 5.5, 5.75,
     6, 6.25, 6.5, 6.75,
     7, 7.25, 7.5, 7.75,
     8, 10, 12, 14,
     16, 18, 20, 25};
/* vector of Pn/Pa values */

```

```

static char nn[TOTAL],      /* indices to correct normals */
          an[TOTAL],      /* "    to incorrect normals */
          aa[TOTAL],      /* "    to correct  abnormals */
          na[TOTAL],      /* indices to incorrect abnormals */
          nna[TOTAL],     /* "    to correct  normal abns*/
          ana[TOTAL];     /* "    to incorrect normal abns*/

/* ***** */
/* Invocation has format: PNN 1.0 0.75 Resultfile [-v] */
/* */
/*   where: gain = 1.0, */
/*          factor = 0.75, */
/*          'Resultfile.m' will contain MATLAB results, */
/*          'Resultfile.ver' is optionally produced. */
/* */
/* This program implements the PNN Classifier and, as presented here, */
/* builds nine training files from the 60 Normal and 25 Abnormal */
/* Delta band relative asymmetrical power feature vectors. The */
/* program can be easily modified to implement the 'leave-one-out' */
/* method for estimates of the true sensitivity and specificity. */
/* */
/* The performance measures are placed in a MatLab readable file */
/* for graphing and analysis purposes. An optional 'verbose' file */
/* '.ver' which explicitly lists the classification of EACH EEG for */
/* each Pn/Pa value can be produced. It takes the form of a large */
/* text file. */
/* ***** */

```

```

main( argc, argv)
int    argc;
char   *argv[];
{
    FILE    *fopen(),*fout1,*fout2;
    char    fname[20],f1[20],f2[20],ver;
    int     loop,num_recs,p;
    void     echo_data();

    gain = 1.0;                                /* Defaults      */
    factor = 1.0;
    strcpy(fname, "results");

    if(argc >= 4){

        gain =  atof(argv[1]);
        factor = atof(argv[2]);
        strcpy(fname, argv[3]);
        if( argc == 5)
            ver = 'Y';                          /* Flag indicates 'verbose' mode.      */

        num_norm_train = SIZEOFNORM;
        num_abn_train =  SIZEOFABN;

        siga = sqrt(gain)/( pow((SIZEOFABN), (0.5*factor)) );
        sign = sqrt(gain)/( pow((SIZEOFNORM),(0.5*factor)) );
        vara = siga*siga;
        varn = sign*sign;

        if(ver == 'Y'){                          /* Open 'Resultfile.ver' if needed.    */
            strcpy(f1,fname);
            strcat(f1,".ver");
            fout1 = fopen(f1, "w");
        }

        strcpy(f2,fname);                        /* Open 'Resultfile.m'.                */
        strcat(f2,".m");
        fout2 = fopen(f2, "w");
    }
}

```

```

strcpy(fname,"all.nna");
num_recs = read_data(fname); /* Read EEG records from 'all.nna'. */
echo_data(num_recs);

for(loop=0; loop<TRAIN_CASES; loop++){ /* For each Training File */
    setup_training_file(loop); /* Build training file. */
    if( ver == 'Y')
        head_verbose(fout1,loop); /* Begin output files */
        head_matlab(fout2,loop);
        recall_all(num_recs); /* Apply test vectors */
        for(p=0; p < NUM_PROBS; p++){
            gather_stats(p,num_recs); /* gather performance statistics */
            if( ver == 'Y')
                update_verbose(fout1,p); /* update output files */
                update_matlab(fout2);
        }
        if( ver == 'Y')
            tail_verbose(fout1); /* Write trailers to output files */
            tail_matlab(fout2);
    }
    complete_matlab(fout2); /* Close files */
    fclose( fout1);
    fclose(fout2);
}
else{
    printf("\n\nUsage: PNW 1.0 0.75 Resultfile [-v]");
    printf("\n\n      where: gain = 1.0,");
    printf("\n      factor = 0.75,");7z
    printf("\n      'Resultfile.m' will contain MATLAB results,");
    printf("\n      'Resultfile.ver' is optionally produced.\n\n");
}
} /* end of main() */

```

```

/* ***** */
/* Read in the EEG records from 'all.nna'. See 'all.nna' for record */
/* formats. This completes the training of the PNW. */
/* ***** */

read_data(f)
char *f;
{
    FILE *fopen(), *fin;
    int i,j,k;
    float b1,b2,b3;
    char c;

    i = 0;
    fin = fopen( f, "r");
    while( (fscanf(fin, "%f", &(d[i].x[0])) != EOF )){
        for(j = 1; j< FEATURES; j++){
            fscanf(fin, "%f", &(d[i].x[j]));
            while( (c = fgetc(fin)) != '&');
            fscanf(fin, "%f %f %f", &b1, &b2, &b3);
            while( (c = fgetc(fin)) != '!');
            c = fgetc(fin); /* peel off ' ' */
            for(j=0; (c = fgetc(fin)) != ' '; j++) /* get the name */
                d[i].name[j] = c;
            d[i].name[j] = '\0';
            k = 1;
            while(k){ /* advance to next record */
                c = fgetc(fin);
                if ( (c == '\n') || (c == EOF))
                    k = 0;
            }
            if( b1 > 0.1){ /* No error checking!!! */
                d[i].type = 'N'; /* Normal Normal */
                in[num_norm] = i;
                num_norm++;
            }
        }
    }
}

```

```

    if( b2 > 0.1){
        d[i].type = 'A';
        ia[num_abn] = i;
        num_abn++;
    }
    if( b3 > 0.1){
        d[i].type = 'T';
        num_na++;
    }
    i++;
}
fclose( fin);
return(i);
}

/* ***** */
/* Echo the records which are read (as a check). */
/* ***** */

void echo_data(num_recs)
int num_recs;
{
    FILE *fopen(), *fout;
    int i,j;

    fout = fopen( "ECHO.DAT", "w");
    fprintf(fout,"A total of %d records were read as follows:\n\n",num_recs);
    for(i=0; i<num_recs; i++){
        if(i%10 == 0)
            fprintf(fout,"\n");
        for(j=0; j<FEATURES; j++){
            fprintf(fout, "%8.5f ",d[i].x[j]);
        }
        fprintf(fout, " %c %s\n",d[i].type,&(d[i].name[0]));
    }
    fclose(fout);
}

```

```

/* ***** */
/* Build the 9 training files. The Normals and Abnormals are both */
/* split into 3 groups. The training files consist of the 9      */
/* combinations of these groups.                                   */
/* ***** */

setup_training_file( i)
int i;
{
    int    incn, inca, half, half_plus, next;

    if(i == 0){
        make_norm_train_file(1,      num_norm_train,1); /* First group */
        make_abn_train_file(1,      num_abn_train, 1); /* First group */
    }
    if(i ==1){
        make_norm_train_file(1,      num_norm_train,1); /* First Group */
        inca = num_abn - num_abn_train;
        make_abn_train_file(1+inca,  num_abn_train, 1); /* Second Group */
    }
    if(i ==2){
        make_norm_train_file(1,      num_norm_train,1); /* First Group */
        half = num_abn_train/2;
        half_plus = (num_abn_train+1)/2;
        inca = num_abn - half_plus;
        next = (num_abn_train+2)/2;
        make_abn_train_file(1,      half      ,1); /* Third Group */
        make_abn_train_file(1+inca,  half_plus,  next);
    }
    if(i ==3){
        incn = num_norm - num_norm_train;
        make_norm_train_file(1+incn,  num_norm_train,1); /* Second Group */
        make_abn_train_file(1,      num_abn_train,1);
    }
    if(i ==4){
        incn = num_norm - num_norm_train;
        make_norm_train_file(1+incn,  num_norm_train,1); /* Second Group */
        inca = num_abn - num_abn_train;
        make_abn_train_file(1+inca,  num_abn_train,1), /* Second Group */
    }
    if(i ==5){
        incn = num_norm - num_norm_train; /* Second Group */
        make_norm_train_file(1+incn,  num_norm_train,1);
        half = num_abn_train/2;
        half_plus = (num_abn_train+1)/2;
        inca = num_abn - half_plus;
        next = (num_abn_train+2)/2;
        make_abn_train_file(1,      half      ,1); /* Third Group */
    }
}

```

```

    make_abn_train_file(1+inca, half_plus, next);
}
if(i ==6){
    half = num_norm_train/2;
    half_plus = (num_norm_train+1)/2;
    incn = num_norm - half_plus;
    next = (num_norm_train+2)/2;
    make_norm_train_file(1, half, 1); /* Third Group */
    make_norm_train_file(1+incn, half_plus, next);
    make_abn_train_file(1, num_abn_train,1); /* First Group */
}
if(i ==7){
    half = num_norm_train/2;
    half_plus = (num_norm_train+1)/2;
    incn = num_norm - half_plus;
    next = (num_norm_train+2)/2;
    make_norm_train_file(1, half, 1); /* Third Group */
    make_norm_train_file(1+incn, half_plus, next);
    inca = num_abn - num_abn_train;
    make_abn_train_file(1+inca, num_abn_train,1); /* Second Group */
}
if(i ==8){
    half = num_norm_train/2;
    half_plus = (num_norm_train+1)/2;
    incn = num_norm - half_plus;
    next = (num_norm_train+2)/2;
    make_norm_train_file(1, half, 1); /* Third Group */
    make_norm_train_file(1+incn, half_plus, next);
    half = num_abn_train/2;
    half_plus = (num_abn_train+1)/2;
    inca = num_abn - half_plus;
    next = (num_abn_train+2)/2;
    make_abn_train_file(1, half, 1); /* Third Group */
    make_abn_train_file(1+inca, half_plus, next);
}
}

/* ***** */
/* Routines to help build training files: */
/* ***** */

make_norm_train_file(start,how_many,spot)
int start,how_many, spot;
{
    int i,j;

    spot--;

```



```

    for(i = start-1; i<(start+how_many-1); i++){
        for(j = 0; j<FEATURES; j++){
            xn[spot][j] = d[ in[i]].x[j];
        }
        spot++;
    }
}

make_abn_train_file(start,how_many,spot)
int    start,how_many, spot;
{
    int i,j;

    spot--;
    for(i = start-1; i<(start+how_many-1); i++){
        for(j = 0; j<FEATURES; j++){
            xa[spot][j] = d[ ia[i]].x[j];
        }
        spot++;
    }
}

/* ***** */
/* Routines to place headers in output files. */
/* ***** */

head_verbose(fout1,loop)
FILE    *fout1;
int     loop;
{
    fprintf(fout1,"For Training File %d (Gain = %5.2f, Exp. = %5.2f):\n\n",
        loop, gain, factor);
}

head_matlab(fout2,loop)
FILE    *fout2;
int     loop;
{
    fprintf(fout2,"t%db = [",loop);
}

```

```

/* ***** */
/* Apply the test vectors to the PHM: */
/* ***** */

recall_all(num_recs)
int    num_recs;
{
    int i,j,k,l;
    float    fa,fn,exponent;

    for(i=0; i<num_recs; i++){          /* For ALL Patients      */
        fa = fn = 0;
        for(j=0; j<num_abn_train; j++){ /* For each ABM train vec */
            exponent = 0;
            for(k=0; k<FEATURES; k++){   /* take the dot products */
                exponent += 2.0* d[i].x[k] * xa[j][k];
                exponent -=      d[i].x[k] * d[i].x[k];
                exponent -=      xa[j][k] * xa[j][k];
            }
            exponent /= (2.0*vara);        /* (exponent)/(2.0*o^2) */
            fa += exp(exponent);           /* sum the result        */
        }
        for(j=0; j<num_norm_train; j++){ /* For each NORM train vec*/
            exponent = 0;
            for(k=0; k<FEATURES; k++){   /* take the dot products */
                exponent += 2.0* d[i].x[k] * xn[j][k];
                exponent -=      d[i].x[k] * d[i].x[k];
                exponent -=      xn[j][k] * xn[j][k];
            }
            exponent /= (2.0*varn);        /* (exponent)/(2.0*o^2) */
            fn += exp(exponent);           /* sum the result        */
        }
        d[i].fa = (1.0/num_abn_train)*fa; /* Store the results.    */
        d[i].fn = (1.0/num_norm_train)*fn;
    }
}

```

```

/* ***** */
/* Gather the sensitivity and specificity for each value of Pn/Pa. */
/* ***** */

gather_stats(p,num_recs)
int    p,num_recs;
{
    int    i,j,k;

    class_nn = 0;                /* Reset counters */
    class_an = 0;
    class_aa = 0;
    class_na = 0;
    class_nna = 0;
    class_ana = 0;

    for(i=0; i<(num_recs+1); i++){ /* Reset indices to EEG records */
        nn[i] = -1;
        an[i] = -1;
        aa[i] = -1;
        na[i] = -1;
        nna[i] = -1;
        ana[i] = -1;
    }
    for(i=0; i<num_recs; i++){ /* For each EEG record */

        if( d[i].fa >= (prob[p]*d[i].fn) ) /* Determine classification */
            d[i].class = 'A';
        else
            d[i].class = 'N';

        switch (d[i].type){ /* Tabulate result */
            case 'N':
                if( d[i].class == 'N'){
                    nn[class_nn++] = i;
                }
                else {
                    an[class_an++] = i;
                }
                break;
            case 'A':
                if( d[i].class == 'A'){
                    aa[class_aa++] = i;
                }
                else {
                    na[class_na++] = i;
                }
                break;
        }
    }
}

```

```
case 'T':
    if( d[i].class == 'W'){
        nna[class_nna++] = i;
    }
    else {
        ana[class_ana++] = i;
    }
}
}
```

```

/* ***** */
/* Update the 'Resultfile.ver' for a given value of Pn/Pa. */
/* ***** */

update_verbose(fout1,p)
FILE *fout1;
int p;
{
    int i;

    fprintf(fout1,"Pn = %5.2f Pa:\n\n",prob[p]);
    fprintf(fout1,"NORMALS classified as NORMAL: ");
    for(i=0; i < class_nn; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ nn[i]].name[4]));
    }
    fprintf(fout1,"\n\nNORMALS classified as ABNORMAL: ");
    for(i=0; i < class_an; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ an[i]].name[4]));
    }
    fprintf(fout1,"\n\nABNORMALS classified as ABNORMAL: ");
    for(i=0; i < class_aa; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ aa[i]].name[4]));
    }
    fprintf(fout1,"\n\nABNORMALS classified as NORMAL: ");
    for(i=0; i < class_na; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ na[i]].name[4]));
    }
    fprintf(fout1,"\n\nNORMAL ABNORMALS classified as NORMAL: ");
    for(i=0; i < class_nna; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ nna[i]].name[4]));
    }
    fprintf(fout1,"\n\nNORMAL ABNORMALS classified as ABNORMAL: ");
    for(i=0; i < class_ana; i++){
        if(i%20 == 0)
            fprintf(fout1,"\n");
        fprintf(fout1,"%4s",&(d[ ana[i]].name[4]));
    }
}

```

```
fprintf(fout1,
  "\n\nIn Summary: %3d/%3d Normals,
                    %3d/%3d Abnormals,
                    and %3d/%3d Normal Abnormals.",
  class_nn,num_norm ,class_aa,num_abn ,class_nna,num_na);
fprintf(fout1,
  "\n-----\n\n\n");
}
```

```

/* ***** */
/* Update the 'Resultfile.m' file for a given value of Pn/Pa.      */
/* ***** */

update_matlab(fout2)
FILE    *fout2;
{
    fprintf(fout2, "\t%4d %4d %4d\n", class_nn, class_aa, class_nna);
}

/* ***** */
/* Routines to complete the trailers for the output files.        */
/* ***** */

tail_verbose(fout1)
FILE    *fout1;
{
    fprintf(fout1, "\f");
}

tail_matlab(fout2)
FILE    *fout2;
{
    fprintf(fout2, "\t\t\t\t;\n");
}

complete_matlab( fout1)
FILE    *fout1;
{
    int    i;

    fprintf(fout1, "d = [");
    for(i=0; i<NUM_PROBS; i++){
        fprintf(fout1, "\t%4d %4d %4d\n", num_norm, num_abn, num_na);
    }
    fprintf(fout1, "\t\t\t\t;\n");
    fprintf(fout1, "p=[");
    for(i=0; i<NUM_PROBS; i++){
        fprintf(fout1, "%4.2f ", prob[i]);
    }
    fprintf(fout1, "];\n\ngain = %8.4f;\n\nfactor = %8.4f;", gain, factor);
}

```

Appendix D

Simulation Set 1 - Apparent Performance Graphs

Figure D.1: Apparent Performance Graph - Training File T1.

Figure D.2: Apparent Performance Graph - Training File T2.

Figure D.3: Apparent Performance Graph - Training File T3.

Figure D.4: Apparent Performance Graph - Training File T4.

Figure D.5: Apparent Performance Graph - Training File T5.

Figure D.6: Apparent Performance Graph - Training File T6.

Figure D.7: Apparent Performance Graph - Training File T7.

Figure D.8: Apparent Performance Graph - Training File T8.

Figure D.9: Apparent Performance Graph - Training File T9.

Appendix E

Simulation Set 2 - Apparent Performance Graphs

Figure E.1: Apparent Performance Graph - Training File T1B.

Figure E.2: Apparent Performance Graph - Training File T2B.

Figure E.3: Apparent Performance Graph - Training File T3B.

Figure E.4: Apparent Performance Graph - Training File T4B.

Figure E.5: Apparent Performance Graph - Training File T5B.

Figure E.6: Apparent Performance Graph - Training File T6B.

Figure E.7: Apparent Performance Graph - Training File T7B.

Figure E.8: Apparent Performance Graph - Training File T8B.

Figure E.9: Apparent Performance Graph - Training File T9B.

END

24.12.92

FIN