Classification of ADHD and Non-ADHD Using AR Models and Machine Learning Algorithms

Juan Lopez Marcano

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Electrical Engineering

A. A. (Louis) Beex, Chair
Scott Bailey
JoAnn Paul

October 26, 2016
Blacksburg, Virginia

Keywords: EEG, ADHD, Classification, Machine Learning, KNN, SVM, GMM, Autoregressive Coefficients

Classification of ADHD and Non-ADHD Using AR Models and Machine Learning Algorithms

Juan Lopez Marcano

ABSTRACT

As of 2016, diagnosis of ADHD in the US is controversial. Diagnosis of ADHD is based on subjective observations, and treatment is usually done through stimulants, which can have negative side-effects in the long term. Evidence shows that the probability of diagnosing a child with ADHD not only depends on the observations of parents, teachers, and behavioral scientists, but also on state-level special education policies. In light of these facts, unbiased, quantitative methods are needed for the diagnosis of ADHD. This problem has been tackled since the 1990s, and has resulted in methods that have not made it past the research stage and methods for which claimed performance could not be reproduced.

This work proposes a combination of machine learning algorithms and signal processing techniques applied to EEG data in order to classify subjects with and without ADHD with high accuracy and confidence. More specifically, the K-nearest Neighbor algorithm and Gaussian-Mixture-Model-based Universal Background Models (GMM-UBM), along with autoregressive (AR) model features, are investigated and evaluated for the classification problem at hand. In this effort, classical KNN and GMM-UBM were also modified in order to account for uncertainty in diagnoses.

Some of the major findings reported in this work include classification performance as high, if not higher, than those of the highest performing algorithms found in the literature. One of the major findings reported here is that activities that require attention help the discrimination of ADHD and Non-ADHD subjects. Mixing in EEG data from periods of rest or during eyes closed leads to loss

of classification performance, to the point of approximating guessing when only resting EEG data is used.

Classification of ADHD and Non-ADHD Using AR Models and Machine Learning Algorithms

Juan Lopez Marcano

General Audience Abstract

As of 2016, diagnosis of ADHD in the US is controversial. Diagnosis of ADHD is based on subjective observations, and treatment is usually done through stimulants, which can have negative side-effects in the long term. Evidence shows that the probability of diagnosing a child with ADHD not only depends on the observations of parents, teachers, and behavioral scientists, but also on state-level special education policies. In light of these facts, unbiased, quantitative methods are needed for the diagnosis of ADHD. This problem has been tackled since the 1990s, and has resulted in methods that have not made it past the research stage and methods for which claimed performance could not be reproduced.

This work proposes a combination of machine learning algorithms and signal processing techniques applied to EEG data in order to classify subjects with and without ADHD with high accuracy and confidence. Signal processing techniques are used to extract autoregressive (AR) coefficients, which contain information about brain activities and are used as "features". Then, the features, extracted from datasets containing ADHD and Non-ADHD subjects, are used to create or train models that can classify subjects as either ADHD or Non-ADHD. Lastly, the models are tested using datasets that are different from the ones used in the previous stage, and performance is analyzed based on how many of the predicted labels (ADHD or Non-ADHD) match the expected labels.

Some of the major findings reported in this work include classification performance as high, if not higher, than those of the highest performing algorithms found in the literature. One of the major

findings reported here is that activities that require attention help the discrimination of ADHD and Non-ADHD subjects. Mixing in EEG data from periods of rest or during eyes closed leads to loss of classification performance, to the point of approximating guessing when only resting EEG data is used.

ACKNOWLEDGEMENTS

I would like to thank Dr. Beex for giving me the opportunity to work on such an interesting and demanding project. His constant guidance, sense of humor, encouragement, and discouragement over the past 10 months gave me the strength to successfully complete this thesis, publish multiple papers, and make relevant contributions to the field. Dr. Beex always encouraged me to go above and beyond, to question everything, and to go where no one has gone. I also want to thank Dr. Paul and Dr. Bailey, for having been my professors in the past, for diligently responding to my questions, and for being part of my thesis committee.

Needless to say, none of this would have been possible without the help of my family, my extended family, and most importantly, God. Thank you for your jokes, words, hugs, and in short, for being there one way or the other. You, along with my close and distant friends, shaped my life experiences, and they transcended to the work that is presented in this thesis.

I also want to thank family and friends who could not live long enough to hear the news of this thesis. You will always be remembered.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AIC	Akaike Information Criterion
ANT	Attention Network Task
AR	Autoregressive
ATT	Attention Deficit without Hyperactivity
AUC	Area Under the Curve
BCI	Brain-Computer-Interface
BMD	Bipolar-Mood Disorders
CDF	Cumulative Distribution Function
CT	Computed Tomography
DET	Detection Error Trade-off
EC	Eyes Closed
EEG	Electro Encephalogram
EP	Evoked Potentials
EER	Equal Error Rate
EM	Expectation Maximization
FN	False Negatives
FP	False Positives
FCM	Fuzzy C-means
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
KNN	K-Nearest Neighbor
LLR	Log-Likelihood Ratio
LSF	Line Spectral Frequency
MAP	Maximum a Posteriori
MEG	Magnetoencephalogram
MRI	Magnetic Resonance Imaging
MRS	Magnetic Resonance Spectroscopy
OCD	Obsessive-Compulsive Disorders
PET	Positron Emitted Tomography
RC	Reflection Coefficient
ROC	Receiver Operating Characteristics
SNR	Signal-To-Noise Ratio
SVM	Support Vector Machine
TBPR	Theta-to-Beta Power Ratio
TN	True Negatives
TP	True Positives
UBM	Universal Background Model

1. INTRODUCTION

In the US, ADHD is a condition that affects approximately 11% of children ages 4 to 17 [1]. Diagnosis of ADHD is done by using the Diagnostic and Statistical Manual of Mental Disorders (DSM), published by the American Psychiatric Association (APA) [2], which provides a list of symptoms that behavioral scientists use to determine whether or not a subject has a mental disorder.

1.1. History of ADHD

Although ADHD has been part of many people's lives, it was not acknowledged until the 1920's, and started to be treated in the late 1930's. Between 1918 and 1925, researchers noted that there was an unusual inattentive, impulsive, and hyperactive behavior in children who had had influenza [3-5]. In the 1930's, researchers agreed that some children had mild brain dysfunctions, which consisted of poor attention, hyperactivity, and behavioral dysfunctions. As early as 1937, children with these symptoms were treated with amphetamines, which is the main component found in Ritalin. Dr. Charles Bradley was the first physician to administer amphetamines to children with these symptoms, and he found that, depending on the dosage, academic achievement improved drastically [6].

Between the 1940s and the 1960s, there were great advances in ADHD research. In 1947, the terms minimal brain dysfunction (MBD) and "Strauss Syndrome" were formally coined to describe people with the aforementioned symptoms [7]. A large number of visual-motor and intelligence tests were created to differentiate people with MBD from people without MBD. By the early 1970s, it became evident that there were too many disorders to place under the MBD umbrella, so MBD was divided into 4 categories: learning disabilities, hyperkinetic disorders, conduct disorders, and attention disorders [5].

Since the late 1980s, there have been changes in the definition of ADHD. In 1987, the DSM-III-R changed attention disorders to attention-deficit disorders (ADD) with or without hyperactivity [8]. In 1994, with DSM-IV [9], ADD with or without hyperactivity was narrowed down to ADHD and three subtypes of ADHD defined: inattentive, hyperactive, and combined.

Finally, in 2013, DSM-V [2] placed ADHD under the umbrella of neurodevelopmental disorders. The symptoms, and the criteria for diagnosis of ADHD have changed from version to version.

1.2. Diagnosis of ADHD

Diagnosis of ADHD is done through subjective observations by teachers and/or parents and finally by behavioral scientists. When teachers or parents suspect that a child exhibits symptoms of ADHD, which comes about by observations, the child is taken to a behavioral scientist to investigate whether or not the child has the condition [10]. The behavioral scientist, in turn, observes the behavior of the child and compares the behavior of the child with the symptoms of ADHD described in the DSM. According to the DSM-V, someone with ADHD "often fails to give close attention to details or makes careless mistakes; often has difficulty sustaining attention to tasks; often does not seem to listen when spoken to directly; often fails to follow instructions carefully and completely; losing or forgetting important things; feeling restless, often fidgeting with hands or feet, or squirming; running or climbing excessively; often talks excessively; often blurts out answers before hearing the whole question; often has difficulty awaiting turn."

Unfortunately, these subjective observations have a large error associated with them. Snyder et al. [10] conducted a study with 159 participants, 101 males and 58 females, aged 6 to 18, where 61% of the subjects (97) were diagnosed ADHD (A) and 39% (62) were diagnosed Non-ADHD (NA). The subjects participated in clinical interviews and were diagnosed according to the DSM-IV and to Conners' Rating Scales-Revised (CRS-R), which is another manual used for the diagnosis of behavioral disorders. The study revealed that parents and teachers can predict ADHD with an accuracy of 47% to 58%. 47% was obtained when comparing the teachers' prediction and the diagnoses based on DSM-IV, and 58% was obtained when the teachers' prediction was compared to the diagnoses based on CRS-R. Likewise, parents' predictions matched 56% of CRS-R diagnosis and 55% of DSM-IV predictions. Therefore, teachers' and parents' can predict ADHD with an accuracy that is only slightly better than a guess.

ADHD is generally treated using behavioral interventions and/or pharmacological interventions [11]. Behavioral interventions consist of therapies that teach the person with ADHD to self-control, self-monitor, and self-evaluate, with the objective of improving the symptoms. According to the guidelines of the American Association of Pediatrics (AAP), behavioral interventions should be

used on children before considering medication, which consists of the use of amphetamine-based or methyliphenidate-based stimulants [12].

Unfortunately, stimulants used for ADHD treatment have side effects. In the short-term, these stimulants are known to cause weight loss, loss of appetite, and sleeping troubles. If taken over a long period of time, these medicines could cause high blood pressure, higher heart rate, and they could also increase the risk of acquiring heart arrhythmias [13]. Indeed, in 2006, the FDA's Drug Safety and Risk Management Committee decided to assign a "black box" warning, the strongest warning used by the FDA, to ADHD medications to indicate cardiovascular risks.

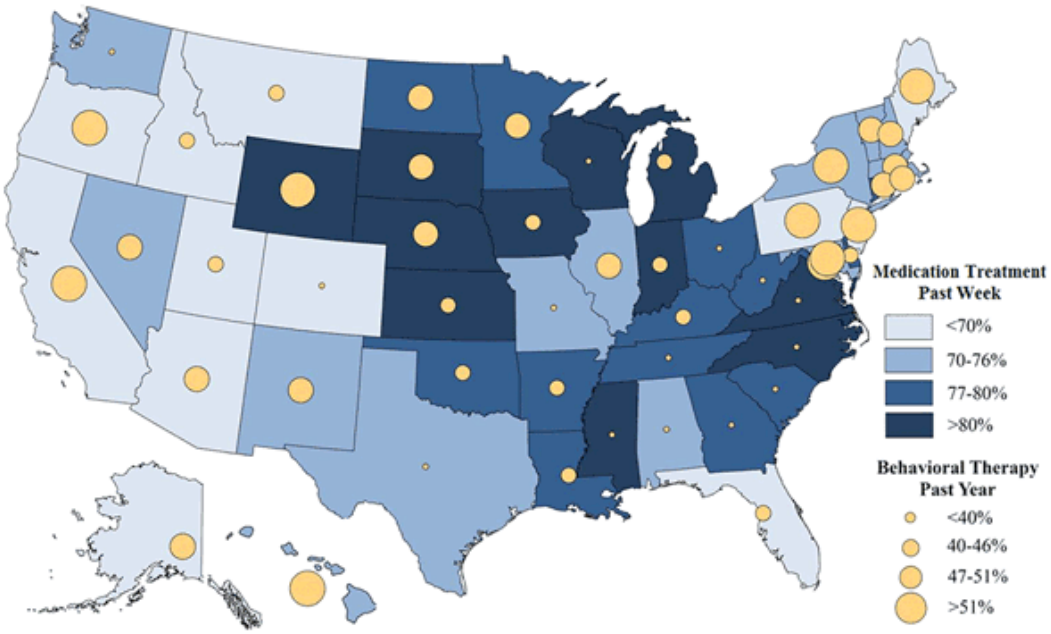

Fig. 1.1: Cool map showing the use of behavioral therapy and medications to treat ADHD in the US [1].

Figure 1.1 shows how ADHD was treated, between 2009 and 2010, in the US. On a different note, it is unknown what week the legends of the figure refer to. Note that there is a considerable number of states where over 80% of people diagnosed with ADHD use medication. In fact, in 2011, it was estimated that 6.1% of children aged 4 to 17 years, approximately 3.5 million, used ADHD medication [1]. Moreover, diagnosis of ADHD grows every year, and with it, consumption of stimulants.

One of the reasons diagnosis of ADHD is growing every year may be financial incentives. In some states, schools receive additional funding from the state depending on how many of their students have special needs. It is in those states where diagnosis of ADHD at an early age is highest and grows every year, whereas diagnosis of ADHD fluctuates slightly in the states where there is no additional funding that depends on the number of students with learning and other disabilities [14, 15].

Chances are, a large portion of the people who are diagnosed with ADHD, especially children, do not have ADHD. Elder conducted a study that analyzed the data from the Early Childhood Longitudinal Study-Kindergarten cohort (ECLS-K), which includes parent and teacher reports of ADHD symptoms, diagnoses, and stimulant-based treatments [16]. His research found that school cut-off dates and a child's date of birth greatly affect whether or not he or she will be diagnosed with ADHD. The study found that children born right before their state's kindergarten eligibility cut-off date are two times more likely to use stimulants, based on ADHD diagnosis, than those who are born right after the cut-off date and have to wait another year to start kindergarten. Elder estimates that 20% out of the 4.5 million children diagnosed with ADHD as of 2005 do not have ADHD. As a result of inadequate expectations and perceptions from their teachers, almost 900,000 children are taking medications they do not need and furthermore will result in negative long-term effects.

As argued, the cost of misdiagnosis is very high. Stimulants such as Ritalin and Adderall not only affect the behavior and experiences of its consumers, but also the lifetime of the consumers. Since the source error seems to be the subjective observations of parents and teachers, this work will explore quantitative/data-based ways to prescreen and/or diagnose ADHD.

In this study, information was extracted from electroencephalogram (EEG) data in order to discriminate between A and NA subjects. Multi-channel EEG data was used not only because of its availability, but also because it has multiple advantages over other methods. One of the advantages of having multi-channel EEG is facilitating minimization of the needed processing; processing data that does not contribute to higher discrimination performance may be detrimental. However, it may potentially provide robustness in the event that one of the channels fails.

1.3. EEG

EEG signals are low-power signals (usually below 0.6 mV) that are measured on the scalp of a human or non-human head and are currently being used for many applications. To record EEG data from a subject, electrodes are used and placed in different locations on the scalp of the subject. The recordings are, then, processed by a Data Acquisition Device, which typically performs sampling and low-pass filtering. As of today, EEG data is used in the health care industry, for the diagnosis of epilepsy, brain tumors, and stroke; in psychological and neurological research, for the study of the development and understanding of the human brain; and also, for brain-computer interfaces (BCIs), which use EEG data as inputs to a system [17].

Electrode placement follows a standardized configuration, which indicates how and where the electrodes should be placed. The configuration used, typically 10-20, indicates the distance between two adjacent electrodes. For 10-20, the distance is 10-20% of the front-back or right-left distance of the scalp [18]. The distances are measured from the Nasion (bridge of the nose) to the Inion (perceptual protuberance on the back of the head) and from preauricular point to preauricular point.

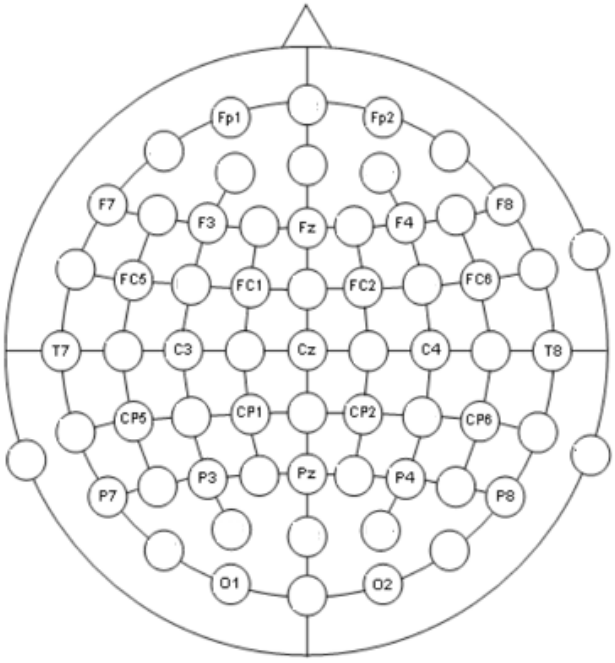

Fig. 1.2: 10-20 EEG electrode configuration.

Figure 1.2 shows the 10-20 configuration used in this work. Every channel starts with a F, T, P, C, or O, which stand for frontal, temporal, parietal, central, and occipital and represent the respective lobes of the brain. When electrode names have 2 letters, i.e. Fc, Cp, etc., this indicates that the electrode is between the two lobes denoted by the two letters. Finally, electrode names also contain a number, 1 through 8 in the example. Odd numbers indicate the left side of the scalp and even numbers the right side of the scalp [18].

EEG has been gaining popularity over the years because it has some advantages over other brain-scanning techniques such as magnetic resonance imaging (MRI), computed tomography (CT), positron emitted tomography (PET), magnetic resonance spectroscopy (MRS), and Magnetoencephalogram (MEG). These advantages include, but are not limited to:

- Cost efficiency: Hardware cost for the collection of EEG data is much lower than that of other techniques, especially because magnetically shielded rooms are not required.

- Resolution: EEG data can provide data at the millisecond level, which is impossible with MRI, CT, and other techniques [19].

- Ubiquity: Although collection of EEG data requires placement of electrodes on the scalp, it is more ubiquitous than other techniques, such as MRI, MRS, and PET, which can even elicit claustrophobia. Further, EEG is silent [20].

- Low radiation: Usage of EEG does not require exposure to intense magnetic fields (over 1 Tesla), which is the case for MRS, MRI, and MEG [21].

All of these advantages make EEG a great option for neuroscience research. However, EEG could be inconvenient for the following reasons:

- Low SNR: Signal-to-noise ratio is low in EEG. This can be mitigated by filtering, either through hardware or software.

- Preparation: EEG electrodes have to placed correctly on the scalp by using gels, saline solutions, or other methods so that the electrodes are in contact with the scalp for the duration of the test. This makes the preparation time for EEG data collection longer than for the other methods.

- Cryptic: EEG data is not in the form of images, which makes it difficult to observe the interaction between different brain regions or the activation of neurotransmitters, which could be done with techniques that use resonance [19].

The latter disadvantage may not be a disadvantage depending on the application. EEG data can be considered “raw” brainwave data, and useful information can be obtained about events in the brain if some processing is done and even if no processing is done.

In the health sciences, derivatives of EEG data are used. These derivatives are Evoked Potentials (EPs) and Event-related Potentials (ERPs). The first consists of averaging the EEG data over a time interval as a stimulus (i.e. auditory, visual) is presented to the subject [22]. The latter consists of averaging the EEG over a period of time when the subject is performing a motor or cognitive task repeatedly. These markers are widely used in cognitive psychology and neuroscience research [23].

By filtering EEG signals, the behavior of the brain in different frequency bands, colloquially known as brainwaves, can be observed. In the literature, 5 different frequency bands are defined: Delta (0.1 to 4 Hz), Theta (4 to 7 Hz), Alpha (8 to 13 Hz), Beta (13 to 30 Hz), and Gamma (over 31 Hz) [24]. These band designations are fairly consistent but, depending on the source, there is some variation. The power in frequency bands (computed using the FFT) has been studied for the classification of A and NA subjects.

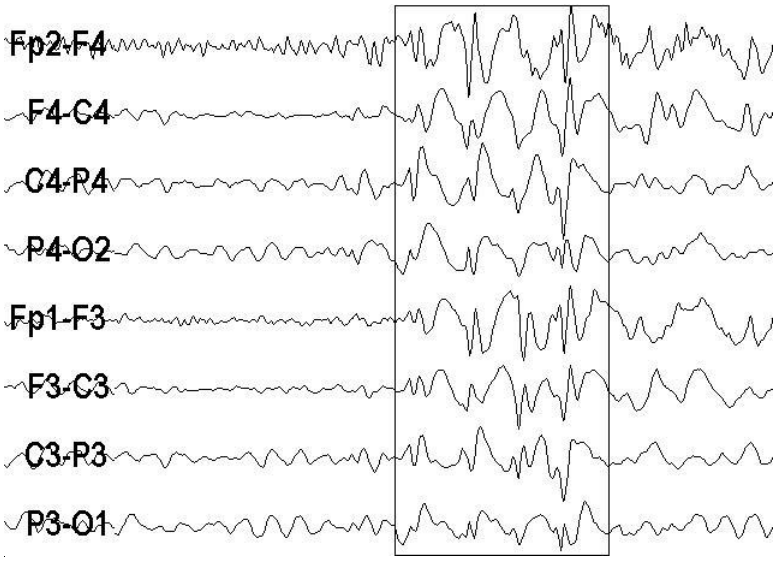

Fig. 1.3: EEG Signals of an epileptic patient during seizure. Image taken from [25].

For epilepsy, EEG data can be used for diagnosis even without processing. Epilepsy is characterized by "abnormal and/or excessive synchronization of the brain cells". This synchronization can be observed with little effort in EEG data, as shown in Fig. 1.3. In the figure, it can easily be seen that there are groups of EEG channels (boxed) that are synchronized [26].

For BCIs and classification problems, more processing is needed because the objective is to automatically and accurately detect patterns in the brainwaves. Machine learning algorithms are generally used in order to create models for the classes (i.e. activities, thoughts, conditions, etc.) that will be detected by the algorithm. The work done in this thesis falls in this category.

This is not the first time signal processing and machine learning algorithms are used for the classification of A and NA. This will be covered in more detail in Chapter 2, but this problem has been approached since 1992, and high performance has been reported. To the best of our knowledge, the closest competitors to the algorithms presented in this thesis are [27] and [28]. The former reported performance in terms of AUC, with average AUC of 0.97 (See Section 5.5 for a definition of AUC) and the latter reported performance in terms of sensitivity (A subjects classified correctly), and accomplished 95.6% of sensitivity, missing 4.4% of the A subjects on average. In this thesis, the method that maximized performance achieved average AUC of 0.99 and missed 3.57% of the A subjects on average.

1.4. Limitations

As will become evident in the next sections, the work presented in this thesis is preliminary and has its limitations. To begin with, this work is a pilot study because the number of subjects whose EEG data was available to us is 8 (4 A and 4 NA). It is understandable if the reader questions whether or not the results presented in this thesis will generalize to larger datasets. It is worth noting, however, that 250 test vectors were extracted from each subject, and training and testing of the classification models developed in this work were performed for 30 unique selections of training and testing subjects, since the data that was used to create and test the models was switched in order to provide average, best, and worst performance values for the classification algorithm.

Besides availability of data, a major limiting factor is the definition of ADHD itself. ADHD is defined by a set of symptoms, which overlap with those of 15 other conditions, such as Bipolar Mood Disorders (BMD), Sleep Disorders, and Obsessive-Compulsive Disorder (OCD) [29, 30].

Therefore, it is possible that a child evaluated by a clinician may be diagnosed with ADHD when the accurate diagnosis is BMD. In fact, it has been reported that a child may exhibit ADHD symptoms as a result of dietary habits, exposure to toxins such as lead, and deficiencies in vitamins, but less than 26% of clinicians in the US use laboratory tests to determine if the observed ADHD symptoms are caused by neurodevelopmental disorders or by organic reactions [29]. Consequently, it is difficult to say if children diagnosed with ADHD actually have mental disorders; and those who do have mental disorders may not necessarily have ADHD.

The way ADHD is diagnosed in the US has changed over the years and it may continue to change. With every new version of the DSM, ADHD symptoms have changed and/or the number of minimum symptoms required in order to receive ADHD diagnosis has changed, but these symptoms are still subjectively observed. Moreover, reports from parents or teachers play a role in the diagnosis of ADHD, and it is known that they can detect it with 47-58% of accuracy [10].

Another concern is whether or not diagnosis of ADHD should be tailored to different age groups. As of now, the DSM-V has different requirements for adolescents and adults over 17 years and children up to age 16. Should there be more groups? Research shows that the age group that a child is compared to has an impact on whether or not the child receives an ADHD diagnosis or not [16], and age group alone is a factor that seems to have caused an estimate of 20% incorrect diagnoses. This occurs, again, because behavioral patterns are observed subjectively. In this thesis, the eight subjects were ages 6 to 8. However, the algorithms presented here do not take into consideration age because it was unknown to us which subjects were 6, 7, or 8 years old.

If the pattern detection work was performed by an objective unit that evaluates objective metrics or features, we hypothesize that these errors would be mitigated. Be mindful, however, that the goal of this work is not to provide a solution to reduce that very high 20% error. The goal of this work is to study the feasibility of automatic classification of A and NA subjects, based on autoregressive coefficient features extracted from multi-channel EEG data, which are then used to create classification models.

The fact that EEG data is used brings up other concerns because EEG is task-dependent. Assuming there are particular mental tasks known to be useful for the classification of A and NA, how would a classification of A and NA be affected if a subject is not performing the activity that he or she is instructed to perform? The answer is that performance most likely decreases. EEG

data is so sensitive to tasks that BCIs can detect different mental tasks with over 90% of accuracy depending on the classification algorithms used [31-33]. Therefore, an A subject may be labeled as NA if inadequate data is used for diagnosis.

Something else that should be considered is severity levels of ADHD. The DSM-V defines three severity levels: mild, moderate, and severe [2]. The level of severity depends on how much of the person's social and occupational functioning is affected by ADHD, and the level of severity is assigned subjectively as well. This thesis performs binary classification (either A or NA), and does not provide information on severity levels since that information was not available to us. Moreover, we believe that ADHD diagnoses should come in a continuum of values and should indicate how confident the diagnosis was. Alas, current diagnosis methods do not provide any of these metrics.

Last but not least, gender may be something that should be taken into consideration. It has been reported that ADHD is more common in boys than girls, and ADHD is manifested differently in boys and girls [34]. This work does not take gender into consideration as the gender of the subjects was not provided to us.

All of these reasons make it difficult to obtain the "golden standard" ADHD subject, which is another limitation to the work presented here. The quality of a classification model depends on the quality of the data that was used to create the model. Since it is difficult to find the "golden standard", it is difficult to create ideal classification models.

In fact, it is questionable whether or not any data is labeled correctly. This is an issue that is faced in Section 4.6.2 of this thesis, where the label of one NA subject was questioned and later flipped to A. Maybe this subject was indeed NA but was not performing the activity that he or she was instructed to perform; or maybe he or she was slightly more active or inattentive than the other NAs; or maybe the EEG data of this subject appeared to be similar to that of A subjects as an organic response to dietary habits; or maybe he or she was actually mislabeled.

In summary, studies that concern the classification of A and NA face many challenges. There are many sources of error, to the point that the labels used in a study may not be trustworthy. Nevertheless, if the A and NA labels used are correct, frameworks for the classification of A and NA can be created, and this work argues that if optimization is done at every stage of the

framework, classification can be done with high levels of accuracy for the best and worst case scenarios.

1.5. Outline

The rest of the thesis is organized as follows. Chapter 2 contains a review of the research that has been done in order to classify A and NA subjects. Special attention is given to controversial methods that were investigated between 1992 and the present time. In Chapter 3 the mathematical techniques used in feature extraction are reviewed. This chapter also contains a review of how EEG channel selection was done in this thesis. In Chapter 4 the use of the K-nearest Neighbor (KNN) algorithm for classifying A and NA subjects is summarized. In Chapter 5 the mathematical background for, and the results obtained from, a Gaussian-Mixture-Model-based (GMM) Universal Background Models (UBM) for the classification of A and NA are provided. In Chapter 6 is shown how KNN and GMM-UBM can be modified to account for the uncertainty in diagnoses or labeling used for training, which in this document are referred to as soft labels. An iterative approach to minimizing the number of channels for KNN and GMM-UBM is described in Chapter 7. Lastly, in Chapter 8 the conclusions and suggestions for future work are provided.

2. LITERATURE REVIEW

Abnormal EEG patterns have been observed in ADHD subjects over the last 70 years. To the best of our knowledge, in 1938, Dr. Bradley presented evidence showing that there were EEG abnormalities in the children he administered amphetamines to, making him the first to report this observation [6]. Numerous studies between the 1950s and 1960s also noted abnormal EEG signals for A subjects.

Quantitative classification of A and NA subjects dates back to the 1980s. Some of the approaches that researchers proposed produced results that had large errors or could not be reproduced. In the last 5 to 8 years, on the other hand, the methods that have been proposed have much higher accuracy. In this chapter will be discussed how EEG has been used over the years in order to classify NA and A.

2.1. Spectral Analysis

Since 1992, advances have been made towards quantitatively finding differences between A subjects and NA subjects. In 1992, Mann et al. [35] studied the power in frequency bands of 25 A subjects and 27 NA subjects while they were in baseline activity, reading, and drawing. The number of windows or epochs and the duration was not reported, but there were between 90 and 100 s of EEG recordings for each activity. The study found that the power in the theta band was higher for A subjects than for NA subjects, and the power in the beta band was much lower for the A subjects than for the NA subjects in channels F3 and F4. These features were used in a discriminant function whose type was not disclosed, and reported A subjects were classified correctly 80% of the times and NA subjects were classified correctly 74% of the times.

In 1996, a study used a 19-channel configuration to collect EEG signals from 310 control subjects and 407 ADHD/ATT (other hyperactivity disorders) subjects aged 6 to 17 [36]. EEG signals were recorded during eyes-closed activity, and 24-48 windows of 2.5 s per subject were used. The study evaluated the use of mean coherence, mean frequency, and absolute power in frequency bands in order to create a discriminant function to classify control subjects and ADHD/ATT subjects [36]. Although it was not specified what kind of decision function was based on these features (i.e. decision tree, linear discriminant, quadratic discriminant, etc.), the approach

achieved 93.1% of sensitivity (correct classification of ADHD/ATT subjects) and 94.8% of specificity (correct classification of control subjects).

2.2. Theta-to-Beta Power Ratio (TBPR)

In 1999, a study reported that the θ/β power ratio (TBPR) of A subjects tends to be higher than that of NA subjects [37]. The hypothesis was that, since theta brainwaves (4-7 Hz) are associated with hyperactivity and beta brainwaves (13-30 Hz) are associated with attention, the ratio of the power in those bands would be larger for A than for NA subjects. To test this hypothesis, EEG signals of 482 subjects 6-17 years old were recorded. 17.63% of the subjects (85) were NA and 82.37% of the subjects (397) were A. The data was recorded from a single EEG channel, Cz, while the subjects were in resting eyes open, eyes closed, reading, and performing visual, and motor activity. At least 15 2-s windows were collected from each subject during each task. In the study, the TBPR was obtained by computing the PSD estimates from the FFT for the theta and beta bands. For classification, the TBPR of NA subjects was averaged, and power ratios that were more than 1.5 standard deviations above the average TBPR of NA (the threshold) were classified as associated with A subjects, whereas those that fell below the threshold were classified as NA. As performance metrics, sensitivity (A subjects classified correctly) and specificity (NA subjects classified correctly) were used. This simple decision rule was reported to yield "86% of sensitivity… 98% of specificity … and 99% overall predictive power", as confusing and strange as this may sound.

Some of the authors of the latter study replicated their approach in 2001 [38]. The same methods were used, but this time, a population of 129 subjects, aged 6 to 20, was used. In this set, there were 96 A and 33 NA. Sensitivity was reported to be 90% and specificity 94%. There were other studies, but sensitivity and specificity could not be analyzed because their datasets only had A subjects.

The methods developed by [37] have been replicated in numerous studies. Snyder et al. used TBPR in their study [10], which had 159 participants, 97 A and 62 NA. EEG data was recorded from 19 channels in a configuration that follows the international 10-20 system while the subjects were resting with eyes closed, eyes open, reading, and listening. TBPRs were computed for channels Fp1, Fp2, F3, F4, F7, and F8 for at least 15 windows of 4 seconds for each activity. Sensitivity, specificity, and overall accuracy were found to be 87%, 94%, and 89% respectively,

which is in line with reported results [37, 38]. Other studies, between 2004 and 2008, used TBPR for a similar number of windows, similar activities, but different number of subjects, and reported results between 87% and 96% accuracy [39-41].

Another study [27] used power in frequency bands along with semi-supervised learning in order to classify A and NA subjects. EEG data was recorded from 10 subjects, 7 A and 3 NA, while they were performing an activity that requires attention that lasted approximately 2 minutes. In this study, the power and power ratios in the α, β, ϴ, and γ frequency bands were computed over windows of 1 s from channels F3, F7, F8, Fz, Fp1, Fp2, and Cz out of a 10-20 configuration. The mutual information criterion was used to choose the least redundant features for training of a Gaussian support vector machine (SVM). The accuracy of classification, measured in AUC, was 0.92 for TBPR and 0.97 for theta; miss rates were not reported.

Although the problem of classifying A and NA subjects seemed to be solved, recent studies failed to replicate the results [42, 43]. In 2014, a study involving 62 A and 55 NA subjects reported accuracy rates between 49.2% and 54.8%. In this study, EEG data was recorded for 3 minutes of eyes closed activity. Although 128 EEG channels were available, only channels Cz, Fz and Pz were used. The TBPR was computed for 2 s windows with overlaps of 1 s, which results in a large number of windows to test with for every subject. For classification, logistic regression was used, and the AUC were found to be between 0.492 and 0.548. Similarly, another study [44], which involved 54 A and 51 NA subjects during eyes closed and eyes open activity, reported 40% to 53% overall accuracy for TBPR using stepwise discriminant analyses and other discriminant functions that were not disclosed. In this study, there was a 10-20 configuration, and TBPRs were computed from Cz. In total, there were approximately 20 2-second windows for each activity.

Although the validity of TBPR became questionable, the studies that question TBPR did not exactly use the method described earlier [37]. For instance [42] used logistic regression rather than a threshold based on standard deviations; likewise, [44] used discriminant functions, which were not thoroughly explained, that may be different from the threshold concept [37]. These factors may have had an impact on the performance of TBPR. Moreover, the data acquisition devices used vary from study to study.

Since it is difficult to say whether or not TBPR is the solution to the classification of A and NA, other methods must be explored. Unfortunately, most of the studies have focused on reproducing

the results found for the TBPR or using the TBPR. However, there are a few studies that have used signal processing techniques and machine learning to approach the classification of A and NA.

2.3. Other Approaches for the Classification of A and NA Subjects

The effectiveness of event-related potentials (ERPs) was studied as well [45]. In that study, 74 A and 74 NA subjects performed a visual two-stimulus GO/NOGO task while their EEG data was recorded, which lasted approximately 22 minutes. Independent component analysis (ICA) was performed on the ERPs, and these features were used to train a SVM classifier, which achieved 92% accuracy of classification (90% sensitivity and 94% specificity).

Another method that has been explored is feed-forward neural networks [28]. The study had 54 subjects, 47 A and 7 NA, whose EEG signals were recorded during eyes closed activity for 3 minutes, which resulted in 14,000 samples for each of the 19 channels used in this study, for each subject. With the wavelet algorithm (no details specified), the EEG data was decomposed into the 5 frequency bands between 0 and 60 Hz (alpha, beta, delta, theta, and gamma) plus the original signal. This was for every channel so there were 6x19 = 114 features at the input layer of the neural network. The hidden layer combined the features non-linearly, and returned a value of 1 or 0 to indicate A or NA respectively. By using this approach, a sensitivity of 95.6% was achieved.

There are other studies, but they focus more on replication than on creation. Therefore, other approaches that have been successful at finding EEG patterns should be considered for the classification of NA and A subjects. The detection of stroke, epilepsy, and the classification of mental tasks for BCIs have inspired many successful algorithms, which should also be considered for the problem at hand.

2.4. Classification of Other EEG Patterns

Autoregressive (AR) coefficients are good candidates to use as features for the classification of A and NA. Autoregressive coefficients have been largely used as features in BCIs, yielding highly accurate results for the classification of mental tasks (reading, performing arithmetic operations, etc.) using short windows of time [31-33]. If the functioning of a NA brain is modeled as an activity and the functioning of an A brain is modeled as another activity, then the results that were found in the latter studies [31-33] should extrapolate to NA and A.

A study explored the use of AR coefficients for the classification of ADHD (A) subjects and bipolar mood disorder (BMD) subjects [46]. In the study, EEG data was recorded from 21 A and 22 BMD subjects while they were in eyes closed and eyes open activities, 3 minutes each. A 10-20 configuration with 22 channels was used, and the AR coefficients, of unspecified order, were extracted from 1-second intervals from each channel. Multiple classifiers were trained and the overall accuracy was slightly over 70%.

As far as algorithms go, Gaussian Mixture Models (GMM) have been used for classification, but not specifically for the classification of A and NA subjects. GMMs have been used for neonatal seizure detection [47]. The study recorded EEG data from 17 subjects who were between 39 and 42 weeks old. The data was recorded using eight combinations of two channels for approximately 15 hours per subject. Approximately 691 seizures were observed during that time for all the subjects. As features, the power in multiple frequency bands and peak frequencies in the spectrum were used, and the performance, measured in AUC was 0.9556.

3. FEATURE EXTRACTION

Feature extraction is an essential component in machine learning, especially when dealing with large datasets. There are two reasons for performing feature extraction. First, feature extraction is done to represent a vector of samples of arbitrary size as a vector of samples of lower size, in order to reduce the number of computations needed in order to classify the vector. Second, depending on the data, raw data may not be enough to detect patterns/perform classification. For stroke detection, it is clear when there is a stroke and where there is not because stroke causes EEG data to spike. Thus, feature extraction may not be needed. Nevertheless, when the changes in the data are very subtle or not visible, features must be extracted to maximize the difference between the different classes that need to be detected.

In this chapter, the mathematical background and algorithms used in order to compute the features used throughout the thesis are presented. This chapter focuses on AR modeling and the method used in order to select the EEG channels used throughout this thesis.

3.1. AR Modeling

The objective of linear prediction is to predict the current value of a signal based on its previous values. Linear prediction theory states that, for an adequate value of *p*, the current value of a discrete stochastic process $x[n]$ can be predicted based on its *p* previous values incurring a prediction error $e[n]$, which is a white process.

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + e[n] \tag{3.1}$$

where the coefficients a_k are AR coefficients. Then, defining the predicted estimates as

$$\hat{x}[n] = -\sum_{k=1}^{p} a_k x[n-k] \tag{3.2}$$

yields the prediction error

$$e[n] = x[n] - \hat{x}[n] \tag{3.3}$$

From a deterministic point of view, the discrete stochastic process $x[n]$ in (3.1) has a Z-transform $X(z)$, which can be expressed as a function of $e[n]$, which has a Z-transform $E(z)$.

$$X(z) = \frac{E(z)}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.4}$$

From (3.4), $E(z)$ can be modeled as the output of a prediction filter $A_p(z)$ when driven by $X(z)$. This process is summarized in Fig. 3.1.

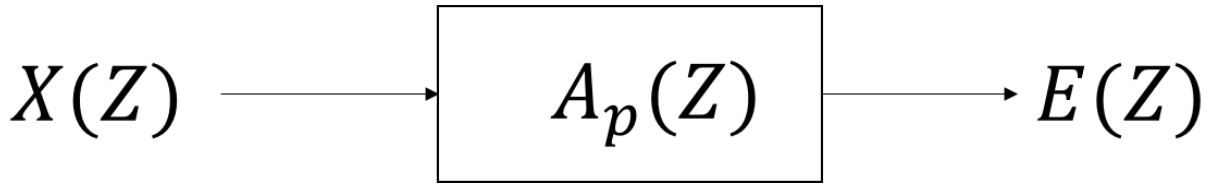

Fig. 3.1: Linear prediction model.

$A_p(z)$ in Fig. 3.1 is a prediction filter of order p that can be expressed as

$$A_p(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{3.5}$$

AR coefficients are computed to "minimize the error" associated with prediction. The error criterion that is minimized depends on the algorithm. This chapter will limit the development to only that for Burg method, since that is the one that was used. This choice was made to minimize a particular error criterion and also guarantee model stability, as will be shown in the next section.

3.2. Burg Method

The Burg method finds the AR coefficients by minimizing the sum of the forward and backward prediction errors, $f_p[n]$ and $b_p[n]$, in the least-squares sense over a time interval of length L samples, in an order-recursive fashion

$$E_i = \sum_{m=i}^{L-1} \left(f_i^2[m] + b_i^2[m] \right) \tag{3.6}$$

where $f_i[n]$ and $b_i[n]$ are expressed as

$$f_i[n] = x[n] + a_{i1}x[n-1] + a_{i2}x[n-2] + ... + a_{ii}x[n-i] \tag{3.7}$$

$$b_i[n] = x[n-i] + a_{i1}x[n-i+1] + a_{i2}x[n-i+2] + ... + a_{ii}x[n] \tag{3.8}$$

where each a_{ik} is an autoregressive coefficient of a model of order *i*.

For the order-recursive development, in the Burg method, $f_i[n]$ and $b_i[n]$ are rewritten as

$$f_i[n] = f_{i-1}[n] - \gamma_i b_{i-1}[n-1] \tag{3.9}$$

$$b_i[n] = b_{i-1}[n-1] - \gamma_i f_i[n-1] \tag{3.10}$$

where γ_i are the reflection coefficients and i=1,2,…, p. Reflection coefficients are a different representation of the AR coefficients. They carry the same information, but their values have different distributions. When using Burg's method, the reflection coefficients are constrained to the range [-1,1].

By substituting (3.10) and (3.9) into (3.6), (3.11) is obtained

$$E_i = \sum_{m=i}^{L-1}\left(\left(f_{i-1}[m] - \gamma_i b_{i-1}[m-1]\right)\right)^2 + \sum_{m=i}^{L-1}\left(b_{i-1}[m-1] - \gamma_i f_i[m-1]\right)^2 \tag{3.11}$$

by taking the derivative of (3.11) and setting it to 0, the maximum is found

$$\frac{\partial E_i}{\partial a_i} = -2\sum_{m=i}^{L-1}\left(\left(f_{i-1}[m] - \gamma_i b_{i-1}[m-1]\right)\right) - 2\sum_{m=i}^{L-1}\left(b_{i-1}[m-1] - \gamma_i f_i[m-1]\right) = 0 \tag{3.12}$$

and solving for γ_i results in

$$\gamma_i = \frac{2\sum_{m=i}^{L-1}\left(f_{i-1}[m] b_{i-1}[m-1]\right)}{\sum_{m=i}^{L-1}\left(f_{i-1}[m]^2 + b_{i-1}[m-1]^2\right)} \tag{3.13}$$

which can be used to compute the values of γ_i. (3.13) is used recursively along with (3.14) in order to find the AR coefficients

$$\mathbf{a}_p = \mathbf{a}_{p-1} - \gamma_p \mathbf{a}_{p-1}^R \tag{3.14}$$

where $\mathbf{a}_p$ is the vector of AR coefficients of order p and $\mathbf{a}_{p-1}^R$ is the time-reversed vector of AR coefficients of order p-1 . (3.14) can be written in matrix form as

$$\begin{bmatrix} 1 \\ a_{p1} \\ a_{p2} \\ \vdots \\ a_{p,p-1} \\ a_{pp} \end{bmatrix} = \begin{bmatrix} 1 \\ a_{p1} \\ a_{p2} \\ \vdots \\ a_{p-1,p-1} \\ 0 \end{bmatrix} - \gamma_p \begin{bmatrix} 0 \\ a_{p-1,p-1} \\ \vdots \\ a_{p2} \\ a_{p1} \\ 1 \end{bmatrix} \tag{3.15}$$

which is known as the Levinson-Durbin recursion [48].

In short, the Burg algorithm can be summarized by the following steps:

0. Initialize the parameters

$f_p[n] = b_p[n] = x[n]$, $A_0(z) = 1$, and $E_0 = \frac{1}{L}\sum_{m=0}^{L-1} x[m]^2$

1. At stage $p-1$, the following information is available

$f_{p-1}[n]$, $b_{p-1}[n]$, and $A_{p-1}(z)$

2. Compute γ_p using (3.13)
3. Compute $A_p(z)$ using (3.15)
4. Compute $f_p[n]$ and $b_p[n]$
5. Compute the error using $E_p = \left(1-\gamma_p^2\right)E_{p-1}$
6. Go to stage p

The Levinson-Durbin recursion could also be executed backwards: If quantities γ_p, E_p, $f_p[n]$, $b_p[n]$, and $A_p(z)$ are known, the steps could be done in reverse to find $f_{p-1}[n]$, $b_{p-1}[n]$, $A_{p-1}(z)$, E_{p-1}, and γ_{p-1}. However, trouble arises when any $|\gamma_i| = 1$ because $E_{p-1} = \frac{E_p}{\left(1-\gamma_p^2\right)}$. Fortunately, $|\gamma_i| = 1$ is unlikely to happen because γ_i is a partial correlation coefficient.

There are other algorithms to find AR models, but they have disadvantages. The autocorrelation method minimizes only the forward prediction error and zero-pads the ends of $x[n]$, which introduces a bias that increases errors on the one hand, but guarantees stability; the covariance method has stability issues because the reflection coefficients are not constrained to [-1, 1]. Levinson-Durbin recursion solves Yule-Walker equations without performing any kind of minimizations. Although the coefficients may not vary too much in practice, these are some of the reasons the Burg algorithm is considered to be more robust than other methods.

Hence, the Burg method produces 2 vectors that contain the same information: a vector of AR coefficients, and a vector of reflection coefficients. Throughout this work, AR coefficients are constantly used. Reflection coefficients and Line Spectral Frequencies (LSF) were used for some experiments in order to investigate which one of these quantities maximized the accuracy of classification.

3.3. LSF

LSF are a different representation of AR coefficients, just like reflection coefficients. LSF are computed from the AR coefficients. This is done by expressing $A_p(z)$ as the sum of two polynomials, $P(z)$ and $Q(z)$, both of order $p+1$,

$$A_p(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} = \frac{P(z) + Q(z)}{2} \tag{3.17}$$

where $P(z)$ and $Q(z)$ can be expressed as

$$P(z) = A_p(z) + z^{-(p+1)} A_p\left(z^{-1}\right) \tag{3.18}$$

$$Q(z) = A_p(z) - z^{-(p+1)} A_p\left(z^{-1}\right) \tag{3.19}$$

The LSF are the phases of the roots of both polynomials, which are defined as

$$\mathbf{w}_{Pp} = \left[w_{P1}, w_{P2}, ..., w_{Pp} \right] \tag{3.20}$$

$$\mathbf{w}_{Qp} = \left[w_{Q1}, w_{Q2}, ..., w_{Qp} \right] \tag{3.21}$$

and are interspersed.

$$\mathbf{L}_p = \left[w_{P1,} w_{Q1}, w_{P2,} w_{Q2}, ..., w_{P\frac{p}{2}}, w_{Q\frac{p}{2}} \right] \tag{3.22}$$

3.4. Akaike Information Criterion (AIC)

Although AR coefficients and its different variations are said to model the behavior of a signal, there is a parameter that has to be tuned: the order of the model. If the order of a model is too low, it will not be able to model the data it was created with (underfit). On the other hand, if the order is too high, it will accurately model the data it was created with, but it will not be able to generalize to model data in future time instants (overfit). Therefore, a reasonable order must be chosen in order to accurately model the existing data and predict data at future time instants with a reasonable error.

A way to compute goodness of fit of a model is by using the Akaike Information Criterion (AIC) [49]. There are other methods, but a study that investigated order selection methods for EEG signals deemed most, if not all, of the methods to be useless, but found AIC to be the only method that tended to not underestimate the order of the AR model [50]. The AIC is computed as follows:

$$AIC(p) = L \ln\left(\sigma^2\right) + 2p \tag{3.23}$$

where L is the length of the window used, σ^2 is the prediction error variance of the model, and p is the order of the model. The order p of a model is taken to be that which "best" fits, i.e. the value of p that minimizes *AIC*. In order to choose the order of a model, the *AIC* was computed on 100 intervals of 51 samples for AR models of orders 1 through 15. The AR coefficients, of orders 1 through 15, were computed on the 100 intervals to obtain σ^2 for every model for every interval.

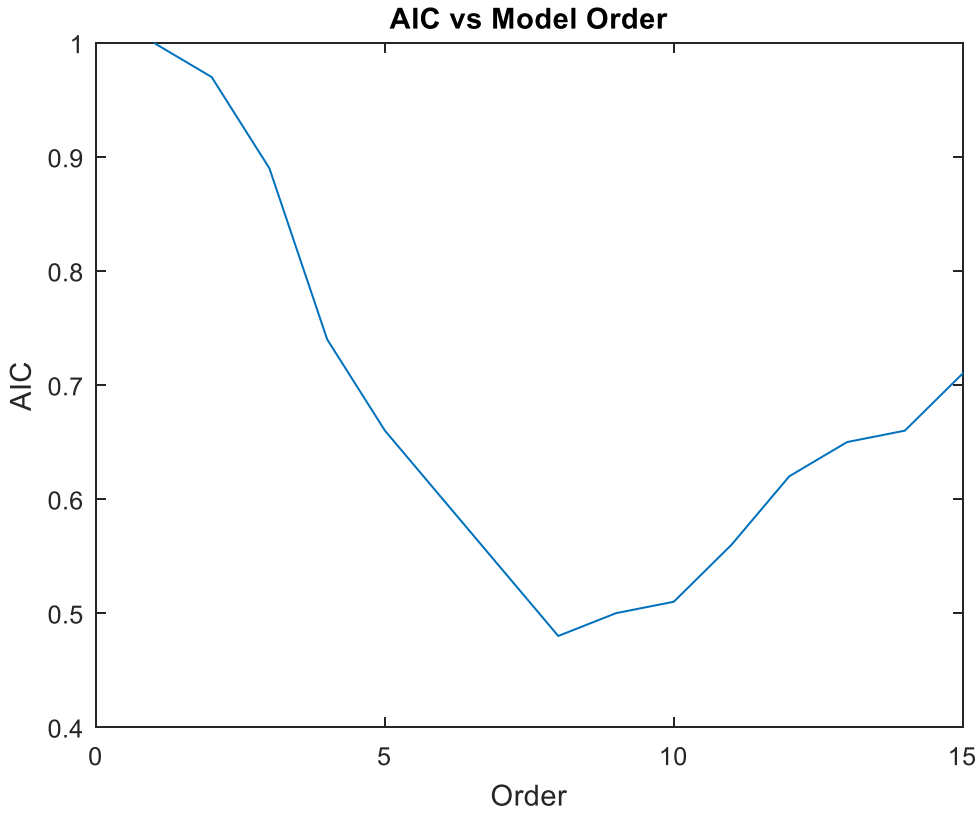

Fig. 3.2: AIC vs model order.

Figure 3.2 shows the results of the experiment. The plot shows the normalized *AICs*, which were obtained by averaging all 100 *AICs* for every order *p*. The graph shows that *AIC* is minimum when the order is 8. Note that 7 would also be a good choice, but not as good as 8 or 9. Since AIC is known to slightly overestimate the order of a model [51], it was set to 7, instead of 8 or 9, for the rest of the thesis.

3.5. Channel Choice

To reduce the number of channels to be used for the analyses described herein, the aim was to determine five channels that probably will provide good discrimination. Previous research indicates that resting state eyes-open and eyes-closed theta/beta power ratios (TBPR) tend to be higher for A subjects than for NA subjects [35, 37]. The preliminary step of channel reduction is therefore executed based on TBPR; however TBPRs were evaluated here during ANT activity for all recorded channels, for all subjects, i.e. not during resting state. TBPRs were computed using the FFT over the entire duration of the ANT task to compute the power spectral densities.

$$\tau_{csk} = TBPR(c,s,k) = \frac{PSD(\theta)}{PSD(\beta)} \tag{3.24}$$

where τ_{csk} indicates the TBPR of class c, subject s, for channel k. For $c = 0, 1$, $s = 0, 1$, and $k = 0, 1, \ldots, 23$. $c = 0$ indicates A and $c = 1$ indicates NA. The next step was computation of all cross ratios, defined as the ratio of TBPR-A over TBPR-NA.

$$\Omega_{slk} = \frac{\tau_{0sk}}{\tau_{1lk}} \tag{3.25}$$

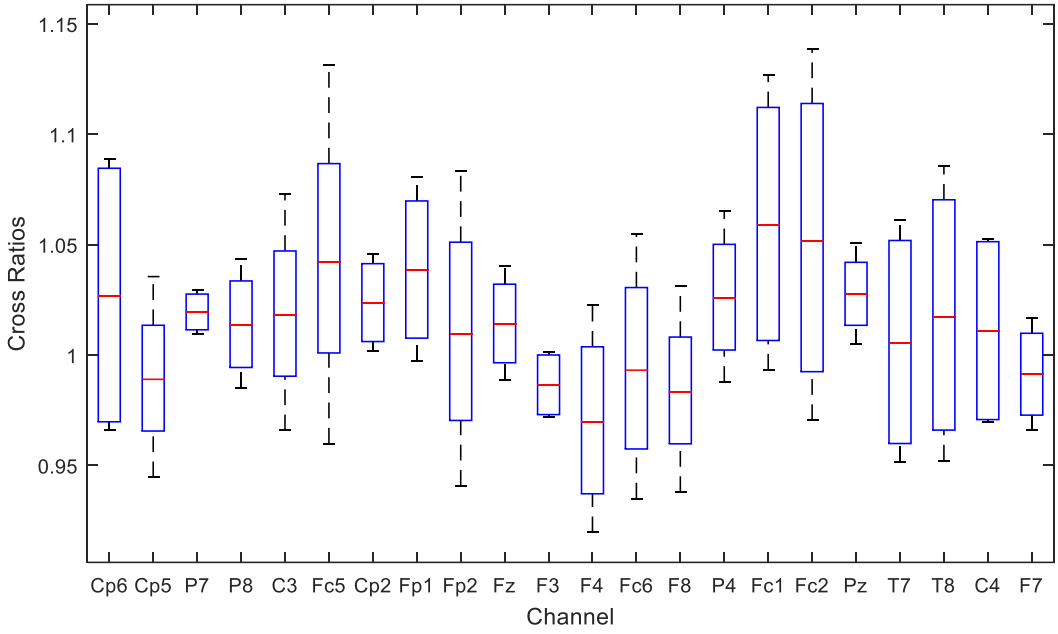

Fig. 3.3: Cross ratios for all channels.

Figure 3.3 reflects the distribution of cross-ratios for each of the EEG channels, in the form of boxplots. The red lines in the middle of boxes represent the means, while the top and bottom sides of the boxes represent the 75th and 25th percentiles respectively, with the upper and lower horizontal lines representing the maximum and minimum values respectively. The cross ratios fall between 0.9 and 1.15. Note that these are not centered about 1, i.e. higher for A than for NA, so there is some truth to [35, 37]. Based on this preliminary analysis step, the channels chosen to proceed with are Fc2, Fc1, Fc5, Cp6, and C3. This choice does not necessarily mean that these five channels produce the very best possible discrimination; after all, the performance of various

methods is yet to be analyzed, and the results of such analysis may indicate that optimization of the channel choice needs refinement when targeting a specific application.

Fc2, Fc1, Fc5, Cp6, and C3 are the 5 channels used for the experiments described in Chapters 4 through 7. Since the order of the AR models was set to 7, the AR(7) were extracted from each of the 5 channels, for every time interval/window of the ANT activity, and/or the activity under study. This resulted in 35-D (5x7) feature vectors. Note that there are 8 AR coefficients for an AR(7), but the first coefficient is always normalized to 1, regardless of the data. Therefore, the first parameter was not included in the feature vector.

3.6. Dataset

The data used in this study was made available to us by our collaborator from the Psychology Department at Virginia Tech, Dr. Martha Ann Bell. The dataset consisted of 8 subjects: 4 A and 4 NA children between the ages of 6 and 8 years, who visited the research lab as part of an ongoing longitudinal study focused on frontal lobe development from infancy through childhood. Information regarding diagnosis of ADHD was obtained via maternal report. EEG was recorded using a stretch cap (Electro-Cap, Inc Eaton, OH: E1-series cap) in the extended 10/20 system pattern. Recordings were made from 26 electrodes located equidistant across the scalp.

Electrode impedances were kept under 20k ohms. The electrical activity from each lead was amplified using separate bioamps (James Long Company, Caroga Lake, NY). During data collection, the high-pass filter was a single pole RC filter with a 0.1 Hz cut-off (3 dB or half-power point) and 6 dB/octave roll-off. The low-pass filter was a two-pole Butterworth type with a 100-Hz cut-off (3 dB or half-power point) and 12 dB/octave roll-off. The EEG signal was digitized at 512 samples per second for each channel so that data were not affected by aliasing. The acquisition software was Snapshot-Snapstream (HEM Data Corp, Southfield MI). Prior to the recording of each subject, a 10 Hz, 50 μV peak-to-peak sine wave was input through each amplifier and digitized for 30 sec. This signal was analyzed and the resulting power values used to calibrate the EEGs.

After the EEG electrodes were applied, children participated in eyes open, eyes closed, and quiet VIDEO baseline events to collect resting EEG data. Then the children completed a battery of cognitive tasks designed to assess various aspects of attention [52] using the child version [53]

of the Attention Network Task (ANT) and various aspects of cognition associated with executive functions (e.g., number Stroop, Dimensional Change Card Sort Task, Digit Span Task). Data from the ANT were used in the analyses that are the focus of this report.

The ANT was designed to assess Posner's brain-based attention networks [15] and yields measures of conflict, alerting, and orienting. The test requires the child to respond to a central target (a yellow fish on a light blue background) displayed on a computer screen and indicate whether the fish is facing left or right. The child is instructed to look at the fixation point, above or below which the target will appear. The target may appear with or without flankers (other fish), which may or may not be congruent with respect to direction they are facing. Reaction time responses to the alert cues, spatial cues, and flankers are manipulated to provide an assessment of the efficiency of each of the attention networks. The ANT is divided into 3 blocks of ~5 minutes each, with a brief rest period between blocks. The EEG during the first block and second block were used in these analyses.

After the research visit, EEG data were analyzed using EEG Analysis software developed by the James Long Company. Data were re-referenced via software to an average reference configuration and then analyzed with a discrete Fourier transform (DFT) using a Hanning window of 1 second width and 50% overlap. Power values were computed at each electrode site for theta (4-7 Hz) and beta (13-30 Hz) frequency bands. Power was expressed as mean square microvolts.

4. KNN CLASSIFICATION

In this chapter, we present our first approach for the classification of A and NA subjects. Using the features and channel selection method described in the previous chapters along with the K-nearest Neighbor (KNN) algorithm, in this chapter is explored how separable the A and NA classes are in the feature domains used. Further, in this chapter a confidence level is proposed. Said confidence level speaks to how much confidence there is in the decision that a subject belongs to one class or the other.

The performance of the KNN algorithm is explored when using AR coefficients, reflection coefficients, and line spectral frequencies as features for the classification of A and NA subjects. Since there is no processing, dimensionality reduction, or space warping associated with KNN, the performance of KNN models is an indicator of how separable the NA and A classes are. To the best of our knowledge, we are the first to evaluate this family of features for the classification of A and NA subjects.

In this chapter, the objective is not only to obtain high accuracy, but also to obtain high confidence of classification. This is an important factor to keep in mind because a decision that comes with 100% confidence is a confident decision. If the decision is right, it means that the subject clearly is part of that class, but if the decision is wrong, it should be investigated why the subject is so strongly classified as being part of the wrong class. On the other hand, a decision that comes with near 50% confidence is nothing more than a guess, regardless of whether the decision is right or wrong.

4.1. K-nearest Neighbor Algorithm

The K-nearest Neighbor algorithm, also known as KNN, is a machine learning algorithm that can be used for classification and regression. Unlike other machine learning algorithms, the process of *training* a KNN model consists of storing the data used in training, which makes it one of the simplest machine learning algorithms [54].

For classification, a KNN algorithm finds the K training vectors that are closest in distance to a test vector **x**. Although Euclidean distance is usually used, any other distance metric (i.e.

Hamming distance) or user-defined function can be used to compute the distance between two vectors. Once the K closest training vectors have been found, the label assigned to **x** is that of the most frequent label of the K nearest neighbors. Figure 4.1 provides an illustration of how the algorithm works.

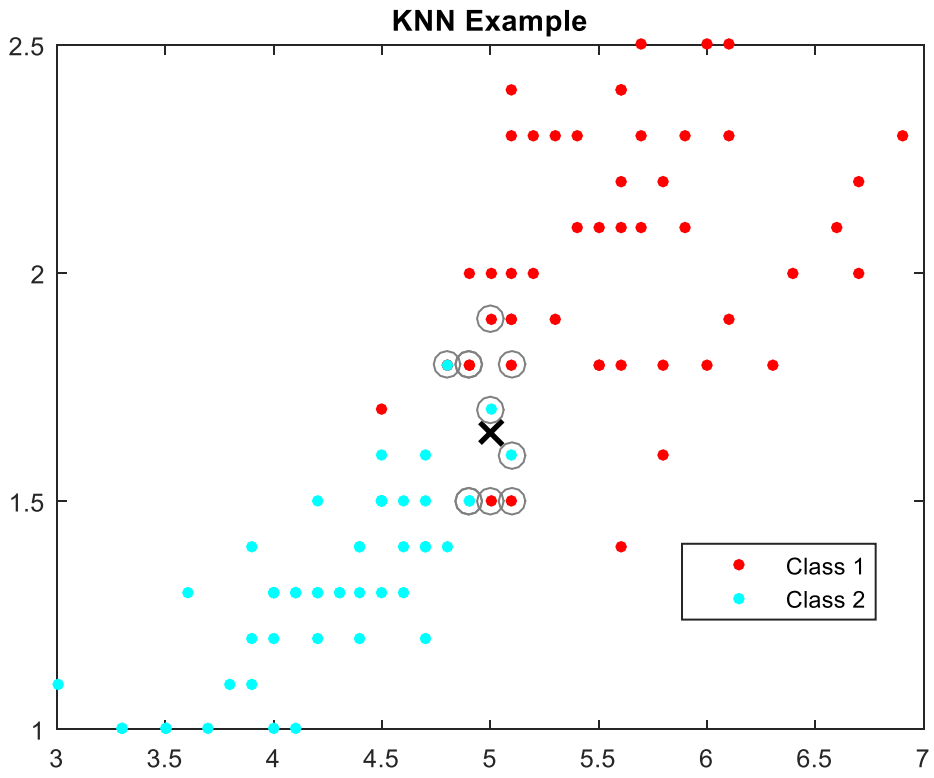

Fig. 4.1: Example of a 2D, 2-class classification using KNN wit K set to 9.

Figure 4.1 presents a 2D, 2-class classification problem using KNN. In the example, the label of a test vector, denoted by **X**, is unknown. Since the value of K was set to 9, the 9 training vectors that are closest to **X** are found (circled in the figure). Note that 5 of the training vectors belong to Class 1 and 4 of the training vectors belong to Class 2. Therefore, **X** is assigned the label "Class 1".

4.2. Confidence in KNN

Since KNN classification is based on vote counts over the total number of votes, a confidence level can be obtained to reflect the level of confidence with which a decision was made. For the example of Fig. 4.1, there are two confidence values:

$$Class1_{conf} = \frac{\#Class1votes}{\mathrm{K}} \tag{4.1}$$

$$Class2_{conf} = \frac{\#Class2votes}{\mathrm{K}} \tag{4.2}$$

Confidence levels are bounded to the range [0,1], with 1 being highest confidence and 0 meaning no confidence. For the example of Fig. 4.1, the test vector was labeled as “Class 1” with a confidence of 5/9. Since this value is between 0.4 and 0.6, it can be considered to be close to guessing.

4.3. Choosing the Value of K

K is the only parameter that can be explored in order to optimize performance. The value of K that maximizes performance always depends on the data. Therefore, a line search from 1 to, typically, half of the size of the training dataset is performed. In other words, KNN models have to be made and then tested for $\mathrm{K} = 1,2,3,\ldots,T$, where T can be 50% of the number of training vectors or a lower number. Then, K is set to the value that maximizes performance [54].

The value of K is also chosen to make the algorithm robust to ties. For example, in two-class classification problems, it is recommended that K be an odd number. In short, it is recommended that for a *C*-class problem, K is set so that *C* is not divisible by K. If *C* is inevitably a multiple of K, then heuristics are used to resolve ties.

4.4. Disadvantages of KNN

There are two known disadvantages associated with KNN. First, just like with any other machine learning model, the performance of a KNN model is highly dependent on the training dataset. However, since no processing is done on the training dataset to create a KNN model, the measured performance of the model will depend on how separable the training dataset is and on how similar the testing dataset is to the training dataset. Other machine learning algorithms, on the other hand, involve processing, which is done to the benefit or the detriment of the model.

The other disadvantage of KNN is the curse of dimensionality, which affects classification algorithms that are highly dependent on distance metrics. If the number of dimensions is too high and/or the N scalar values of the N-Dimensional training vectors are large, the distance between two vectors may become very large (approach infinity), even for neighboring elements, which

causes misclassification. Fortunately, the curse of dimensionality can be addressed by reducing the number of dimensions and/or normalizing the data so that the distances do not approach infinity [55].

4.5. Performance Evaluation

Performance is defined in terms of accuracy of classification, which is defined as the number of true positives (TP) plus the number of true negatives (TN) over the total number of tests.

$$Accuracy = \frac{TP + TN}{\#tests} \tag{4.3}$$

4.6. KNN Experiments

In this section, the experiments are covered that were performed with KNN models. Starting with parameter selection, next the experiments are discussed when 2 subjects (1 A and 1 NA) were used for training and 2 others were used for testing (1 A and 1 NA).

4.6.1. Window Size and Choice of K

The preliminary experiments were based on 4 subjects (all are identified by a numerical value together with the given label): 18316NA, 18396A, 18586NA, and 18606NA. For the selected channels, during the ANT, the distribution of estimated AR orders based on 20 random sets of 0.1 sec of data (51 samples) peaked at 7, 8, and 9. To compensate for the tendency of AIC to overestimate the order of AR models, the order used in this study is set to 7. The 4 subjects were variously paired for training as follows: AB (18396A,18316NA), AD: (18606A,18316NA), CB: (18396A,18586NA), and CD: (18606A,18586NA); for each of these cases, the 2 subjects not part of the pairing for training were used for testing.

To have an idea of the effect of observation window length on classification performance, AR(7) coefficients were computed from windows of 0.05, 0.1, 0.2, 0.5, 1, and 2 seconds long. Given 5 channels were selected, the feature vectors that are being used consist of the concatenation of 5 sets of 7 AR coefficients, i.e. 35-D vectors.

By using two subjects for training (one A and one NA) and the other two for testing, KNN classifiers were built. For training, 200 random observation intervals were used; for testing, using

an overlap of 50%, all windows possible over the ANT interval were used (from 243 2-sec windows to 9776 0.05-sec windows). This process was executed for K = 1, 3, 5, …, 99.

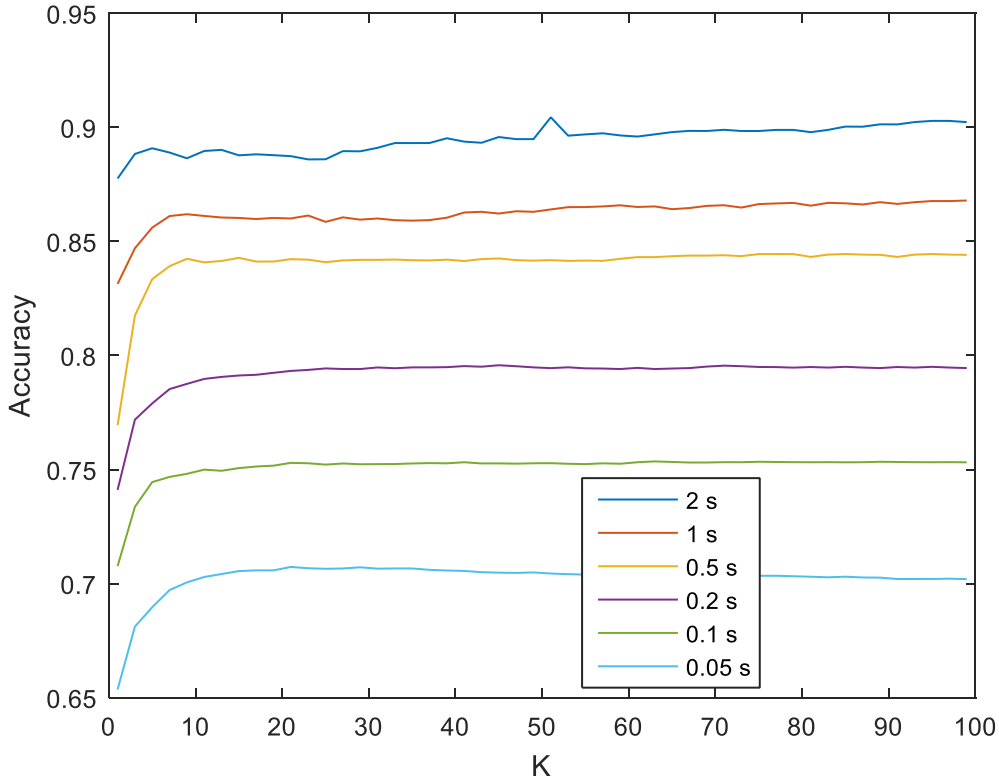

Fig. 4.2: Accuracy for different window sizes and values of K.

Figure 4.2 summarizes the optimization process executed to determine the window size and K to use. The accuracy values displayed in Fig. 4.2 reflect the mean accuracy of the 4 pairs of subjects (AB, AD, CB, and CD) for hundreds of test vectors. From the graph, it is clear that windows of 0.5 s or less should not be used, since the accuracy is below 0.85 for any value of K. Windows of 2 s, seem to achieve higher accuracy than any of the shorter window lengths. For windows of 2 s, there are several local maxima, at K = 5 and K = 99, but the global maximum is at K = 51. Thus, K was set to 51 and the window size to 2 s.

Since the value of K for KNN classifiers always changes depending on the data and the application, the effect of window length is explored in more detail. Figures 4.2 through 4.4 examine how accuracy, true positive rate (TPR), true negative rate (TNR), and confidence levels (A_{conf} and NA_{conf}) change as window length changes.

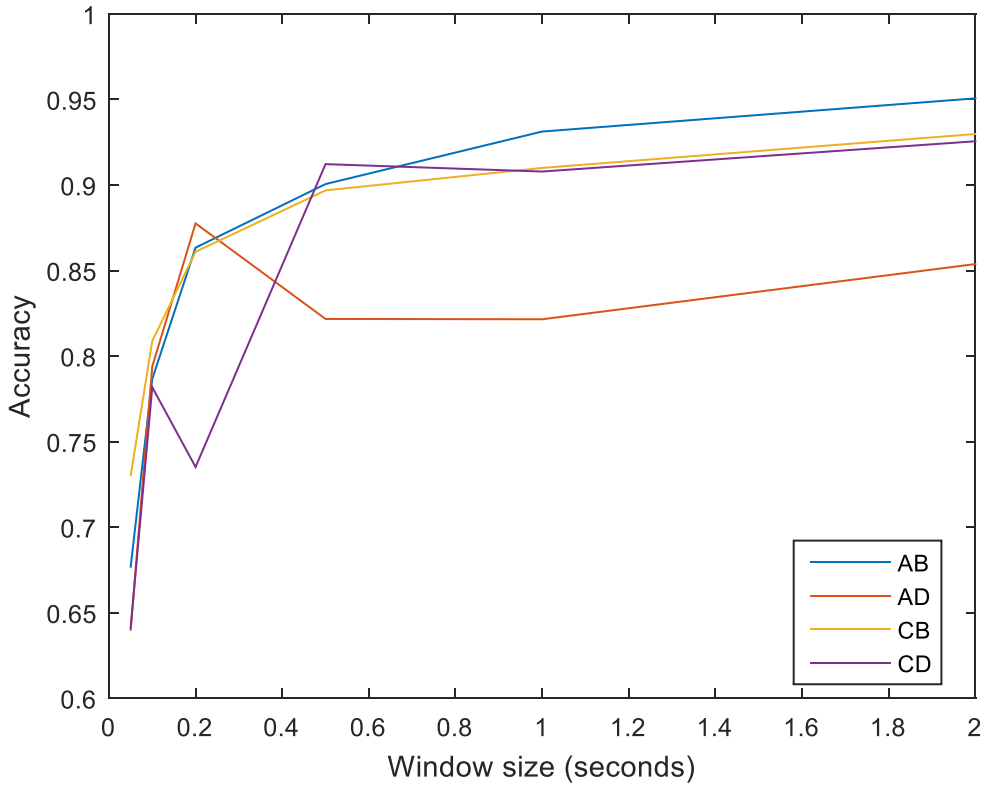

Fig. 4.3: Classification accuracy for 4 pairings as window size changes.

Figure 4.3 shows the results that were aggregated to obtain the dark blue curve of Fig. 4.2 (K=51, 2-second windows). As seen previously, accuracy increases as window length increases. For windows of 2-sec duration, classification accuracy varies from 85% to 95%, depending on which pairings were used for training and testing. From a classification point of view, these results imply that the two classes (A and NA) are separable in the feature domain selected. For windows of 1 s, accuracy varies from 82% to 92%, which is not that far off from its 2 s counterpart. For windows of 0.5 s, accuracy varies from 82% to 91%, which is almost equal to the 1 s case. However, for windows below 0.2 s, accuracy is below 80% even for the best case scenario. Interestingly enough, accuracy increases sharply in going from 50 msec to 100 msec; the latter is perhaps indicative of the size of time-frequency atoms for EEG [56].

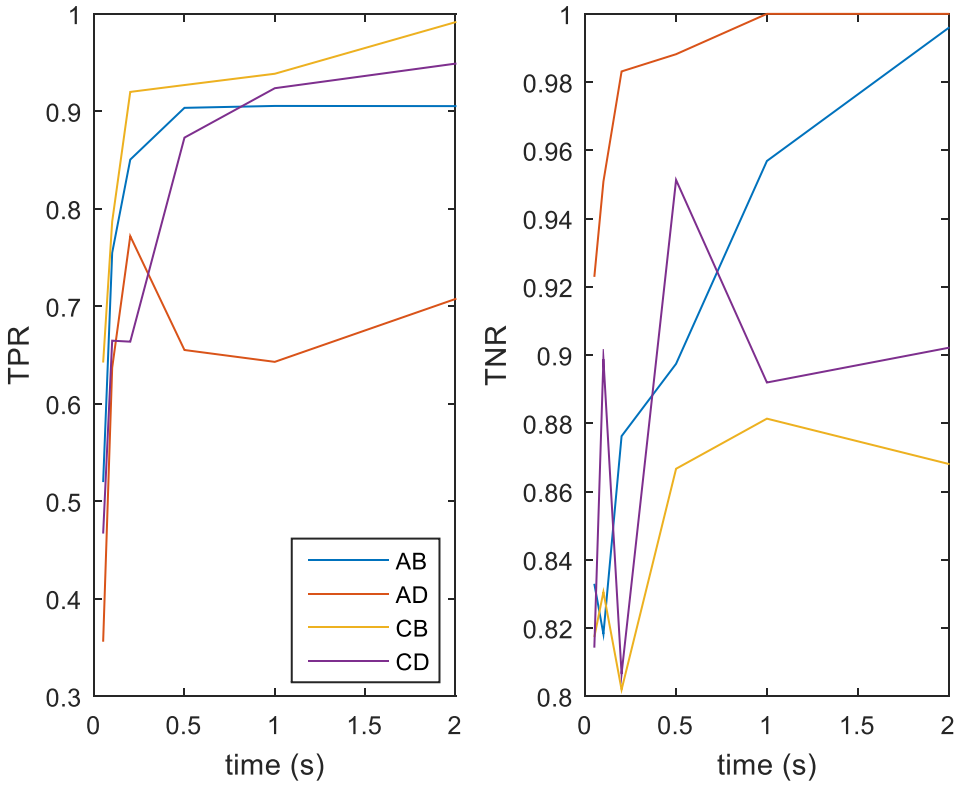

Fig. 4.4: TPR (left) and TNR (right) for 4 pairings as window size changes.

Since accuracy is computed as a function of TPR (A vectors classified correctly) and TNR (NA vectors classified correctly), these will be examined in more detail in the left and right graphs of Fig. 4.4 respectively. As suggested by Fig. 4.4, TPR and TNR tend to increase as window length increases. For pairings CB and CD, the TNR display pronounced up-down-up behavior as window length increases. Unlike these two cases, the TNRs obtained from pairs AD and AB seem to increase more monotonically with window length, and reach TNR of 1 and 0.99 respectively. Interestingly enough, the TPR obtained from CB and CD continuously increase. The TPR obtained from AB reaches almost 0.90 at 0.5 s and stays at that value; the TPR obtained from AD reaches 0.78 at 0.2 s, but then behaves erratically.

Figure 4.4 reveals that there may be some biasing. For TNR at 2 sec, the highest value is 1, for pair AD, and the lowest TNR is 0.87, for pair CB. Note that there is another large TNR, of 0.99, for pair AB. For TPR, the largest value is 0.99, for pair CB, which happens to correspond to the lowest TNR. Likewise, the lowest TPR of 0.70 is achieved by pair AD, which achieved the highest TNR. Hence, pair AD seems to be biased to classify test vectors as NA and pairs CB and CD are

biased towards classifying test vectors as A. Lastly, pair AB is slightly biased to classify subjects as NA.

The left graph of Fig. 4.5 shows how A_{conf} changes with window length and the right graph of Fig. 4.5 shows how NA_{conf} changes with window length. The y-axis label refers to 1- A_{conf} .

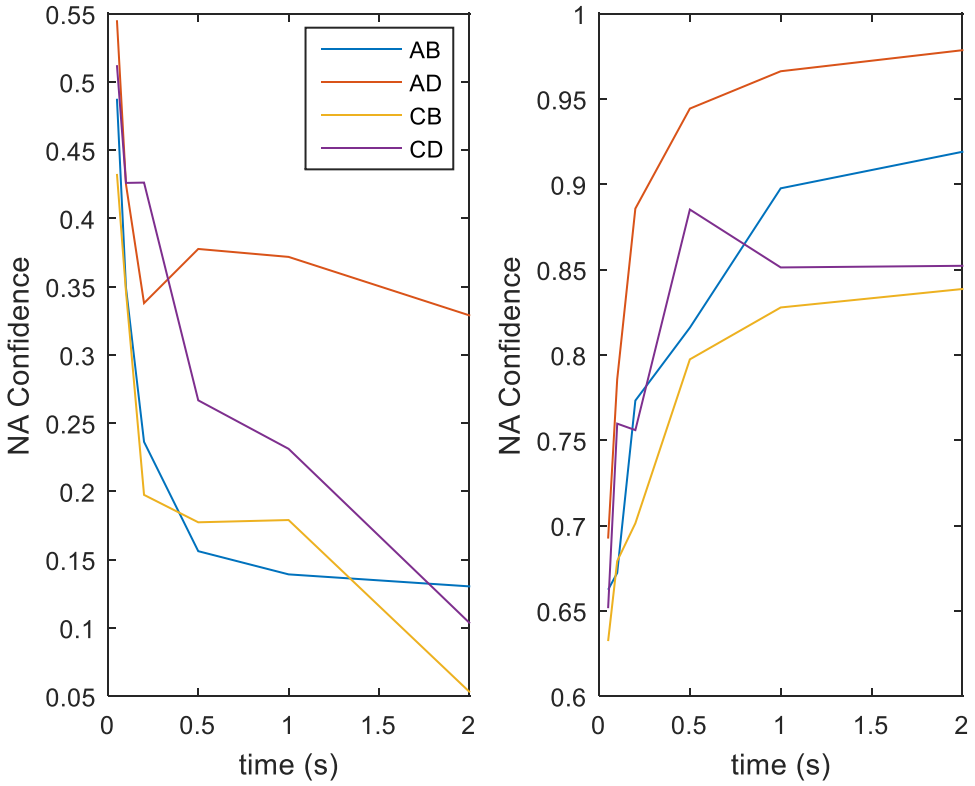

Fig. 4.5: A Confidence (left) and NA Confidence (right) levels for 4 pairings as window size changes.

The pattern observed in Fig. 4.4 is also seen in Fig. 4.5. Values of NA_{conf} that are close to zero represent high confidence as classification in A, and values of NA_{conf} that are close to one represent high confidence in classification as NA. Note that A_{conf} for pairs CB and CD decreases with window length, as expected, and they reach 0.05 and 0.10 respectively for windows of 2 s. These A_{conf} levels are the lowest in the left graph, which indicates high confidence when these pairs are used in training. Nevertheless, the NA_{conf} levels for CB increase as window length increases until they reach 0.84, and those of CD behave somewhat erratically, but reach 0.85. This makes CB and

CD the pairs with the lowest NA_{conf} levels. The opposite is observed for pair AD, whose A_{conf} reaches 0.35 for windows of 2 s, the largest in the left graph, but its NA_{conf} levels of 0.98 are the largest in the right graph. Pair AB seems to have good performance for both cases, but its NA_{conf} is slightly better than its A_{conf} .

In each of Figs. 4.6 through 4.12 the window durations used were 2 sec and K was set to 51. The figure titles on top indicate the sources of the training data, and the legends indicate the sources of the testing data. The Confidence x-axis label refers to NA_{conf} (so that mostly correct *NA* decisions concentrate the histogram on the right, and v.v.). Note that classification confidence less than 0.5 implies a classification error for the associated test vector when classifying NA subjects. On the other hand, confidence levels greater than 0.5 are considered classification errors when classifying A subjects. Generally, when the fraction of votes is between 0.4 and 0.6, the confidence is equivalent to guessing.

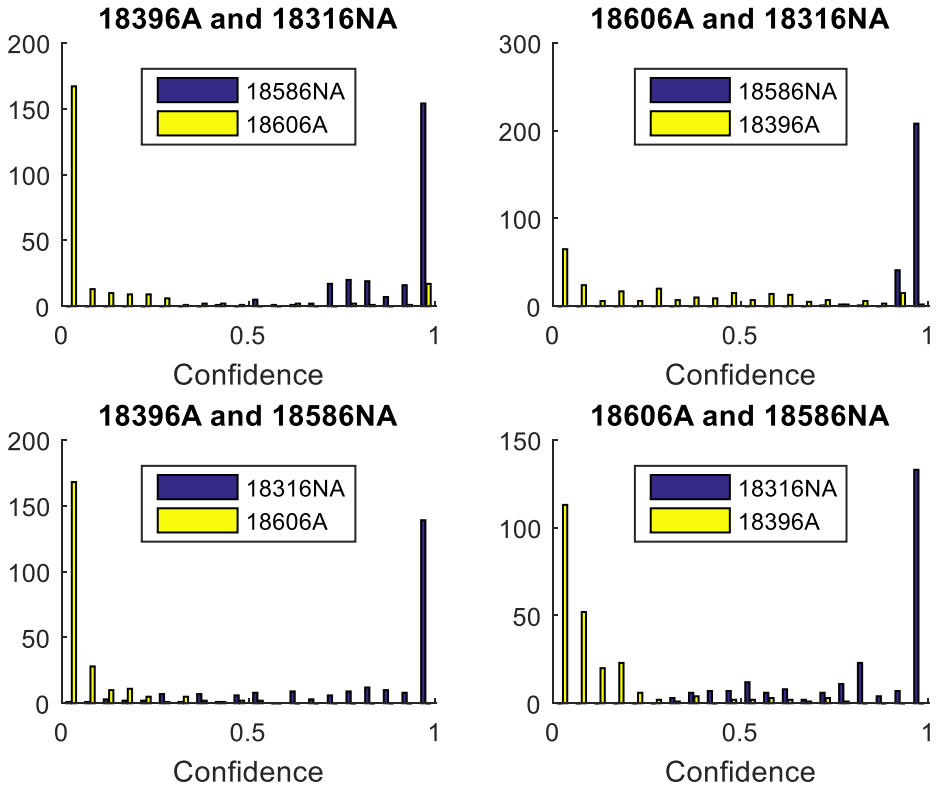

Fig. 4.6: Confidence histograms from training pairings (in title) for test cases (in legend box).

Figure 4.6 shows the confidence histograms when using 2-sec windows in order to classify A subjects (yellow) and NA subjects (blue) based on the same training pairings used for Figs. 4.1 through 4.5. These histograms serve to explore in greater detail the meaning of Fig. 4.5. In each histogram, 240 to 260 test vectors were used. For most test vectors, the confidence of the subject belonging to the NA class is over 0.8. In the top right and top left graphs, which correspond to pairs AB and AD respectively, a large concentration of NA_{conf} levels is close to 1. In fact, for the top right histogram, 250 test vectors were classified as NA with confidence over 0.9. The lowest NA_{conf} level for this pair was 0.72. For the bottom left and bottom right graphs, NA_{conf} is lower. There is a small portion of values that are between 0.6 and 0.4, which indicate guesses, and there is an even smaller portion of values below 0.4, which render the mean NA_{conf} to 93.2%.

Similarly, when testing vectors from the A class most of the decisions are made with over 80% confidence. However, the top right histogram shows more guesses than any of the others and more misclassification errors when testing with A subjects. For the top right histogram, the overall A_{conf} values, for subject 18396A, were 83%. Still, averaging the A_{conf} values over all test cases (95.3%, 95.1%, and 92.5% for pairings AB, CB, and CD respectively) yields 91.9% for A subjects.

4.6.2. Additional Test Subjects

After an additional set of subjects became available, the classification approach was repeated: Training with a pair of subjects and testing with a different pair. The number of training subjects was kept to 2 to test whether or not KNN, with the chosen parameters, would generalize and correctly classify new test subjects.

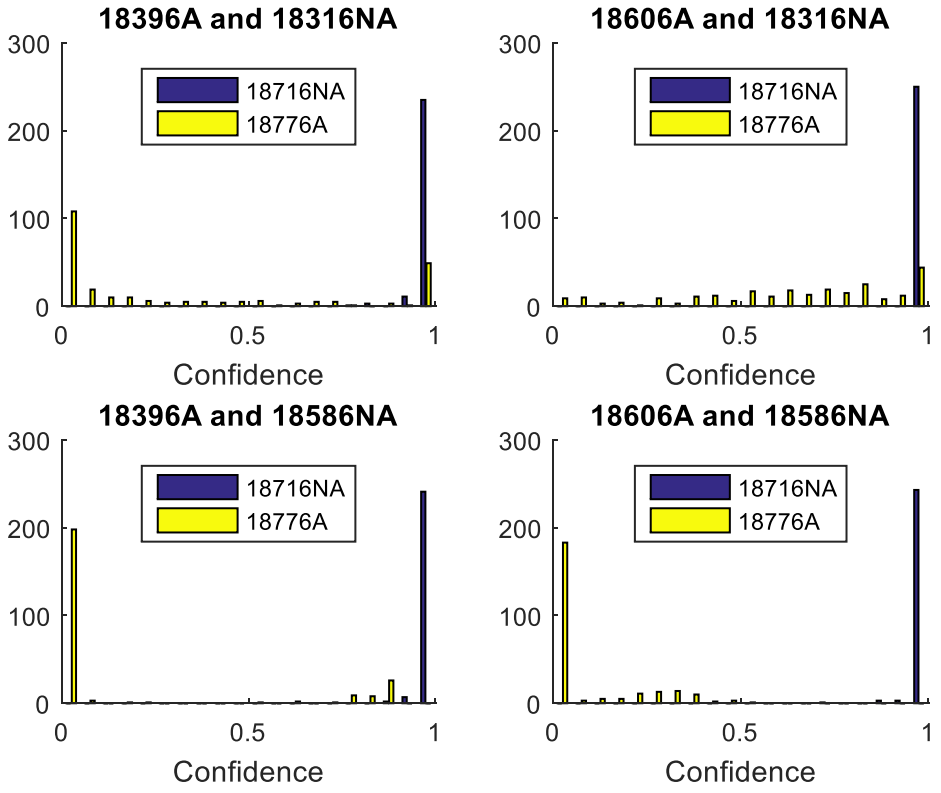

Fig. 4.7: Confidence histograms for original training pairings, when testing with 18776A and 18716NA.

In Fig. 4.7, using the same training pairings as in Section 4.6.1, the additional subject 18716NA is classified correctly for all test windows, and with a very high level of averaged confidence (99.6%), whereas additional subject 18776A is correctly classified for three out of the four training cases shown. It is worth noting that 18776A is highly misclassified when subjects 18606A and 18316NA are used in training, and Section 4.6.1 revealed that this combination is biased towards classifying test vectors as NA. Even so, the average or overall decision, for 18776A, is for belonging to the A class with 92.6% confidence.

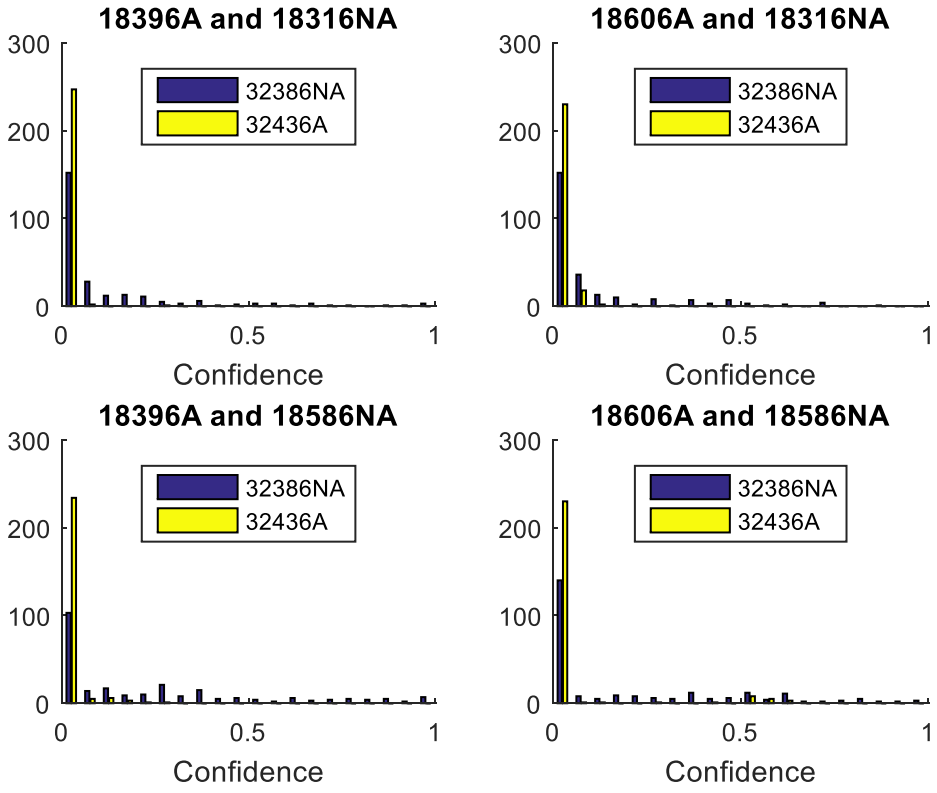

Fig. 4.8: Confidence histograms for original training pairings, when testing with 32436A and 32386NA.

Figure 4.8 shows intriguing results. Subject 32436A was classified correctly for most of its test vectors and for all the original training pairings, with an averaged confidence of 98.4%. However, subject 32386NA is misclassified from almost all test windows, and the confidence in these classifications is a very high 91.4%. The latter result could be due to several reasons. One might be that this subject was the least calm of all NA subjects. Another reason might be that the subject was not performing the activity as instructed. Finally, there is the possibility that the subject was mislabeled. In any case, from a classification perspective, Fig. 4.8 shows that subject 32386NA is much more similar to A subjects than to NA subjects. Moreover, the confidence levels reflected in Fig. 4.8 show that subject 32386NA is very distant from the NA training subjects.

To try to diagnose what might be off with subject 32386NA, it was used for training. Subject 32386NA was paired with each of the 4 original A subjects for training, and for testing, the remaining subjects were used (3 A and 3 NA). Table 1.1 summarizes which subjects were used for training and which ones were used for testing for each case.

Table 1.1: Combinations of training and testing subjects when 32386NA is used for training.

Training	Testing
18396A and 32386NA	18316NA, 18586NA, 18716NA, 18606A, 18776A, 32436A
18606A and 32386NA	18316NA, 18586NA, 18716NA, 18396A, 18776A, 32436A
18776A and 32386NA	18316NA, 18586NA, 18716NA, 18396A, 18606A, 32436A
32436A and 32386NA	18316NA, 18586NA, 18716NA, 18396A, 18606A, 18776A

For these experiments, approximately 500 vectors were used for training and 1500 for testing. Since 32386NA could not be correctly classified, we anticipate that classification will be poor if this subject is used for training.

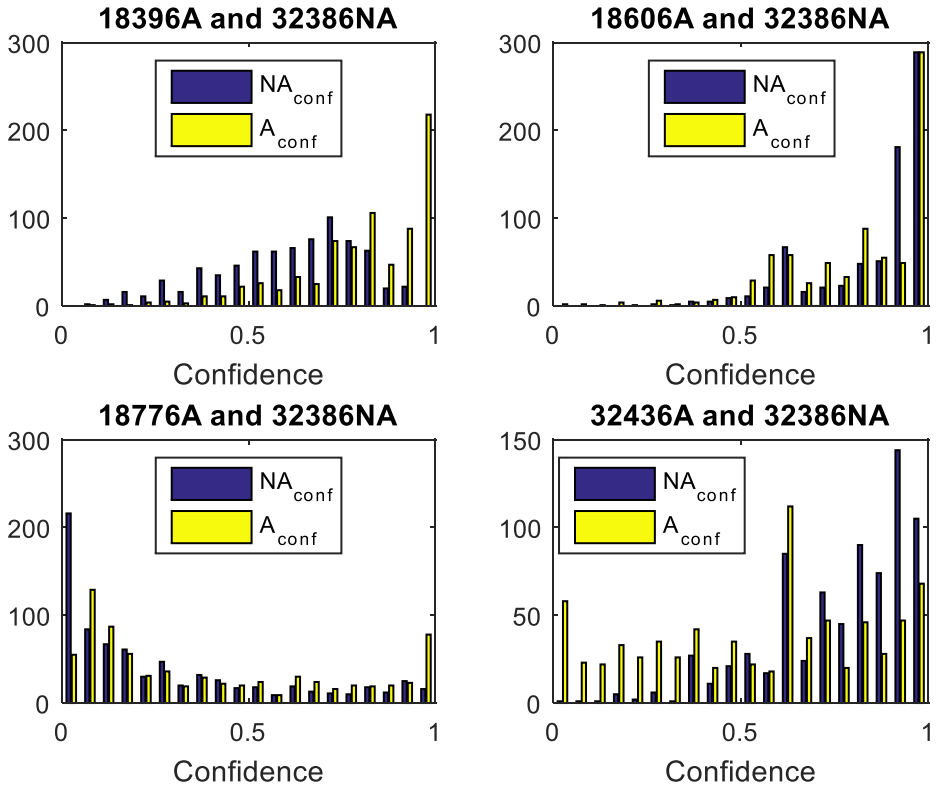

Fig. 4.9: Confidence histograms for pairings involving subject 32386NA.

Figure 4.9 summarizes what occurs when subject 32386NA is used for training along with an A subject. All four graphs support our hypothesis that subject 32386NA carries the wrong label. For the top right and top left graphs, classification was highly biased towards the NA class; most

of the test vectors from the A subjects were classified as NA. For the top left graph, NA_{conf} seems to be slightly Gaussian, with certainties going from almost 0 to almost 1. The top left graph shows high NA_{conf}, which is desired, and high A_{conf} confidence, which is not desired.

The bottom graphs of Fig. 4.9 also show poor performance. For the bottom left graph, histogram values of NA_{conf} are highly concentrated below 0.3, meaning that NA subjects tend to be mislabeled. For the same graph, the distribution of A_{conf} values indicates that classification is usually done correctly for A vectors, but there is a large number of guesses and low confidence in the decision. Lastly, the bottom right graph shows randomness in the classification of A subjects, but NA test vectors are correctly classified most of the time.

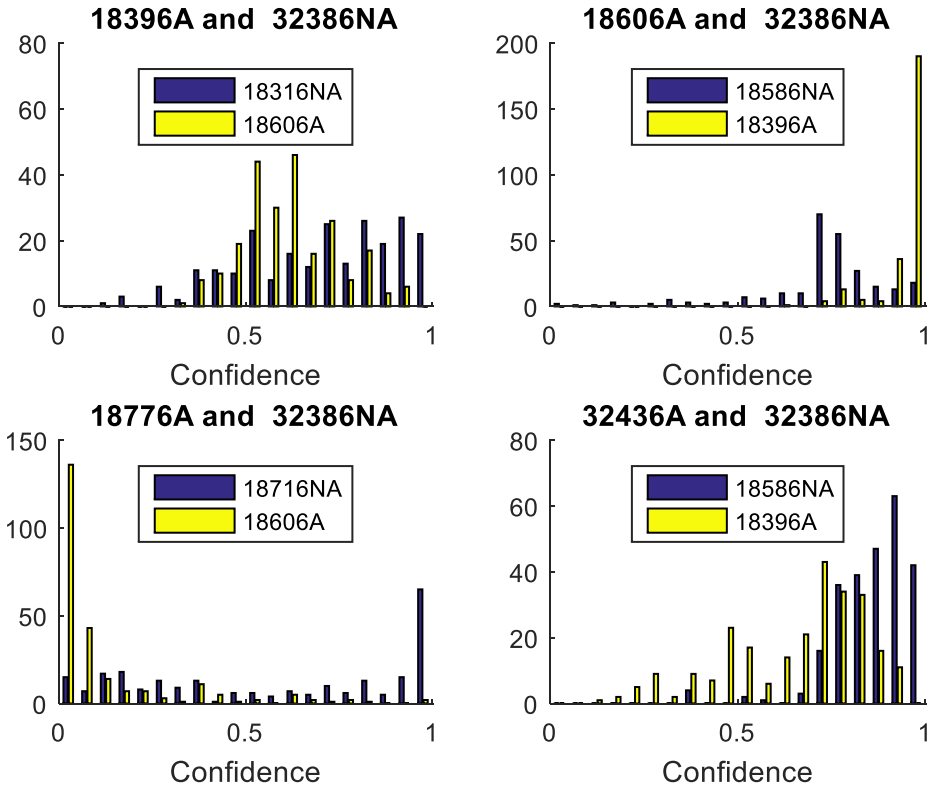

Fig. 4.10: Confidence histograms for pairings involving subject 32386NA and displaying two test subjects only.

Figure 4.10 shows examples of how individual subjects are classified when 32386NA is used in training. The top left graph shows pseudo-Gaussian behavior for the classification of both

subjects; the top right graph displays a highly biased classifier; the bottom left graph shows low A_{conf} values and relatively high NA_{conf} values; finally, the bottom right graph shows poor A_{conf} and relatively high NA_{conf} values. Overall, the average confidence level for these cases is 63.91% for A subjects and 62.31% for NA subjects. These results are only slightly above guessing.

Since classification seemed to be poor when training with 32386NA and an A subject, it was used for training along with another NA subject next. In other words, the label of 32386NA was flipped (designated 32386a) to test what effect that would have on training and classification. For these experiments, training was done by pairing 32386a with each of the 3 NA subjects whose labels do not seem questionable, and tested with the remaining subjects (4 A and 2 NA). The combinations are shown in Table 1.2.

Table 1.2: Combinations of training and testing subjects when subject 32386NA is used for training,

Training	Testing
32386a and 18316NA	18586NA, 18716NA, 18396A, 18606A, 18776A, 32436A
32386a and 18586NA	18316NA, 18716NA, 18396A, 18606A, 18776A, 32436A
32386a and 18716NA	18316NA, 18586NA, 18396A, 18606A, 18776A, 32436A

Just as for Figs. 4.9 and 4.10, approximately 500 vectors were used for training and approximately 1500 were used for testing.

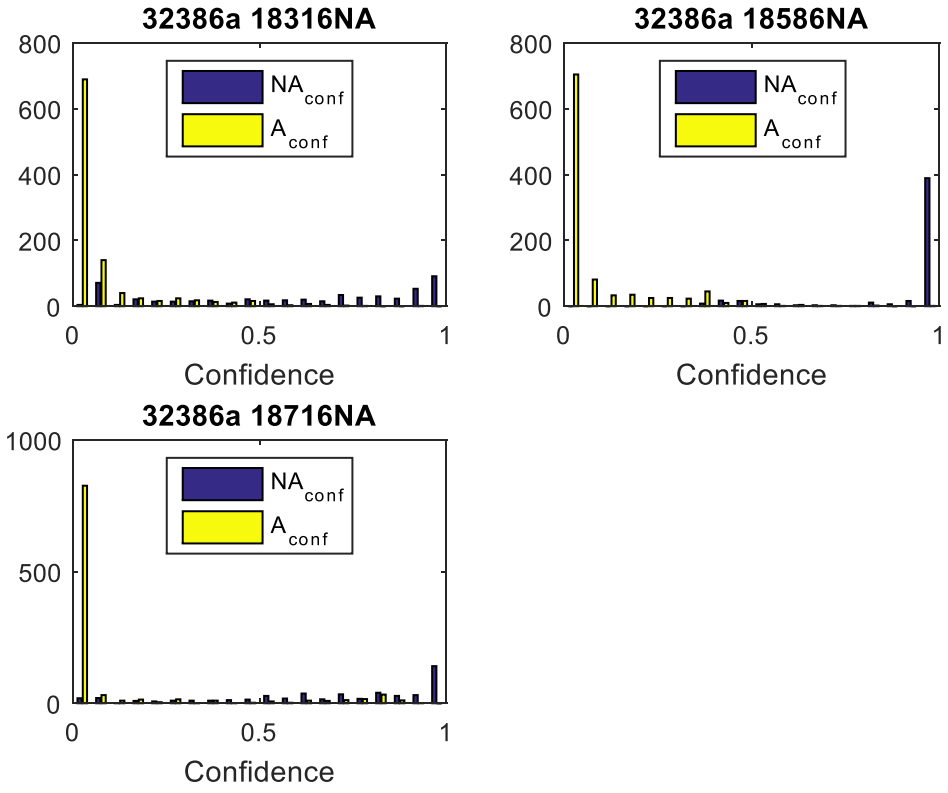

Fig. 4.11: Confidence histograms for pairings involving subject 32386NA and all other NA subjects.

Figure 4.11 summarizes what occurs when subject 32386a is used for training along with an NA subject. In all the graphs, it is difficult to see the NA_{conf} values because the number of A test vectors almost doubles that of the NA test vectors. As can be seen, classification of A subjects is done with very high confidence. Some A vectors were misclassified or correctly classified with low confidence, but the proportion is negligible compared to the high confidence decisions. As far as NA_{conf} goes, the top right graph shows a large concentration of NA_{conf} levels close to 1. The top left histogram shows that classification is relatively poor, and for the bottom left graph, NA_{conf} levels tend to be over 0.7.

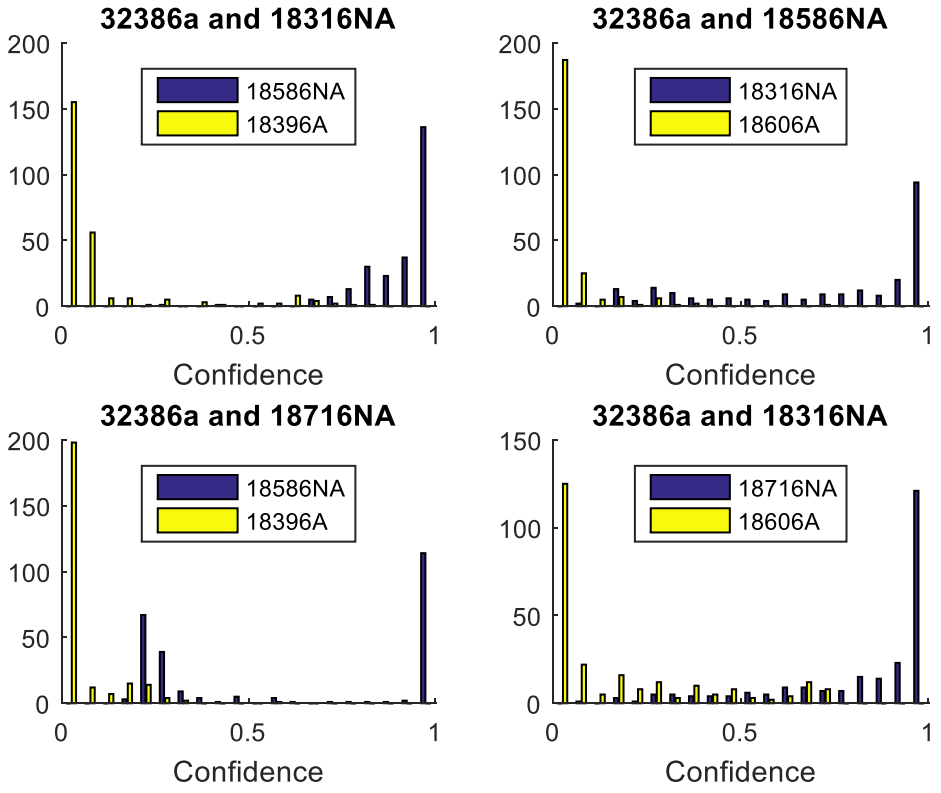

Fig. 4.12: Confidence histograms for pairings involving subject 32386NA.

Figure 4.12 shows some examples of how individual subjects are classified when 32386a is used in training. The graphs repeat the pattern shown in Fig. 4.10: high confidence (low A_{conf} values) across all graphs; NA_{conf} levels are mostly concentrated close to 1 for most test vectors across all graphs, except for the bottom left graph. There is a relatively high proportion of NA_{conf} values under 0.5, but these do not outweigh the performance obtained from the other tests. Overall, the average confidence level for these cases is 94.58% for A subjects and 92.87% for NA subjects. Flipping the label assigned to 32386NA, correct results were obtained, making it likely that subject 32386NA had been mislabeled early in the process.

Since flipping the label of 32386NA from NA to A yielded the best results, it will be used in the subsequent experiments as an A subject. Therefore, there will be 3 NA subjects and 5 A subjects.

4.6.3. Increasing the Training Dataset

In this section the investigation of the effect of increasing the size of the training dataset from 2 (1 A and 1 NA) subjects to 4 (2 A and 2 NA) is reported. For testing, 3 A subjects and 1 NA subject will be used. As a consequence, the number of training vectors used for training will be approximately 1000, and the number of vectors used for testing will be approximately 1000 as well. For these experiments, the number of possible combinations of 2 A subjects and 2 NA subjects for training is 30. Our hypothesis is that the effect of outliers will be suppressed as more data is added, which is expected to result in either an increase in performance or no changes.

Figure 4.13 shows the distribution of accuracy values obtained when training with 2 A and 2 NA and testing with 3 A and 1 NA. The confidence values (bottom) were obtained by averaging A_{conf}, converting $1-A_{conf}$ to NA_{conf} prior to averaging, and NA_{conf}.

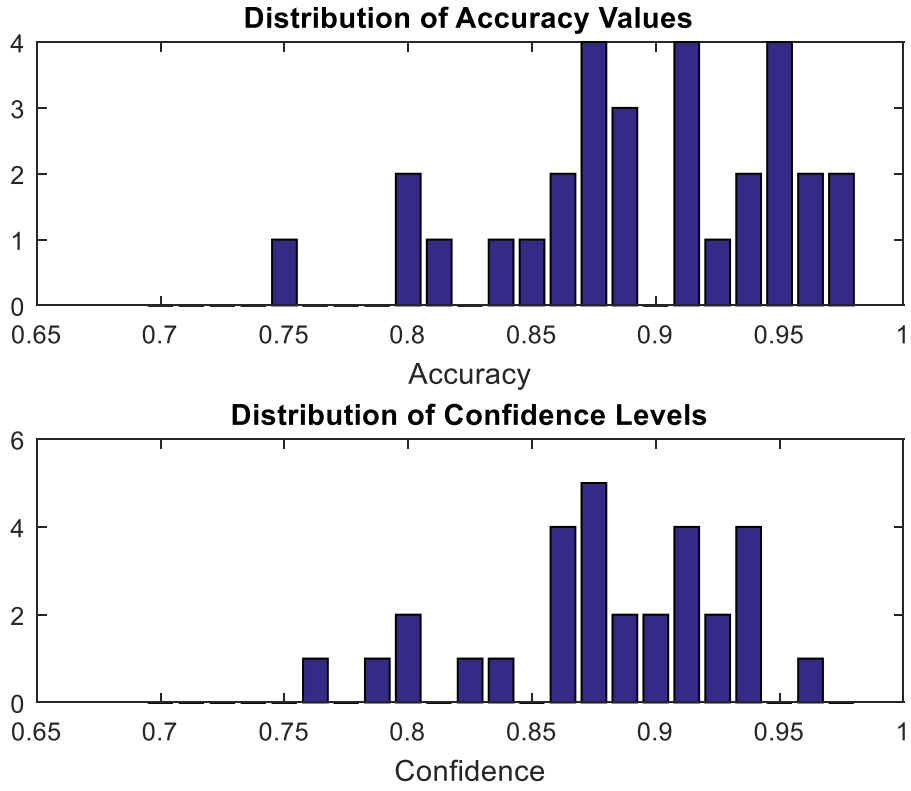

Fig. 4.13: Accuracy values (top) and confidence levels (bottom) obtained from all 30 combinations of 2 NA subjects and 2 A subjects for training.

Figure 4.13 shows that increasing the number of subjects used in training has a strange effect on performance. In the previous section, accuracy varied from 85% to 95%, and so did confidence. However, the top and bottom graphs in Fig. 4.13 show distributions between 75% and 100%, which may suggest that performance in the worst case deteriorated. Further, the mean accuracy for all 30 cases is 89.63% and the mean confidence is 88.14%, whereas the mean accuracy was 91% when 2 subjects are used for training and the mean confidence was 90.48%.

Since what was obtained is the opposite of what was expected, the meaning of these results will be examined more carefully. The TPR, TNR, NA_{conf}, and A_{conf} will be explored in more detail to understand why performance did not improve after adding more data to the training dataset.

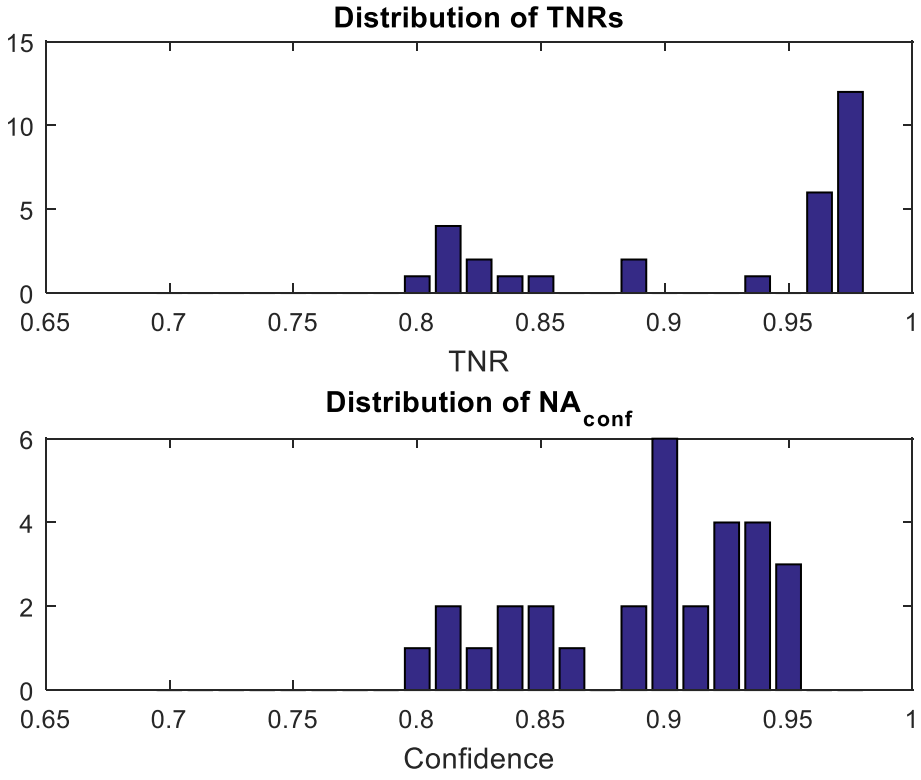

Fig. 4.14: Distribution of TNRs (top) and NA_{conf} (bottom) obtained from all 30 combinations of 2 NA subjects and 2 A subjects for training.

The results shown in Fig. 4.14 are more in line with our expectations. The distribution of TNRs (top graph) shows that most of the TNR are over 0.9. In fact, the mean TNR is 0.9231, and the

worst 0.8008. The NA_{conf} levels display a similar behavior: Most of the NA_{conf} levels are accumulated over 0.9, with a mean of 0.9031 and a worst case of 0.7988.

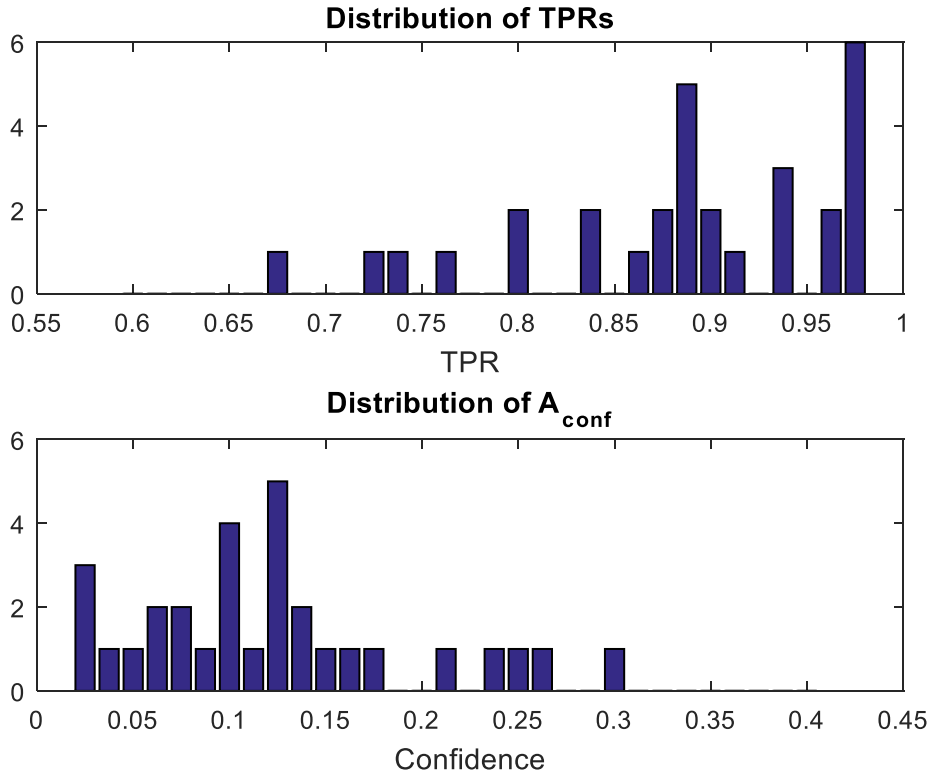

Fig. 4.15: Distribution of TPRs (top) and A_{conf} (bottom) obtained from all 30 combinations of 2 NA subjects and 2 A subjects for training.

The results shown in Fig. 4.15 may provide an explanation for why performance seems to have deteriorated. The distribution of TPR (top graph) shows that half of the TPRs are over 0.9, but the other half is scattered between 0.668 and 0.9. The mean TPR is 0.8877 and the worst is 0.668. A similar pattern can be observed for the A_{conf} levels (bottom). The A_{conf} levels tend to be under 0.15, and the average and worst levels are 0.1227 (87.73%) and 0.2961 (79.31%) respectively.

The TPR and A_{conf} levels decrease the performance of the classifier not only because they are smaller than the TNR and NA_{conf} levels, but also because the classes are skewed. Since testing is done for 3 A subjects and 1 NA subject, TPRs have a higher weight in performance than TNRs.

For instance, the case where TPR is lowest (0.668) happens to be one of the cases where TNR was high (0.9872). For this case, the overall accuracy was 74.78% because it was computed as $((0.668*3)+0.9872)/4$. If the classes had not been skewed, the mean accuracy would have been computed as $(0.668+0.9872)/2$, which equals 82.76%. This value is still below 85%, but it is closer to 85%.

The random variations on the performance of the KNN models trained with 4 subjects were investigated. The first 0.5 seconds of the ANT activity were removed from the dataset to induce a 0.5-second delay on the time-series data, and training and testing were performed as explained at the beginning of Section 4.6.3. Histograms are not shown because the differences are difficult to tell. The standard deviation of the overall accuracy values for the dataset that has a delay is 0.0580, and the standard deviation for the original experiments (which were used to generate Figs. 4.13-4.15) is 0.0580. Further, the mean overall accuracy values over all 30 combinations are 0.896 and 0.902 for the original and delayed versions. Therefore, random variations have a minuscule effect on performance.

In conclusion, contrary to expectations, increasing the size of the training dataset does not have a tremendous impact on performance. The initial results even suggested a decrease in performance, but the TPR and TNR showed that the results were partially due to the fact that class sizes became unbalanced after subject 32386NA was turned into 32386a. Also, the models appear to be robust to random variations.

4.6.4. Reflection Coefficients and Line Spectral Frequencies

In this section we report on the investigation of how classification performance varies when reflection coefficients (RC) or line spectral frequencies (LSF) are used as features, instead of AR coefficients. The results obtained so far indicate that AR coefficients concatenated in feature vectors create a high dimensional space where classification is done with high accuracy and confidence. Since LSF and RC contain the same information as contained in AR coefficients, we explore whether or not KNN classification using RC and LSF can be done with the same, better, or worse level of accuracy and confidence as with AR coefficients.

For these experiments, training and testing are performed as in Section 4.6.2: 2 A and 2 NA subjects for training and 3 A and 1 NA for testing. K was set to 51 and window length to

2 seconds. Just as in the previous sections, the accuracy, confidence level, TPR, and TNR will be explored.

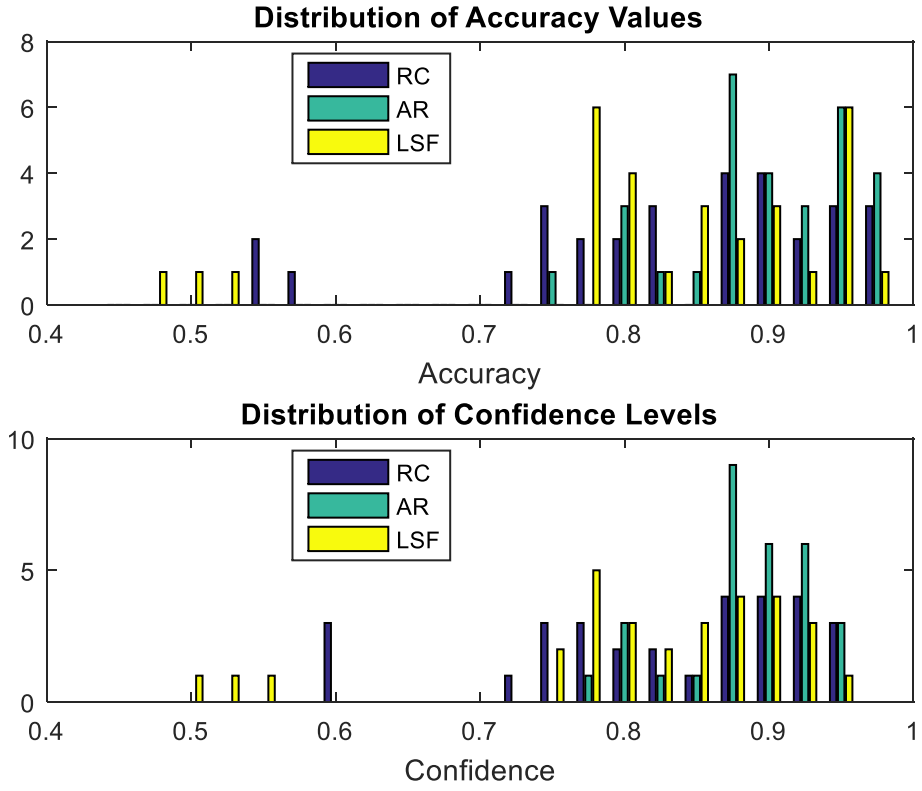

Fig. 4.15: Accuracy values (top) and confidence levels (bottom) when using RC (blue), AR (green), and LSF (yellow) as features.

As seen in Fig. 4.15, classification performance is highest when AR coefficients are used as features. The averaged accuracies (top) reveal that for the 3 kinds of features, the results are concentrated above 0.7, but the worst case for RC and LSF are 0.54 and 0.47 respectively, whereas the worst case for AR is 0.75. For these 3 cases, the mean accuracy values are 0.89 for AR, 0.83 for RC, and 0.82 for LSF.

The observations made for the accuracy values transcend to the confidence levels. Confidence is highest when AR coefficients are used as features, since the average confidence level is 88.14% and the worst is 76.50%. On the other hand, the mean confidence levels when RC and LSF are used as features are 82.72% and 81.28% respectively, and their worst confidence levels are 59.16% and 51.13% respectively.

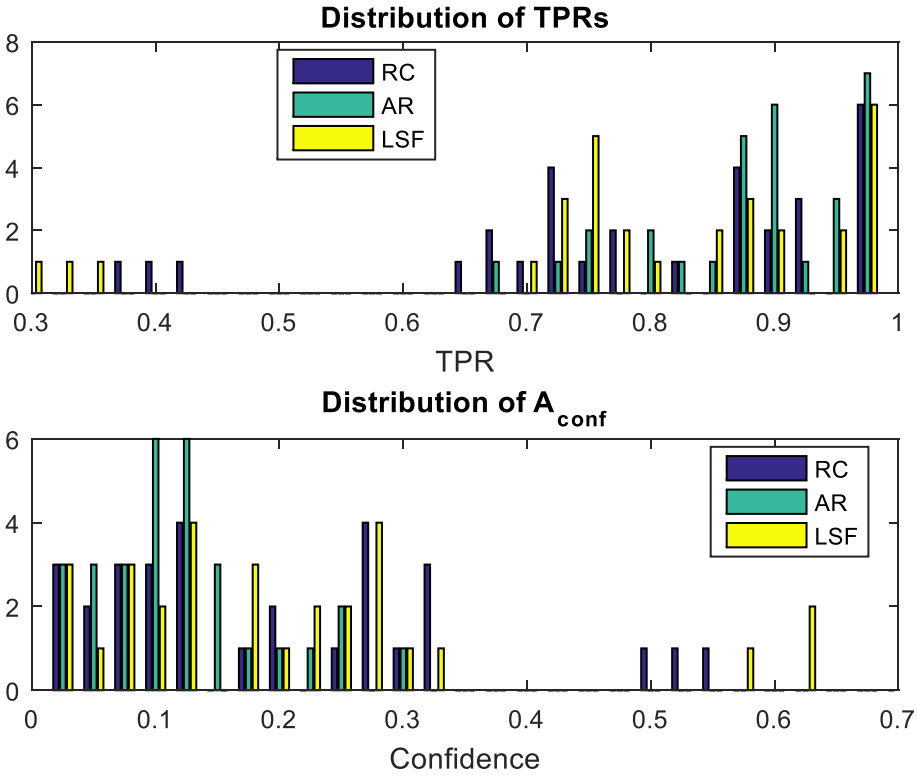

Fig. 4.16: TPRs (top) and A_{conf} levels (bottom) when using RC (blue), AR (green), and LSF (yellow) as features.

The answer as to why RC and LSF were outperformed by AR may lie in Fig. 4.16. The TPRs (top) for the 3 kinds of features are concentrated above 0.65, but the worst cases are between 0.43 and 0.38 for RC, and between 0.30 and 0.36 for LSF, whereas the worst case for AR is 0.668. The mean TPRs are 0.7958, 0.7992, and 0.8877 for LSF, RC, and AR respectively. This evidence shows that KNN classifiers using RC or LSF features are more likely to misclassify A subjects.

The A_{conf} levels (bottom) agree with our recent observations. The mean A_{conf} in the bottom graph are 0.1224 (87.76%), 0.1973 (80.27%), and 0.2088 (79.12%) for AR, RC, and LSF respectively. The worst cases for KNN classifiers using these features are 0.6278 (37.22%), 0.5396 (46.04%), and 0.2951 (70.49%) for LSF, RC, and AR respectively.

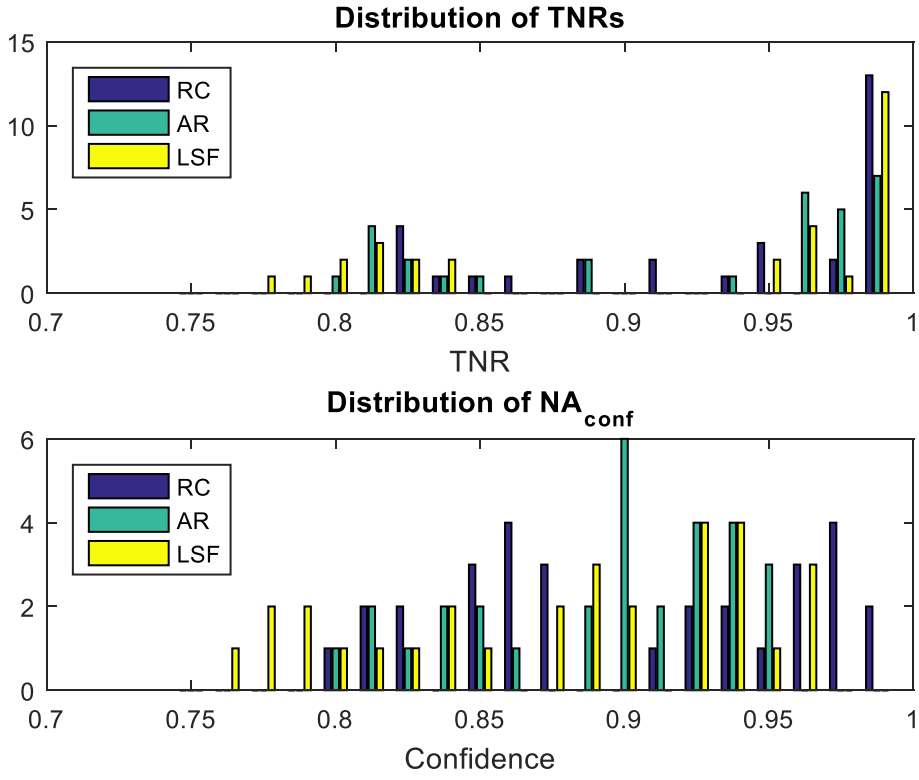

Fig. 4.17: TNRs (top) and NA_{conf} levels (bottom) when using RC (blue), AR (green), and LSF (yellow) as features.

Lastly, Fig. 4.17 provides further insights into the performance of KNN classifiers using these features. The TNRs (top) obtained when using LSF as features (yellow) are either concentrated above 0.95 or below 0.85, with a worst case of 0.7782, the lowest TNR in the graph, and a mean of 0.9203. Interestingly enough, the TNRs obtained when using RC are higher than those obtained when using AR. The worst case for RC is 0.8200 and that of AR is 0.8008; the mean TNR for RC is 0.9369 and that of AR is 0.9231.

The NA_{conf} levels repeat the pattern observed previously: Highest confidence is achieved by RC (mean = 90.14%, worst = 80.59%), followed by AR (mean = 89.36%, worst = 79.80%), and LSF (mean = 87.84%, worst = 79.54%).

These experiments suggest that the feature spaces created by AR feature vectors work better with KNN classifiers for the classification problem at hand than those created by RC or LSF. It seems that classification performance of a KNN classifier using AR as features is

better than that of a KNN classifier using LSF in terms of accuracy, TPR, TNR, and confidence. The same can be said about KNN using AR features versus using RC features. Although NA_{conf} and TNR improved for KNN classifiers using RC, this improvement was at the expense of A_{conf} and TPRs, which makes AR coefficients better than RC for the problem at hand.

These results are both surprising and unsurprising at the same time. They are surprising because even though AR, RC, and LSF coefficients carry the same information, the accuracy found when using AR coefficients as features outperformed that found when using RC and LSF. On the other hand, for KNN, the representation of the data has a large impact on performance. Since AR(7), RC, and LSF have different representations, it is understandable that performance varies somewhat.

To summarize, in this chapter, the KNN algorithm was presented and used for the classification of A and NA. After careful selection of the value of K and that of the window length, experiments were conducted to evaluate the performance of the KNN algorithm in conjunction with AR(7) as features to classify A and NA test vectors. A confidence metric was introduced and was used to question the validity of the labels of one the subjects. The subject was originally labeled as NA, but the confidence histograms determined that the subject was distant from the NA class. As a result, the label of the subject was switched to A, and that label will be used throughout the rest of this thesis. High accuracy (85 – 95% for 2 subjects, and 75 – 100% for 4 subjects) was observed along with high certainty (over 90% for 2 subjects, 80 – 100% for 4 subjects), which shows that the A and NA classes are separable for the feature domain created by the AR coefficients, even without any processing.

5. UNIVERSAL BACKGROUND MODEL

As reported in the previous chapter, KNN classification performance was encouraging, even though no processing was done on the feature vectors to maximize performance. As mentioned in the previous chapter, KNN is an indicator of how separable the classes are in the particular feature domain that was explored. Moreover, we argue that the performance values obtained in the previous chapter can be seen as a lower bound. Therefore, in this chapter the use of a machine learning algorithm is explored, in which statistics are used to maximize the separation between the two classes in order to further maximize performance.

In this chapter, Gaussian-Mixture-Model-based (GMM) universal background models (UBM) are explored for the classification of A and NA subjects. UBMs have been used in the past for speaker verification and identification, and have achieved high levels of accuracy under different noise conditions [57, 58]. Moreover, GMMs and UBMs have recently been studied for the detection and classification of EEG patterns [47, 59].

The hypothesis addressed here is that a UBM can potentially address the shortcomings of other classification schemes. Over the last 30 years, the A/NA classification problem has been tackled by extracting features from EEG data when the subjects are resting with their eyes closed or performing some activity. However, when test subjects do not perform the activity they are instructed to perform, classification accuracy is more likely to suffer (perhaps even to the point of resembling guessing). Therefore, a UBM trained using a large number of feature vectors, extracted from several activities, may make classification more robust.

5.1. Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a model for a probability density function (pdf) expressed as a weighted sum of Gaussian probability density functions. The main reason for using GMMs for classification problems is that Gaussian distributions can approximate any arbitrary pdf [58]. The pdf of a GMM λ is expressed as

$$p(\mathbf{v} \mid \lambda) = \sum_{m=1}^{M} w_m g_m \left(\mathbf{v} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right) \tag{5.1}$$

where $\mathbf{v}$ is an *N*-dimensional feature vector, w_m are the weights, and g_m the individual *N*-variate Gaussian pdf, which have the following form:

$$g_m\left(\mathbf{v}\,|\,\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m\right)=\frac{1}{(2\pi)^{N/2}\,|\,\boldsymbol{\Sigma}_m\,|}\exp\left(-\frac{1}{2}\left(\mathbf{v}-\boldsymbol{\mu}_m\right)^T\boldsymbol{\Sigma}_m^{-1}\left(\mathbf{v}-\boldsymbol{\mu}_m\right)\right) \tag{5.2}$$

where $\boldsymbol{\mu}_m$ is an *N*-dimensional column vector of feature element means and $\boldsymbol{\Sigma}_m$ is the covariance matrix of the feature element vector.

5.2. Expectation Maximization Algorithm

To train the GMMs, i.e. finding the model parameters, the expectation maximization (EM) algorithm was used. In the expectation-step (E-step) of the EM algorithm, the a posteriori probabilities $\psi_{t,c,m}$ of a feature vector $\mathbf{v}_t$ belonging to the Gaussian mixture model λ_c, also known as class membership weights, are computed iteratively over variable *i* in this fashion:

$$\psi_{t,c,k}=P(w_{c,k}\,|\,\mathbf{v}_t,\lambda_{c,i})=\frac{w_{c,k}g_k(\mathbf{v}_t\,|\,\boldsymbol{\mu}_{c,k,i},\boldsymbol{\Sigma}_{c,k,i})}{\sum_{m=1}^{M}w_{c,m}g_m(\mathbf{v}_t\,|\,\boldsymbol{\mu}_{c,m,i},\boldsymbol{\Sigma}_{c,m,i})} \tag{5.3}$$

where $i=1,2,\cdots,I$, $c=1,2,\cdots,C$, $m=1,2,\cdots,M$, and $t=1,2,\cdots,T$, where *I* is the total number of iterations, *C* is the total number of classes, *M* is the total number of mixture components, and *T* is the total number of feature vectors.

In the maximization step (M-step), the weights, means, and covariance matrices that parameterize the Gaussian mixture models λ_c are computed as follows:

$$w_{c,k,i+1}=\frac{1}{T}\sum_{t=1}^{T}\psi_{t,c,k} \tag{5.4}$$

$$\boldsymbol{\mu}_{c,k,i+1}=\frac{\sum_{t=1}^{T}\psi_{t,c,k}\mathbf{v}_t}{Tw_{c,k,i+1}} \tag{5.5}$$

$$\boldsymbol{\Sigma}_{c,k,i+1}=\frac{\sum_{t=1}^{T}\psi_{t,c,k}\left(\mathbf{v}_t-\boldsymbol{\mu}_{c,k,i+1}\right)\left(\mathbf{v}_t-\boldsymbol{\mu}_{c,k,i+1}\right)^T}{Tw_{c,k,i+1}} \tag{5.6}$$

With every iteration *i* the likelihood for which the parameters are computed increases, so that a maximum in the likelihood occurs at the last iteration; however, that maximum may have reached a plateau at an earlier iteration. In other words,

$$l\left(\mathbf{v}_t \mid \lambda_{c,i+1}\right) \geq l\left(\mathbf{v}_t \mid \lambda_{c,i}\right) \tag{5.7}$$

Figure 5.1 illustrates how EM iterations occur when creating 2 models to characterize randomly generated data. The randomly generated data are 2 Gaussians, one with 0.75 variance and mean equal to 1 on both dimensions, and the other with 0.5 variance and mean of -1 on both dimensions. The left graph shows the initial EM iterations, where every ellipsoid in every cluster represents an iteration. Model 0 appears to show only 2 ellipsoids because several overlay the 2 that are separately visible. For Model 1, most of the iterations results are quite visible and distinct from one another. The right graph shows the final iteration results, i.e. the model that was obtained after 10 iterations.

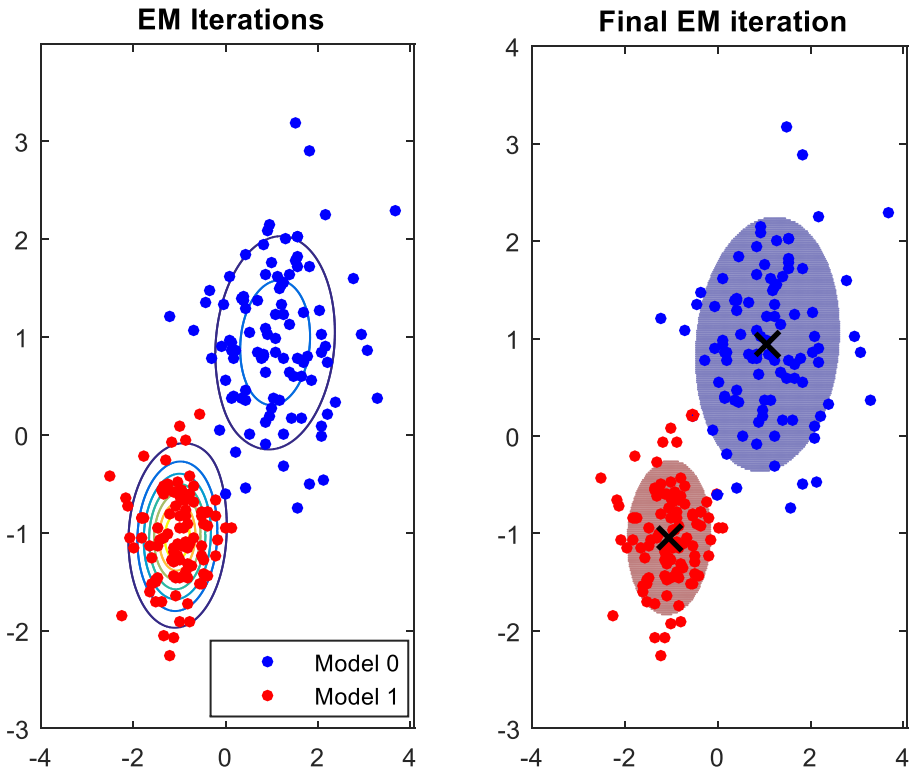

Fig. 5.1: EM iterations (left) and final EM iteration (right).

In this study, EM was executed for a maximum of 15 iterations. This number represents a safe choice. For speech data, when the parameters are initialized randomly, the number of iterations needed tends to be over 10 [60], whereas the number of iterations needed is less than 10 when using K-means clustering for initialization, as is used here.

Figure 5.2 shows an example of how EM with random initialization compares to EM using K-means clustering for initialization. The advantage of using K-means is quite visible: The starting point is different in both plots, and the one when K-means clustering is used for initialization is closer to the convergence value than that from random initialization. The results in the left graph reflect faster convergence to the final value than those in the right graph.

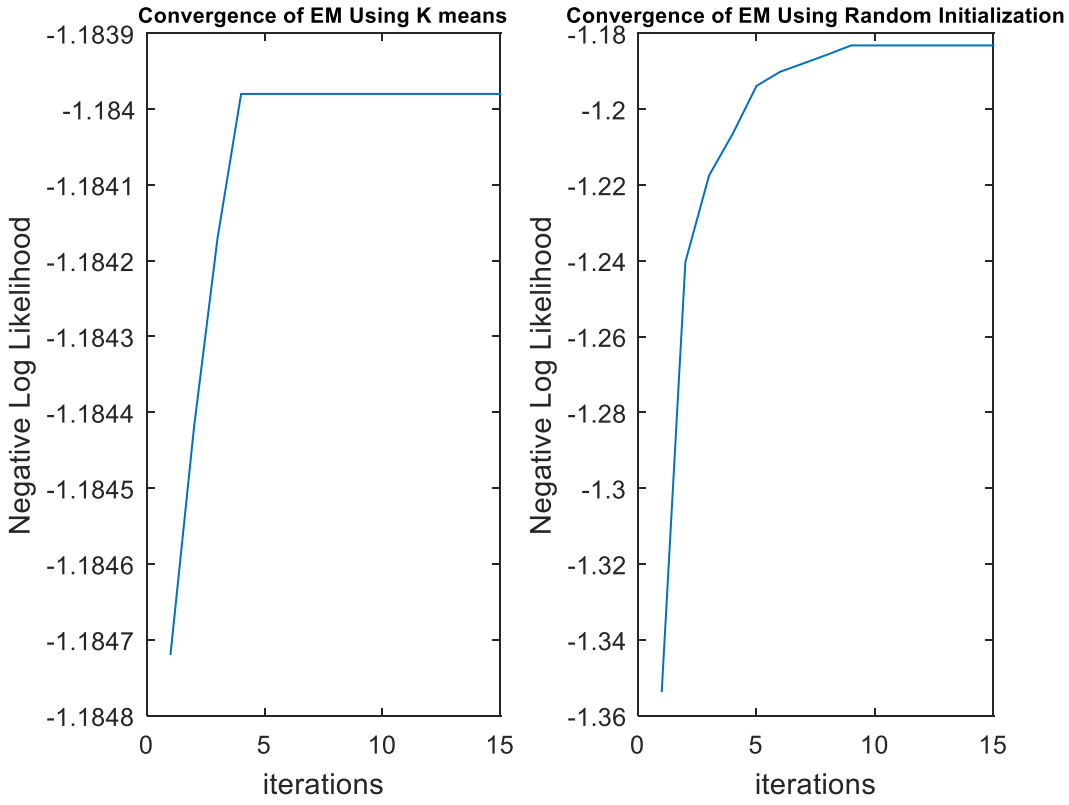

Fig. 5.2: Convergence of EM algorithm with random initialization (left) and convergence using K-means clustering for initialization (right).

5.3. K-means Clustering Algorithm

The objective of K-means clustering is to find N centroids to partition a dataset X that contains T vectors x_t, where $t = 1, 2, ..., T$, into $X_1, X_2, ..., X_N$ clusters with centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_N$ so that

the cumulative distance *J* between the centroids and the vectors that lie within the clusters is minimized. *J* can be expressed as

$$J = \sum_{n=1}^{N} \sum_{\mathbf{x}_t \in X_n} \left\| \boldsymbol{\mu}_n - \mathbf{x}_t \right\|^2 \tag{5.8}$$

The algorithm is initialized by randomly choosing *N* vectors from the dataset and setting them to the initial centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_N$. At every iteration, the algorithm begins with some initial estimates (random estimates for the first iteration), and assigns to every cluster X_n the vectors that are closest to the centroid $\boldsymbol{\mu}_n$.

$$X_{n,i} = \left\{ \mathbf{x}_t : \left\| \boldsymbol{\mu}_{n,i} - \mathbf{x}_t \right\|^2 \le \left\| \boldsymbol{\mu}_{p,i} - \mathbf{x}_t \right\|^2 \forall n, 1 \le n \le N, 1 \le t \le T, 1 \le p \le N \right\} \tag{5.9}$$

The next step is to re-estimate the centroids of the clusters, which is done as follows:

$$\boldsymbol{\mu}_{n,i+1} = \frac{1}{\left| X_{n,i} \right|} \sum_{\mathbf{x}_t \in X_{n,i}} \mathbf{x}_t \tag{5.10}$$

where $\left| X_{n,i} \right|$ is the cardinality of set $X_{n,i}$, $i = 1, 2, ..., I$ were *i* represents the current iteration and *I* the maximum number of iterations. Once $\boldsymbol{\mu}_{n,i+1}$ has been computed, the next iteration starts, with $\boldsymbol{\mu}_{n,i+1}$ being the latest centroid estimates.

In this study, the total number of iterations used was 1000. Moreover, to mitigate the probability of choosing clusters that are not very optimal, the entire process is done 100 times, starting with different random initializations.

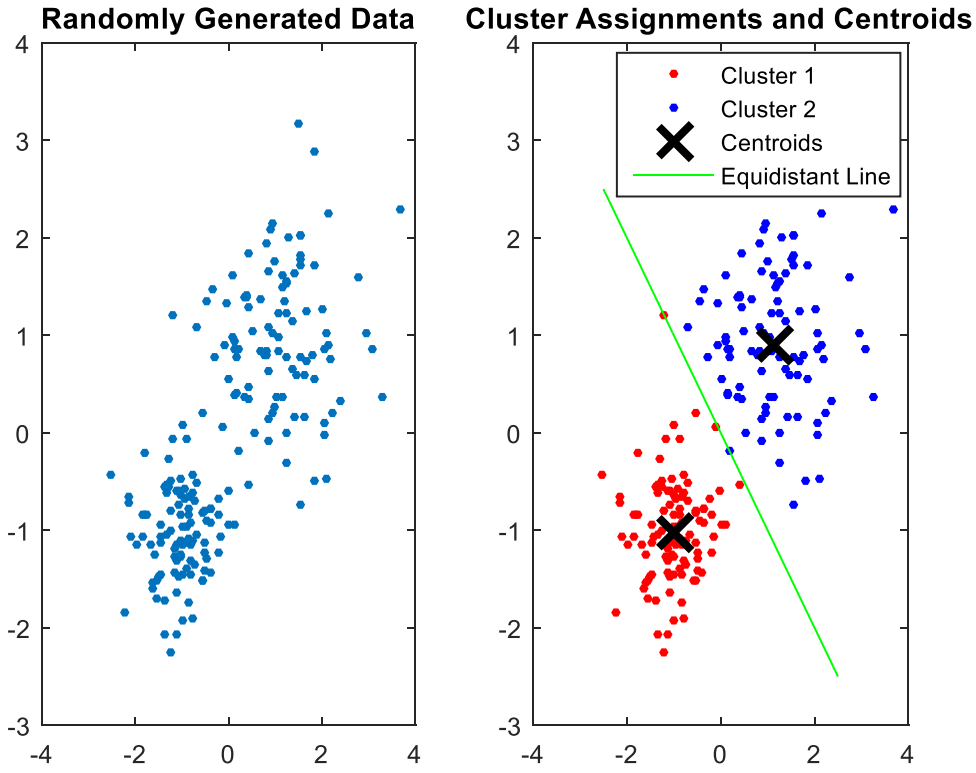

Fig. 5.3: Example of K-means clustering algorithm with data before clustering (left) and after clustering (right).

Figure 5.3 shows an example of using K-means clustering for finding centroids. The number of centroids to compute was arbitrarily set to 2. As seen, the algorithm ends up with 2 centroids that may not be globally optimal, but should be locally optimal. Observe in the figure that all the datapoints on the left of the equidistant line are clustered to Cluster 1 and all the datapoints on the right of the equidistant line is clustered to cluster 2.

5.4. UBM Adaptation

Once the GMMs have been formed, a UBM λ_{UBM} can be created based on the GMMs. In order to do so, each GMM λ_c that will be part of the UBM is adapted by performing the maximum a posteriori (MAP) adaptation. This is done by, first, computing the a posteriori probabilities of each feature vector belonging to the UBM.

$$\tilde{\psi}_{t,c,k} = P\left(w_{c,k} \mid \mathbf{v}_t, \lambda_{UBM}\right) = \frac{w_{c,k} g_k\left(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k}\right)}{\sum_{m=1}^{M} w_{c,m} g_m\left(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,m}, \boldsymbol{\Sigma}_{c,m}\right)} \tag{5.11}$$

In our classification problem, $C = 2$. Therefore, there will be 2 GMMs, λ_{ADHD} and $\lambda_{Non-ADHD}$, and $\lambda_{Non-ADHD}$ will be adapted to form λ_{UBM}.

Sufficient statistics are then computed to obtain the weights, means, and variances of λ_{UBM}. These parameters are the count, first, and second moment of the posterior probabilities found in (5.12) through (5.14).

$$n_{c,k} = \sum_{t=1}^{T} \tilde{\psi}_{t,c,k} \tag{5.12}$$

$$E_{c,k}\left(\mathbf{v}_t\right) = \frac{\sum_{t=1}^{T} \tilde{\psi}_{t,c,k} \mathbf{v}_t}{\sum_{t=1}^{T} \tilde{\psi}_{t,c,k}} \tag{5.13}$$

$$E_{c,k}\left(\mathbf{v}_t \mathbf{v}_t^T\right) = \frac{\sum_{t=1}^{T} \tilde{\psi}_{t,c,k} \mathbf{v}_t \mathbf{v}_t^T}{\sum_{t=1}^{T} \tilde{\psi}_{t,c,k}} \tag{5.14}$$

Once the sufficient statistics have been computed, the weights, means, and variances are adapted. In theory, adaptation should improve performance by making the mixtures in the target class tighter [61]. Adaptation is performed by using (5.15) through (5.17)

$$\tilde{w}_{c,k} = \left[\frac{a_{c,k,w} n_{c,k}}{T} + 1 - \left(1 - a_{c,k,w}\right) w_{c,k}\right] \tilde{\psi}_{t,c,k} \tag{5.15}$$

$$\tilde{\boldsymbol{\mu}}_{c,k} = a_{c,k,\mu} E\left(\mathbf{v}_t\right) + \left(1 - a_{c,k,\mu}\right) \boldsymbol{\mu}_{c,k} \tag{5.16}$$

$$\tilde{\boldsymbol{\sigma}}_{c,k}^2 = a_{c,k,\sigma} E_{c,k}\left(\mathbf{v}_t \mathbf{v}_t^T\right) + \left(1 - a_{c,k,\sigma}\right)\left(\boldsymbol{\sigma}_{c,k}^2 + \boldsymbol{\mu}_{c,k}^2\right) - \tilde{\boldsymbol{\mu}}_{c,k}^2 \tag{5.17}$$

where $a_{c,k,w}, a_{c,k,\mu}, a_{c,k,\sigma}$ are the adaptation coefficients for the weights, means, and variances respectively. They control the balance between the new and old coefficients and are computed by using the following formulae [61]:

$$a_{c,k,w} = \frac{n_{c,k}}{n_{c,k} + \alpha_w} \tag{5.18}$$

$$a_{c,k,\mu} = \frac{n_{c,k}}{n_{c,k} + \alpha_\mu} \tag{5.19}$$

$$a_{c,k,\sigma} = \frac{n_{c,k}}{n_{c,k} + \alpha_\sigma} \tag{5.20}$$

where $\alpha_w, \alpha_\mu, \alpha_\sigma$ are the relevance factors of the weights, means, and variances. In this study, relevance factors were set to 10, since relevance factors between 8 and 20 seem to not affect performance [61]. Outside of this range, performance may be affected positively or negatively [62].

In this study, GMM-UBMs were found using features extracted during various EEG tasks from the NA subjects (impostors). The activities that were experimented with were eyes closed, VIDEO, and ANT. Models were also found to fit the class of ADHD subjects (targets). Figure 5.4 summarizes how classification is done in this study.

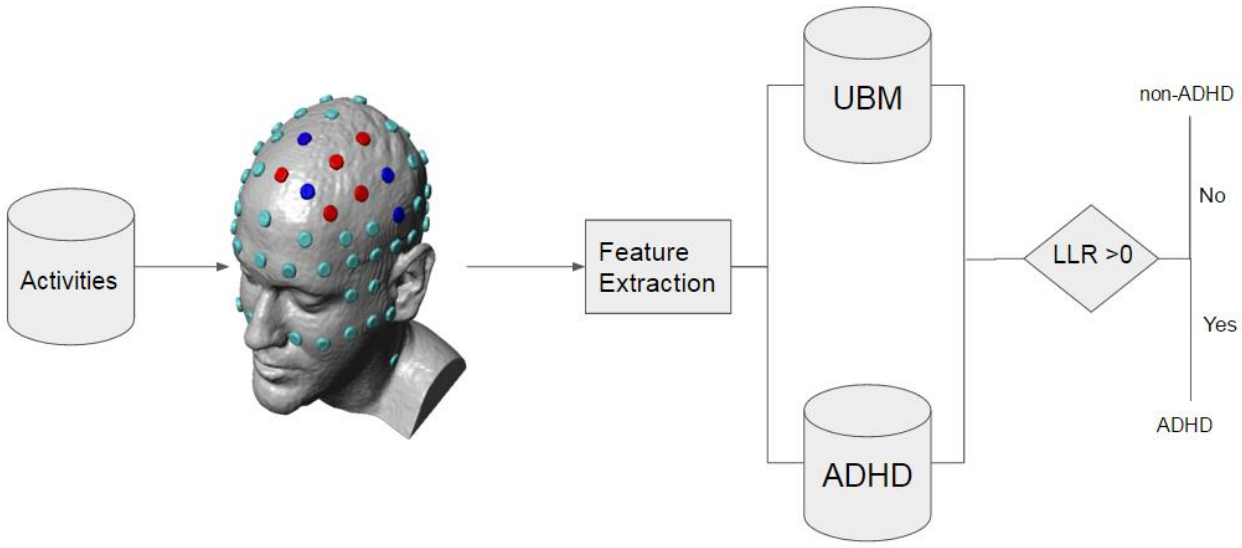

Fig. 5.4: GMM-UBM for the classification of A/NA subjects.

For classification, the log-likelihood ratio (LLR) is used, i.e. the ratio of the likelihood of a test vector $\mathbf{v}_t$ belonging to the ADHD model over the likelihood of $\mathbf{v}_t$ belonging to the universal

background model. If the LLR is greater than or equal to zero, the subject is classified as A, otherwise the subject is classified as NA.

$$LLR = \log\left(\frac{p(\mathbf{v}_t \mid \lambda_{ADHD})}{p(\mathbf{v}_t \mid \lambda_{UBM})}\right) \tag{5.21}$$

To speed up the process of training the GMM-UBMs, the MSR Identity toolbox [63] was used.

5.5. Performance Evaluation

Throughout, the performance of classification in the form of the receiver operating characteristic (ROC) was analyzed. When characterizing the performance of systems that employ biometrics, EER (Equal Error Rate) is often used. EER corresponds to the point on the ROC curve where the miss rate or false negative/NA rate (FNR) is equal to the false positive/alarm/A rate (FPR). In addition, the area under the curve (AUC) of the ROC plot is a scalar metric (ideally approximating 1) that is often used to compactly describe the detection (or true positive/ADHD) rate vs the false positive rate (FPR). AUCs and EERs are thus used as performance indicators.

5.6. Experiments

The experiments were conducted based on 8 subjects: 4 A and 4 NA subjects. However, since the findings of the previous chapter revealed that one NA label may be incorrect, the label was flipped for this chapter. Therefore, that subject was relabeled as A, which means that there were 5 A and 3 NA subjects.

As indicated in the previous chapter, 5 channels were used: Fc1, Fc2, Fc5, Cp6, and C3. AR(7) parameters were extracted from these channels and these parameters concatenated, forming 35-D feature vectors. Feature vectors were extracted during ANT, VIDEO, and eyes closed (EC) activities.

By using 4 subjects for training (2 NA and 2 A) and the other 4 for testing (1 NA and 3 A), GMM-UBMs were built. The training dataset consisted of the 35-D AR parameter vectors extracted from 2-s windows with 50% overlap during the ANT, VIDEO, and/or eyes closed (EC) activities. Given the available recorded data, when overlapping 2-s windows were used, for every subject 423 feature vectors were extracted during ANT; 112 during VIDEO; and 56 during EC.

5.6.1. Number of Mixture Components

For speaker verification and/or identification problems, it is common practice to use one mixture model per speaker [9]. Because this experiment concerns a binary classification problem, two mixture models are formed (one for A and one for NA subjects). To decide how many mixture components should be computed, the effect of the number of mixture components is explored.

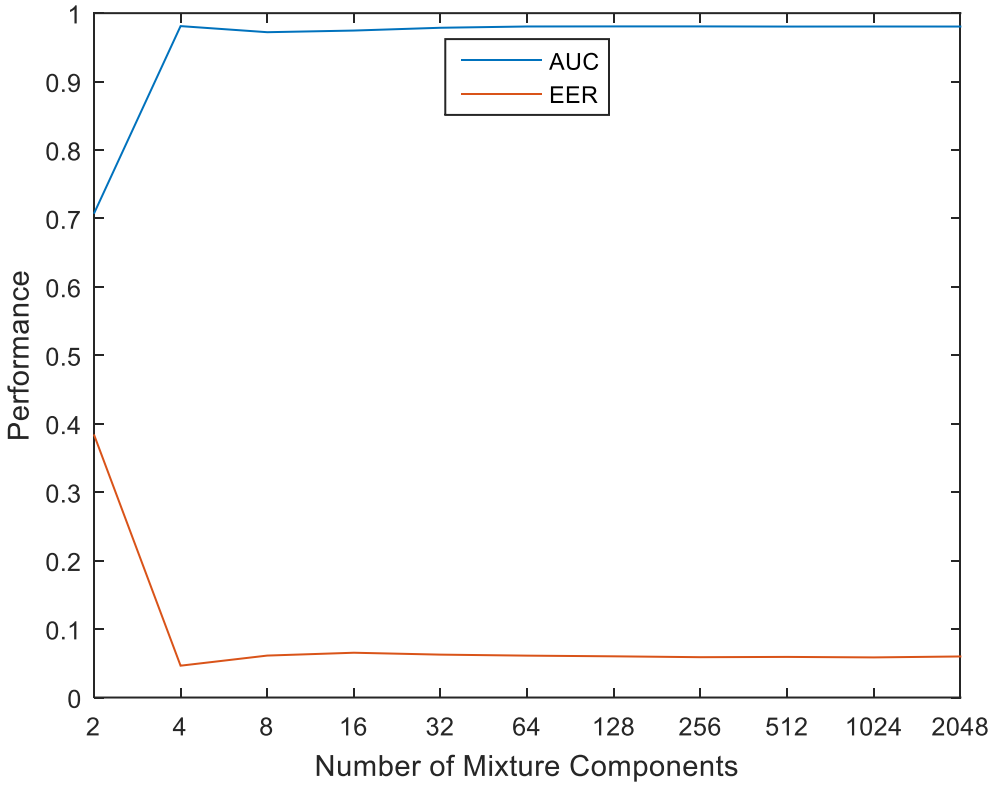

Fig. 5.5: Effect of the number of mixture components.

The results in Fig. 5.5 were obtained using 2 A and 2 NA subjects for training and 2 A subjects and 0 NA subjects for testing during ANT. Peak values are found when the number of mixture components equals the number of subjects used for training. When 4 mixture components are used to fit the data, the AUC is 0.981 and the EER is 0.047. When 8 or more mixture components are computed, the AUC fluctuates between 0.972 and 0.981 and the EER between 0.059 and 0.066. To minimize computational burden and maximize detection, the number of mixture components used in the rest of the experiments is set equal to the number of training subjects.

5.6.2. Effect of Activities

The hypothesis is that the accuracy of ADHD detection depends on the activity performed by the subjects, i.e. some activities elicit stronger discrimination statistics than others. Furthermore, if the activity a test subject performs differs from that presumed to be performed by the training subjects, the miss rate is expected to increase.

For the results reflected in Fig. 5.6, GMM-UBMs are trained with 2 subjects (1 NA and 1 A) and tested with the other 6. The training subjects were paired in (5x3=) 15 different ways in order to train and test with all possible combinations. Four scenarios were considered for training and testing: using eyes closed (EC) data for training and testing (dark blue histogram); using ANT for training and testing (blue histogram); using ANT for training and EC for testing (olive histogram), and using EC for training and ANT for testing (yellow histogram).

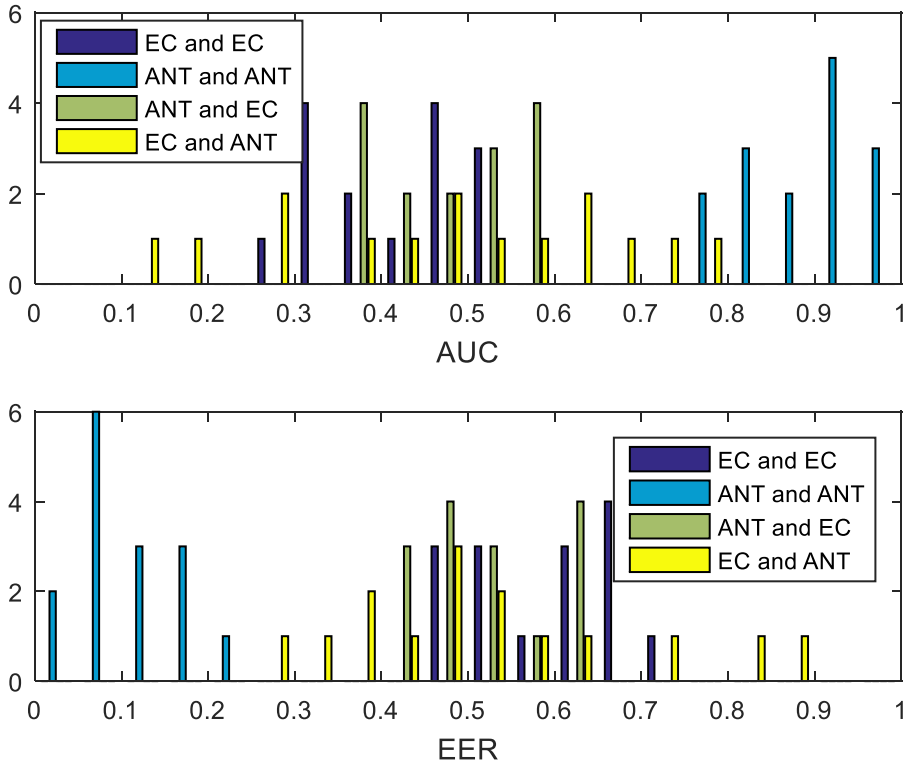

Fig. 5.6: Distribution of AUCs (top) and EERs (bottom) when training/testing = EC/EC (dark blue), ANT/ANT (blue), ANT/EC (olive), and EC/ANT (yellow); all combinations of 2 subjects (1 A and 1 NA) used for training and all other non-overlapping subjects for testing.

Figure 5.6 shows that detection performance is poor when ANT data is used for training and EC is used for testing (olive) and vice versa (yellow); apparently there is not much commonality in the functioning of the brain between when the eyes are closed and when actively paying attention to some task. For both scenarios, there is a large concentration of miss rates between 0.4 and 0.6 for both EERs and AUCs, which is the equivalent of guessing. The average EER for this case is 0.59. Similar results are found when EC data is used for training and testing (dark blue), with the EERs and AUCs concentrated even tighter about 0.5. For the fourth case, when ANT data is used for training and for testing (light blue), the EERs tend to be below 20%, with a mean EER of about 11% and worst case of 23%, and the AUCs tend to be above 0.8, with a mean of 0.8858 and a worst case of 0.76.

Hypothesizing that doing so would increase the performance, training was done next with 4 subjects and testing with the other 4, which resulted in 30 combinations of 4 (picking 2 A and 2 NA, from the available 5 A and 3 NA). The training/testing cases EC/EC, ANT/ANT, ANT/EC, and EC/ANT were explored, and the EERs and AUCs were computed for all 30 combinations. Figure 5.7 summarizes the result of these experiments.

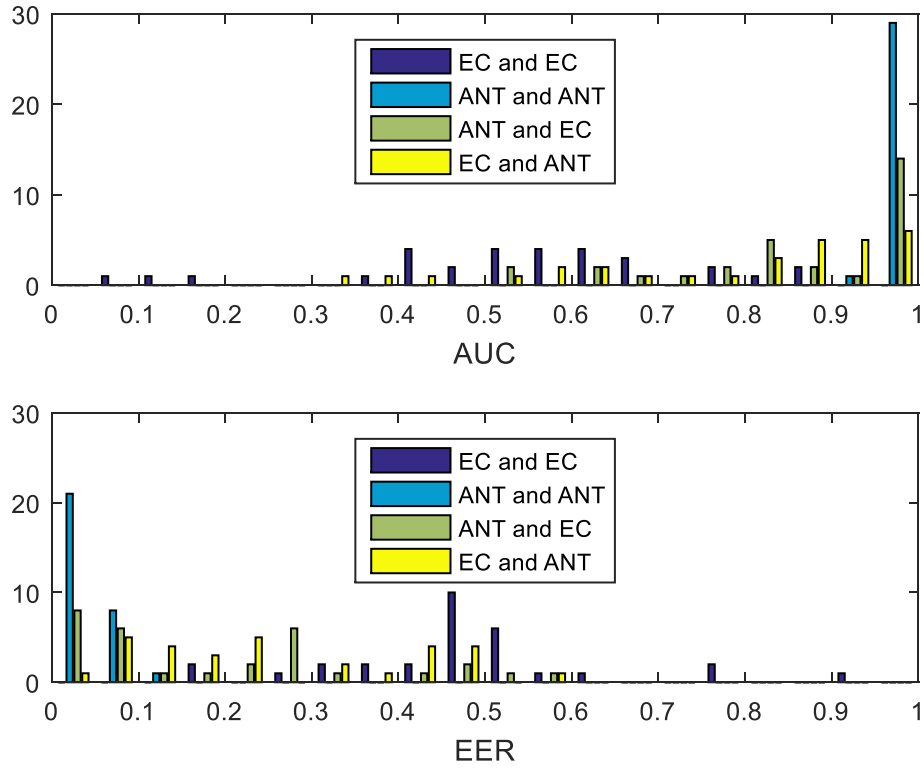

Fig. 5.7: Distribution of AUCs (top) and EERs (bottom) for training/testing cases EC/EC (dark blue), ANT/ANT (blue), ANT/EC (olive), and EC/ANT(yellow); all combinations of 4 subjects (2 A and 2 NA) used for training and all other non-overlapping subjects for testing.

When using EC for training and testing, the EERs range from 0.18 to 0.91 with a mean of 0.49, indicative of a classifier that is (as with less training), still equivalent to guessing. Likewise, the AUCs for this EC/EC scenario are spread between 0.08 and 0.87 with a mean of 0.55, representative of guessing as well. When using ANT for training and EC for testing, the results improve: the EERs are now spread between 0.02 and 0.58 with a mean of 0.22. The AUCs for this case follow a similar pattern: they range between 0.53 and 0.98, with a mean of 0.82, which means that the classifier makes correct decisions most of the time. When EC is used for training and ANT for testing (yellow), the AUCs and EERs are distributed very much as in the ANT/EC case, and the mean AUC and EER are 0.78 and 0.26 respectively. The fourth case (ANT for training and ANT for testing) has improved substantially, with 29 out of 30 AUCs over 0.98 and the outlier at approximately 0.93. The mean AUC is 0.985. The EERs for this ANT/ANT case are now spread between 0.02 and 0.13. The mean EER is now 0.044 with most of the EERs below 0.03. As expected, these results make a strong case for having more subjects for training.

Figure 5.8 explores in more detail the distribution of the case where 4 subjects during ANT activity are used for training.

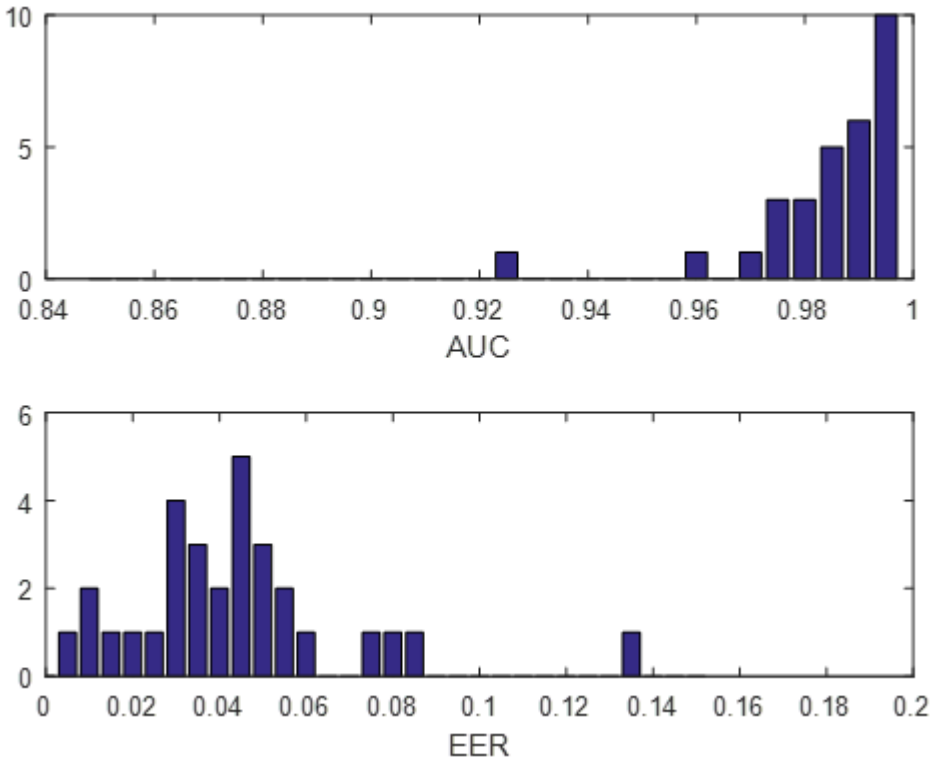

Fig. 5.8: Distribution of AUCs (top) and EERs (bottom) when ANT feature vectors from 4 subjects are used for training and from another 4 ANT subjects for testing.

The distribution of AUCs (top graph) has a long-tail-like shape, with most of the values over 0.98. In fact, 1 in 3 of the AUCs is at 0.995. As pointed out for Fig. 5.7, there is an outlier at 0.925. Except for the one outlier, the EERs (bottom graph) are distributed between 0.005 and 0.085 and centered at 0.045, which happens to be close to the mean EER. There is an outlier with an EER of 0.135, and this outlier is verified to be the one that produced the AUC of 0.925.

These experiments suggest that the more ANT data is used during training, the better the performance; when 2 subjects were used for training, EERs were larger not only because the number of feature vectors was smaller, the number of mixture components was smaller also. When only EC is used for training, classification performance becomes akin to guessing. When ANT was used for training and EC for testing and 2 subjects are used for training, the mean EER was found to be 0.59. On the other hand, when 4 subjects are used, doubling the number of feature vectors in the training dataset, the mean EER drops to 0.22. Finally, when only ANT data is used,

the mean EER is 0.11 when two subjects are used for training and drops to less than 0.05 when four subjects are used for training.

5.6.3. GMM-UBM with EC and ANT data

In this section, the hypothesis will be explored that classification performance degrades as resting data is mixed in – as a contamination – with the modeled training data. GMM-UBMs were trained using 4 subjects and tested with the other 4, those that were not used for training (recall that 5 A and 3 NA subjects are used for these experiments).

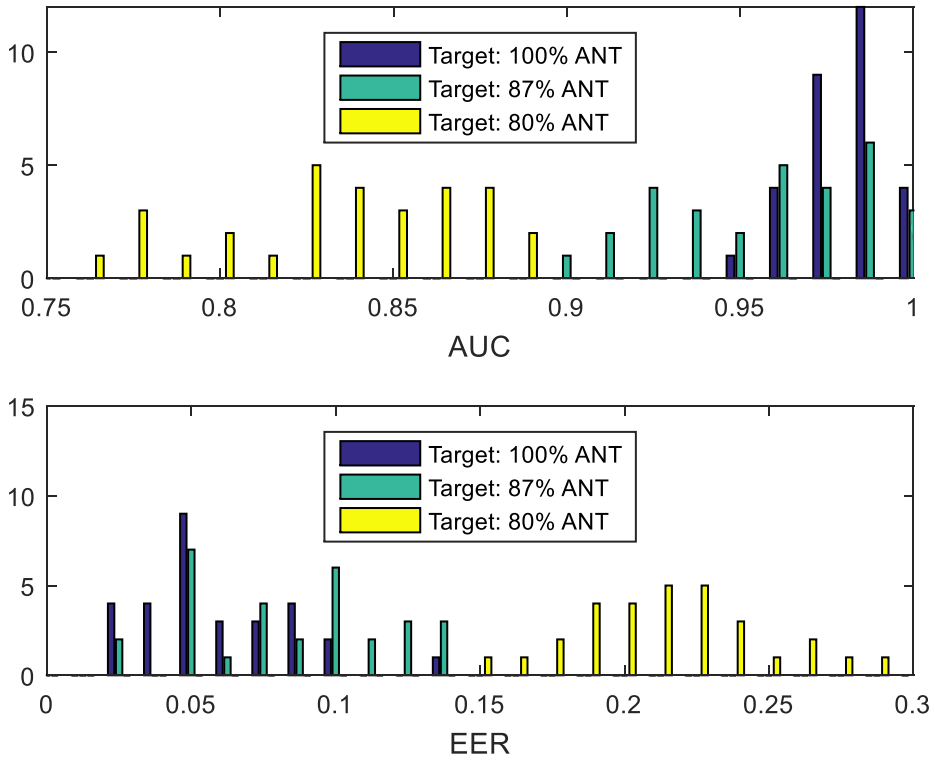

Fig. 5.9: AUCs (top) and EERs (bottom) of GMM-UBMs with ANT+EC (mixed) composition training datasets.

Figure 5.9 shows how performance degrades as ANT data is combined with EC data for training of the UBM. As was shown in Fig. 5.8, when only features extracted during ANT activity were used for training, AUCs vary between 0.96 and 0.995 with a mean of 0.98; EERs vary between 0.02 and 0.08 with a mean of 0.0438. When 87% of the training dataset comes from ANT activity (384 feature vectors during ANT and 56 during EC), performance degrades slightly: AUCs now

vary between 0.9 and 0.995 with a mean of 0.96; EERs vary between 0.03 and 0.13 with a mean of 0.085. When 80% of the data comes from ANT activity (220 feature vectors from ANT and 56 from EC), the AUCs are concentrated between 0.76 and 0.9 with a mean of 0.84. Lastly, for this case, the EERs are distributed between 0.16 and 0.28 with a mean of 0.22.

Figure 5.10 contains sample ROCs based on the distributions shown in Fig. 5.9.

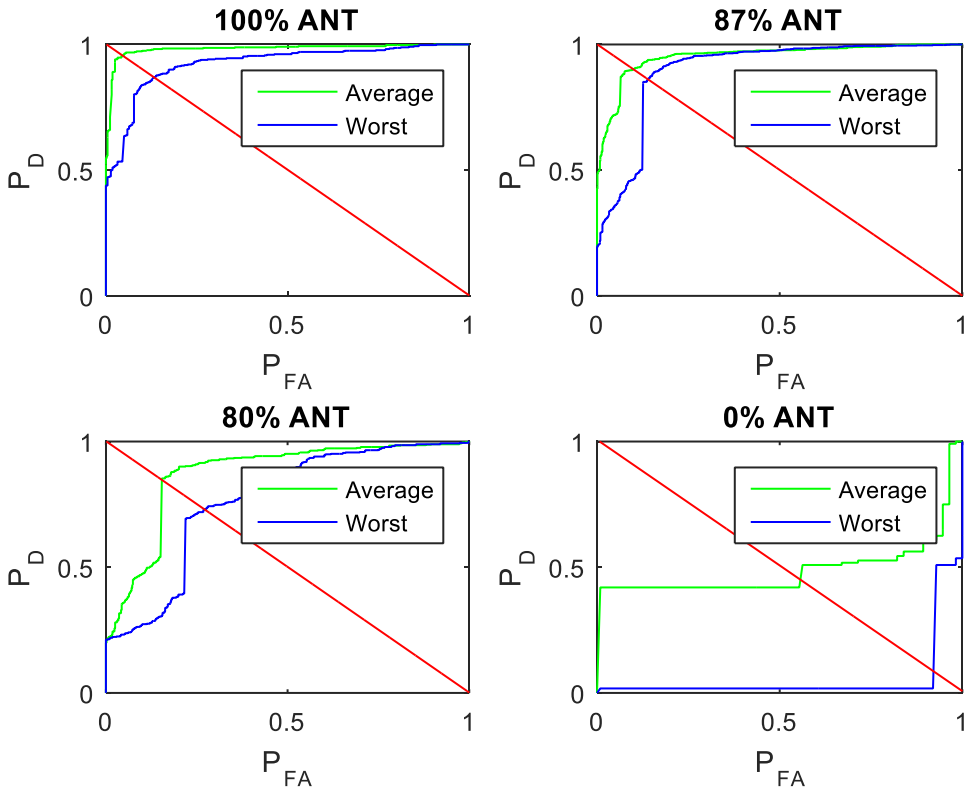

Fig. 5.10: Sample ROC plots with different training datasets.

The top right and left ROC plots show how performance degrades as some EC (resting) data is mixed in with ANT data to train the UBM. The average ROC plot for the top left graph has an AUC of 0.98, whereas the worst ROC has an AUC of 0.92. The average AUC decreases for all the other ROC: 0.96 (top left ROC), 0.87 (bottom left), and 0.55 (bottom right), which is equivalent to guessing. Similarly, the least favorable AUC of the ROC decreases as more resting data is added to the training mixture.

Figure 5.11 shows a different representation of the ROC plots shown in Fig. 5.10 in log scale, which are called detection error trade-off (DET) curves.

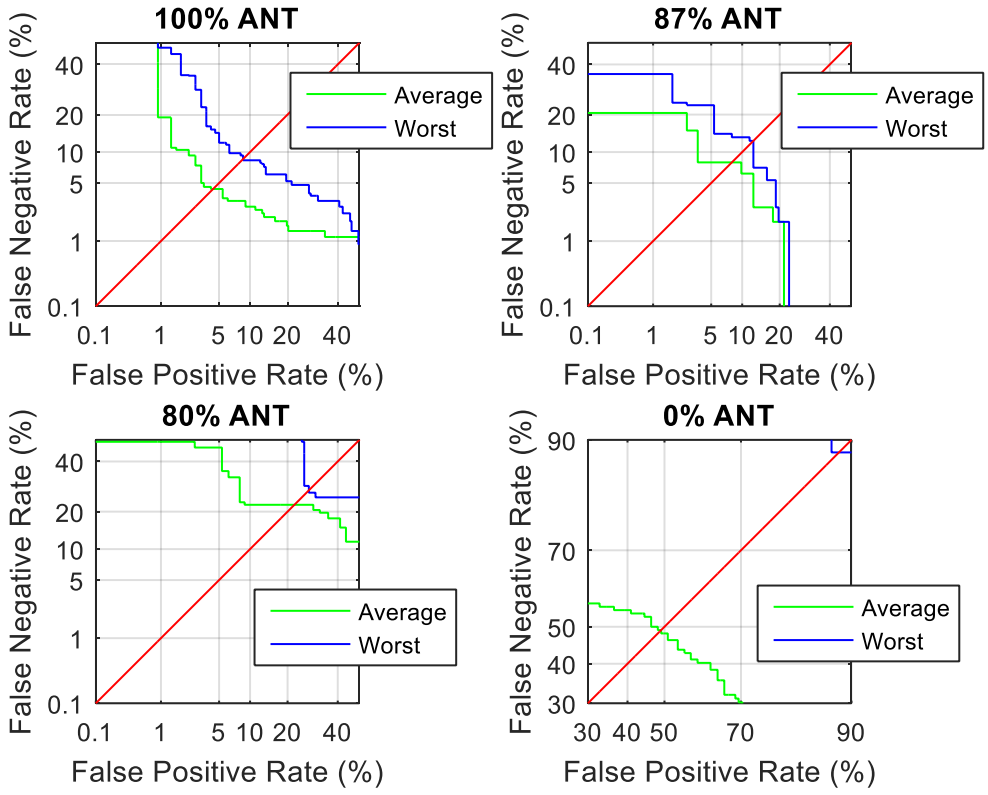

Figure 5.11: Sample DET curves with different datasets.

Through the use of EERs, Fig. 5.11 confirms that performance decreases as resting data is added to the dataset. When 100% of the training dataset is ANT data, the average EER is 0.0438, and the worst EER is 0.0785. The latter implies that the worst case probability of detection is above 92%. When 87% of the dataset is ANT data, the average EER becomes 0.085 and the worst EER becomes 0.13. When 80% of the data is ANT data, the average EER becomes 0.22 and the worst becomes 0.28. Lastly, when 0% of the dataset is ANT data, the average EER is 0.49. In other words, most GMM-UBM models based on EC data will tend to guess whether a test vector belongs to the A class or to the NA class. For models using only EC data, the distribution of EERs goes from 0.1 to 0.9, and is heavily concentrated around 0.5 (see Fig. 5.7).

Figure 5.12 shows how performance decreases when the percentage of resting data in the training dataset increases even further. Since only 56 feature vectors could be extracted during resting EC activity, VIDEO activity was used. The latter is another baseline (resting activity) during which EEG recordings were taken as contamination the subjects.

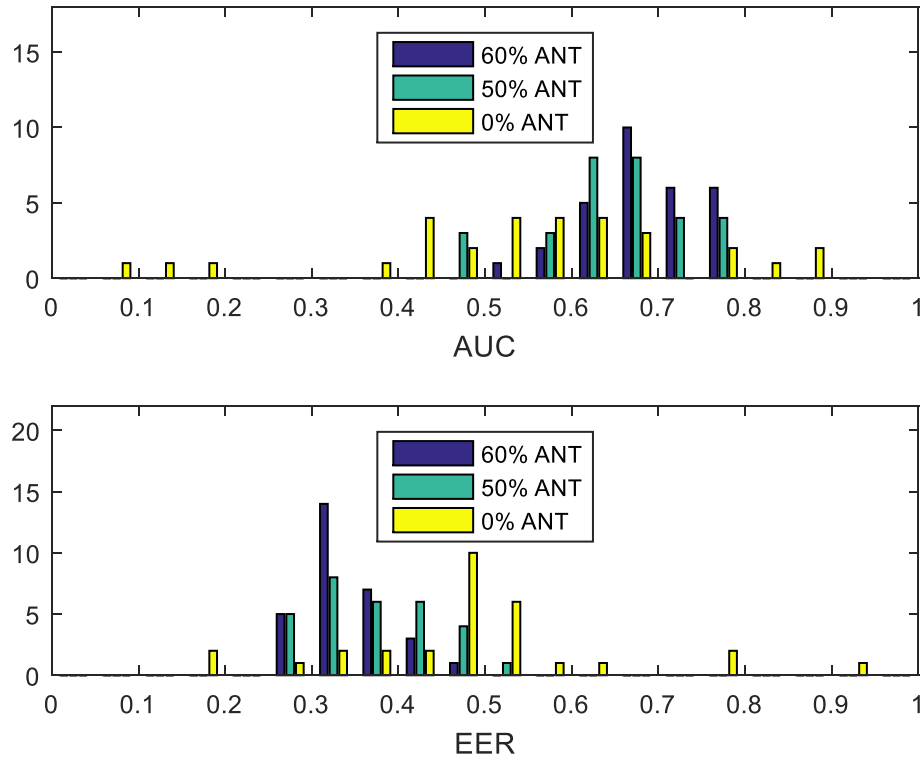

Fig. 5.12: AUCs (top) and EERs (bottom) of GMM-UBMs with ANT+EC+VIDEO (mixed) composition training datasets and same composition testing sets.

In Fig. 5.12, the scenario where 60% of the data was during ANT consisted of 166 feature vectors extracted during ANT, 56 during EC, and 54 during VIDEO. The scenario where 50% of the dataset was during ANT consisted of 166 vectors extracted during ANT, 56 during EC, and 110 during VIDEO. Figure 5.12 shows an exaggeration of the pattern behaviors observed in Fig. 5.10. When 60% of the training and testing data comes from ANT, the AUCs are distributed between 0.52 and 0.78 with a mean of 0.69, which is where most of the AUCs are concentrated. The EERs, for this 60% mixed case, are scattered between 0.28 and 0.48 with a mean of 0.34. When 50% of the training and testing data comes from ANT, performance drops slightly. The distribution of AUCs for this case is similar to the latter, but there are some cases where the AUCs are slightly below 0.5 and the EERs are above 0.5. The mean AUC and EER for this 50% mixed case are 0.65 and 0.38. For these 60% and 50% mixed cases, the performance is slightly, but consistently, better than guessing.

The final case shown in Fig. 5.12 is when all the training and testing feature vectors are extracted from EC. The AUCs and EERs are spread between 0.08 and 0.87, and 0.18 and 0.91 respectively, meaning that classification is akin to guessing. The AUC for 0% ANT, i.e. 100% EC, falls roughly in the 60% range. This is just a little better than a pure guess. This performance is very much like the result found based on phase synchrony processing of eyes closed EEG [64], an entirely different approach than GMM-UBM. As the performance of a KNN approach (Section 4.6.1) was also good when using ANT data, improved detection performance appears to be highly correlated with using an attention task instead of the eyes closed condition.

The mixed training data cases show that even 13% of resting data in the training phase has a perceptible effect on classification performance. While the results make clear that it is inadvisable to use resting EEG as part of the training data, note that inadvertent, temporary inattention during an ANT task could well look like resting data. Mitigation is provided by collecting more data across more subjects, and/or actively detecting (and removing from consideration) resting data segments.

Table 5.1 summarizes the results of the experiments that were done in order to study the effect of different activities on the classification of A/NA. The left column represents the activities used for training and the top row represents the activities used for testing. A tabular presentation of the data (histogram range and mean) is given to avoid using too many histograms.

Table 5.1: Summary of AUC under different training and testing scenarios (percentages in mix).

Testing Training	ANT (100)	ANT + EC (87,13)	ANT + EC + VIDEO (50, 33, 17)	EC (100)
ANT (100)	0.92 to 1.00 **Mean 0.98**	0.92 to 0.99 **Mean 0.97**	0.77 to 0.98 **Mean 0.93**	0.53 to 0.98 **Mean 0.81**
ANT + EC (87, 13)	0.76 to 0.97 Mean 0.89	0.89 to 0.995 **Mean 0.96**	0.753 to 0.91 Mean 0.82	0.17 to 0.87 Mean 0.55
ANT + EC + VIDEO (50, 33, 17)	0.57 to 0.87 Mean 0.82	0.61 to 0.84 Mean 0.78	0.17 to 0.90 Mean 0.65	0.13 to 0.82 Mean 0.57
EC (100)	0.34 to 0.99 Mean 0.78	0.25 to 0.99 Mean 0.65	0.25 to 0.99 Mean 0.61	0.13 to 0.99 Mean 0.52

Table 5.1 shows the pattern observed in Figs. 5.10 and 5.13: The higher the proportion of ANT data that is used for training and testing, the higher the AUC. When only features extracted during ANT activity are used for training (first row) the AUCs are higher than those in any of the other rows. For instance, the AUC of entry (1,1) (train/test = ANT/ANT) is statistically higher than for cells (2,1), (3,1), and (4,1). This pattern is observed to generally hold for the other entries of row 1. However, an interesting case is represented by data cell (2,2). When the training/testing scenario is ANT+EC/ANT+EC, with 87% ANT and 13% EC, the mean AUC is 0.96, which is higher than that of cell (2,1) and very close to that of data cell (1,2). This case shows that performance may be better, more robust, when using some EC data in the training dataset, 13% in this experiment, if it is known that some EC data will be used/present when testing. Another way to interpret this result is that even with up to 13% of contamination with EC data the detection results have degraded gracefully rather than abruptly. However, when EC is used for training (last row), the mean AUCs are between 0.78 and 0.52, which suggests that training models with EC data only should be avoided. When ANT, EC, and VIDEO are used for training, classification performance is not much better than that of the EC only case. Thus – for the channels and features used in this study – training with ANT data only, or as much ANT data as possible, yields the highest classification performance.

5.6.4. AR vs RC and LSF

In this section the investigation is reported of how GMM-UBMs trained with RC and LSF as features compare to those trained with AR coefficients. Based on the results from Section 4.6.4, we anticipate that GMM-UBMs trained with AR coefficients as features will yield the highest performance, followed by using RC, and then LSF features.

For these experiments, 4 subjects were used for training (2 NA and 2 A) and the other 4 for testing (1 NA and 3 A). The training dataset consisted of 35-D feature vectors of AR, RC, or LSF, coefficients depending on the case. The features were extracted from 2-s windows with 50% overlap during the ANT, VIDEO, and/or eyes closed (EC) activities. Given the available recorded data, when overlapping 2-s windows were used, for every subject 423 feature vectors were extracted during ANT; 112 during VIDEO; and 56 during EC.

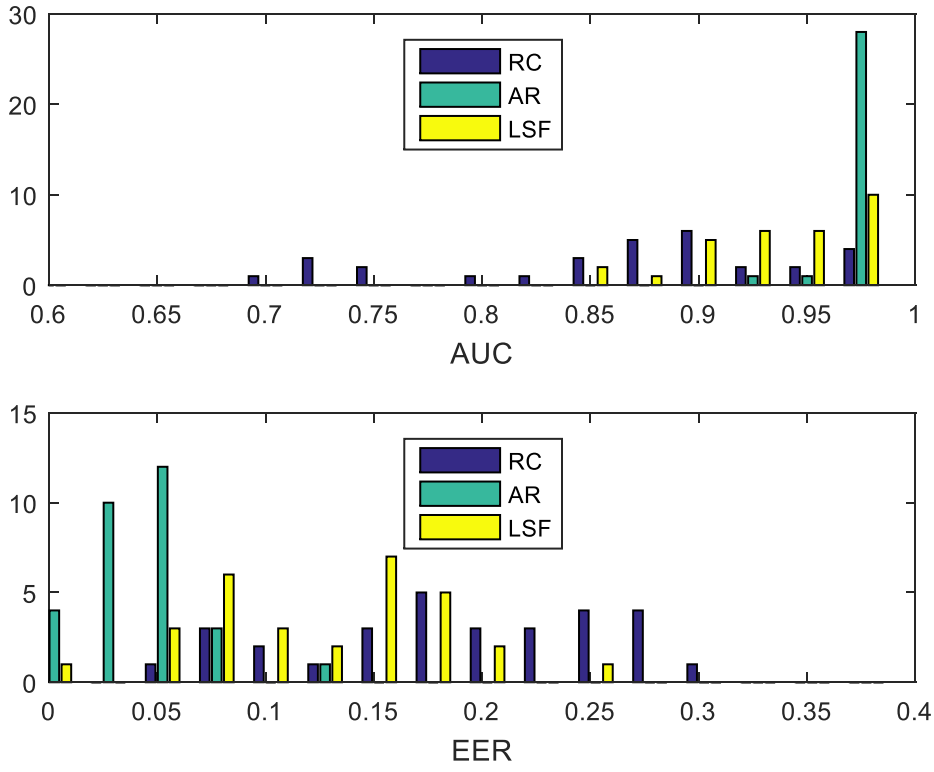

Fig. 5.14: AUCs (top) and EERs (bottom) of GMM-UBMs trained with RC (blue), AR (green), and LSF (yellow) coefficients.

Figure 5.14 shows results that are not totally in line with what was expected. As discussed earlier, when using AR coefficients, the worst EER is almost 0.135 and the worst AUC is almost 0.925. As seen, the worst EER and AUC for AR coefficients are much better than those for RC and LSF, which was expected. On the other hand, unlike for KNN, GMM-UBMs trained with LSF features appear to outperform GMM-UBMs trained with RC. For RC, the worst and mean AUC (top graph) are 0.695 and 0.865 respectively, and most of the AUC are under 0.9. For LSF, on the other hand, the worst and mean AUC are 0.8430 and 0.9365 respectively, and the AUC are highly concentrated above 0.9000. Since LSF and RC were outperformed once again by the AR coefficients, they will not be used in the next chapters.

In short, this chapter explored GMM-UBM for the classification of A and NA. The parameters of the models were carefully chosen by taking into account computational complexity and performance, and they were both optimized. The hypotheses addressed in this chapter were supported; first, KNN was the lower bound of how well classification could be done, and the

GMM-UBM approach was able to separate the data even further, achieving an AUC of 0.92 and an EER of 0.14 for the worst case. Second, training models, using the GMM-UBM approach, with EEG of data different types does make classification more robust, as long as the majority of the data is from the attention task. Contrary to what many researchers have reported, especially those supporting TBPR, this work makes the argument that eyes-closed data behaves as a contaminant, and attention data should be used instead.

6. CLASSIFICATION USING SOFT LABELS

So far, training of models has been done assuming that the subjects either have ADHD or not, with absolute confidence. In the previous chapters, the fact that there may be a level of confidence associated with the labels assigned to the subjects in our dataset was not taken into account. In the literature, when a label has a confidence level associated with it, it is called a fuzzy label or a soft label. In this chapter, the term soft label will be used. An example of the need or practicality of using soft labels is subject 32386NA, whose label was flipped to A because it seemed very distant from the NA subjects used for training in Section 4.6.4. While positive results were obtained by flipping the label, another option would have been for a clinician to provide a level of confidence describing how likely the subject seemed to be A or NA.

The latter point will be addressed here, by observing the effect that soft labeling has on classification. Since the effect of soft labels will be observed for KNN and GMM-UBM, these approaches will be referred to as Soft KNN and Soft GMM-UBM, as opposed to Hard KNN and Hard GMM-UBM, which are the terms that are going to be used in this chapter to describe KNN and GMM-UBM that use hard labels instead of soft labels.

6.1. Soft KNN

The conventional algorithm for KNN classification consists of finding the K training vectors that are closest in distance to a test vector **x**. Then, the label assigned to **x** is the most frequently occurring label of the K nearest neighbors.

KNN with soft labels, or Soft KNN, is described in detail in [65, 66]. To account for uncertainty in the labels, a membership function is created so that

$$\sum_{c=1}^{C} u_{cj} = 1 \tag{6.1}$$

where u_{cj} represents the confidence of vector j belonging to class c, and $c = 1,2,\dots,C$, where C is the total number of classes.

In our experiments, K-means clustering was used to assign the values of each u_{cj}. K-means clustering was used to cluster the data into two clusters, one for A and one for NA. Once the clusters and their respective centroids were found, the a posteriori probabilities for each vector $\mathbf{x}$ belonging to each cluster were computed as follows:

$$P_c(\mathbf{x}) = p(c \mid \mathbf{x}) = \frac{w_c g_c(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{j=1}^{c} w_j g_j(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{6.2}$$

where $P_c(\mathbf{x})$ is the a posteriori probability of vector $\mathbf{x}$ belonging to class c, w_c is the weight of the cluster, and $g_c(\mathbf{x} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ is the conditional probability that vector $\mathbf{x}$ is within a normal distribution of a vector with mean $\boldsymbol{\mu}_c$ and covariance $\boldsymbol{\Sigma}_c$.

To classify a test vector $\mathbf{x}$, the conventional algorithm for KNN classification is slightly modified. First, the K training vectors that are closest to $\mathbf{x}$ are found. Then, the vote count (confidence) of $\mathbf{x}$ belonging to any of the C classes is computed as follows [65]:

$$u_c(\mathbf{x}) = \frac{\sum_{j=1}^{K} \frac{u_{cj}}{\left\| \mathbf{x} - \mathbf{x}_j \right\|^2}}{\sum_{j=1}^{K} \frac{1}{\left\| \mathbf{x} - \mathbf{x}_j \right\|^2}} \tag{6.3}$$

where the $\mathbf{x}_j$ vectors are the K nearest neighbors. Vector $\mathbf{x}$ is assigned the label that maximizes the vote count $u_c(\mathbf{x})$. As can be seen, vote count $u_c(\mathbf{x})$, and as a consequence class membership, is affected by the weights of the soft labels u_{cj}, which are not considered in hard KNN.

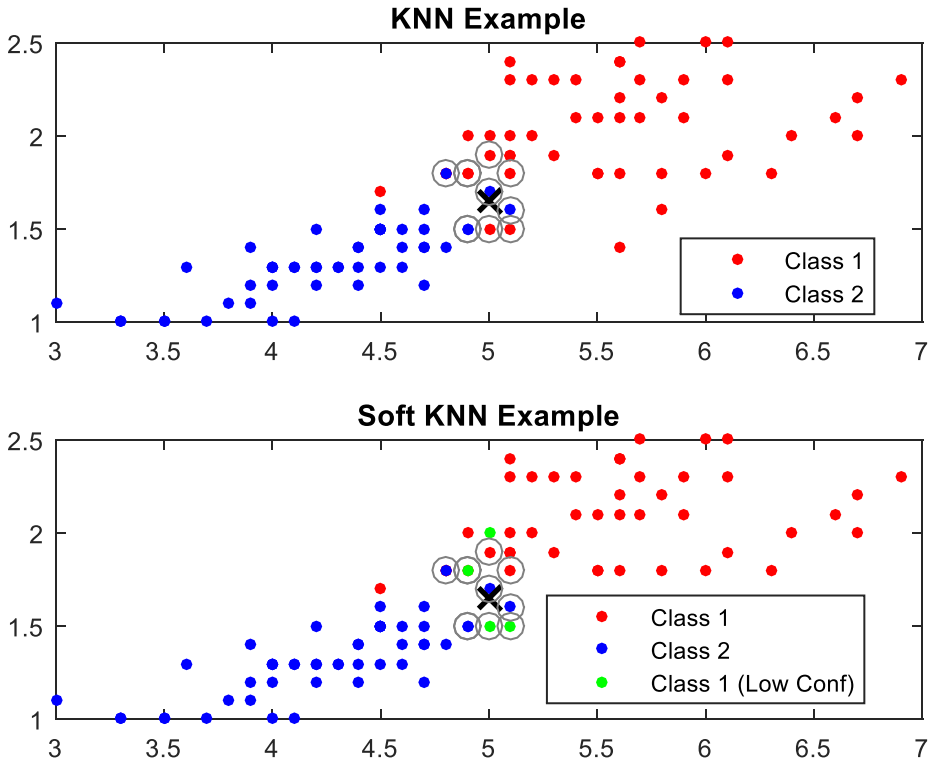

Fig. 6.1: KNN example (top) and Soft KNN example (bottom) with K = 9.

Figure 6.1 compares regular KNN with Soft KNN using an example. The top graph shows the example used for illustration in Chapter 4 (See Fig. 4.1). In this example, the test vector is labeled as "Class 1" because 5/9 neighbors belong to Class 1. The Soft KNN example (bottom), shows that this decision can be heavily influenced by the confidence of the labels. For this example, let us assume that the confidence of the green vectors is 0.5 for Class 1 and Class 2. Consequently, for the test vector shown, the votes are 3.5/9 for Class 1 and 5.5/9 for Class 2, which means that the test vector will now be labeled as Class 2 instead of Class 1. However, the confidence in the label being Class 2 is 5.5/9, i.e. much less than 1.

6.2. Soft GMM-UBM

As explained in Sections 5.1 through 5.4, the expectation maximization (EM) algorithm is used to train a GMM. In the expectation-step (E-step) of the EM algorithm, the a posteriori probabilities $\psi_{t,c,m}$ of a feature vector $\mathbf{v}_t$ belonging to the Gaussian mixture model λ_c, also known as class membership weights, are computed iteratively over variable *i* in this fashion:

$$\psi_{t,c,k} = P(w_{c,k} \mid \mathbf{v}_t, \lambda_{c,i}) = \frac{w_{c,k} g_k(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,k,i}, \boldsymbol{\Sigma}_{c,k,i})}{\sum_{m=1}^{M} w_{c,m} g_m(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,m,i}, \boldsymbol{\Sigma}_{c,m,i})} \tag{6.4}$$

where $i = 1, 2, \cdots, I$, $c = 1, 2, \cdots, C$, $m = 1, 2, \cdots, M$, and $t = 1, 2, \cdots, T$, where I is the total number of iterations, C is the total number of classes, M is the total number of mixture components, and T is the total number of feature vectors.

In the maximization step (M-step), the weights, means, and covariance matrices that parameterize the Gaussian mixture models λ_c are computed as follows:

$$w_{c,k,i+1} = \frac{1}{T} \sum_{t=1}^{T} \psi_{t,c,k} \tag{6.5}$$

$$\boldsymbol{\mu}_{c,k,i+1} = \frac{\sum_{t=1}^{T} \psi_{t,c,k} \mathbf{v}_t}{T w_{c,k,i+1}} \tag{6.6}$$

$$\boldsymbol{\Sigma}_{c,k,i+1} = \frac{\sum_{t=1}^{T} \psi_{t,c,k} \left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k,i+1}\right)\left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k,i+1}\right)^T}{T w_{c,k,i+1}} \tag{6.7}$$

With every iteration *i* the likelihood for which the parameters are computed increases, so that a maximum in the likelihood occurs at the last iteration; however, that maximum may have reached a plateau at an earlier iteration. In other words,

$$l\left(\mathbf{v}_t \mid \lambda_{c,i+1}\right) \geq l\left(\mathbf{v}_t \mid \lambda_{c,i}\right) \tag{6.8}$$

After convergence, or after a certain number of iterations is reached (15 in this study), a UBM λ_{UBM} is created by performing a MAP adaption on each GMM λ_c that will be part of the UBM.

For classification, the log-likelihood ratio (LLR) is used, i.e. the ratio of the likelihood of a test vector $\mathbf{v}_t$ belonging to the ADHD model over the likelihood of $\mathbf{v}_t$ belonging to the universal background model. If the LLR is greater than or equal to zero, the subject is classified as A, otherwise the subject is classified as NA.

$$LLR = \log\left(\frac{p(\mathbf{v}_t \mid \lambda_{ADHD})}{p(\mathbf{v}_t \mid \lambda_{UBM})}\right) \tag{6.9}$$

To account for fuzzy labels, a membership function is used. Just as for KNN, the membership function was obtained by computing the posterior probabilities of every vector belonging to class c, for $c = 1,2,\ldots,C$, which resulted in a matrix of 2x2021 vectors.

Since GMM-UBMs are parametric models, there are multiple ways to soften a GMM-UBM. The first method is by softening the weights, means, and covariance matrices of the GMMs using the membership function, as shown in (6.10)-(6.12)

$$w_{c,k,i+1} = \frac{\sum_{t=1}^{T} u_{c,t}}{\sum_{c=1}^{C}\sum_{t=1}^{T} u_{c,t}} \tag{6.10}$$

$$\mu_{c,k,i+1} = \frac{\sum_{t=1}^{T} u_{c,t}\mathbf{v}_t}{w_{c,k,i+1}\sum_{c=1}^{C}\sum_{t=1}^{T} u_{c,t}} \tag{6.11}$$

$$\mathbf{\Sigma}_{c,k,i+1} = \frac{\sum_{t=1}^{T} u_{c,t}\left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k,i+1}\right)\left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k,i+1}\right)^T}{w_{c,k,i+1}\sum_{c=1}^{C}\sum_{t=1}^{T} u_{c,t}} \tag{6.12}$$

where (6.10)-(6.12) are modified versions of (6.6)-(6.9).

When using soft GMM-UBM, all the λ_c models change, and this change is dependent on the membership function. In Fig. 6.2, Hard GMMs and Soft GMMs are illustrated. The randomly generated data shown in Fig. 6.2 are 2 Gaussians. The x and y components of one are independent and identically distributed (i.i.d.) normal distributions with $N(1, 0.75)$ and the x and y components of the other are i.i.d. normal distributions with $N(-1, 0.5)$. For the example on Soft GMMs (right), the membership function was generated from a zero-mean, unit variance Gaussian distribution normalized to the range [-1, 1] by dividing by the maximum absolute value present in the set of values. The mean of this dataset was then shifted to +1 to turn the range to [0,2]. Arbitrarily, only the first half of the resulting distribution, range [0,1], was taken. As a result, there should be a large

number of membership weights close to 1, which will cause some of the elements in one of the Gaussians to be more penalized than those of the other.

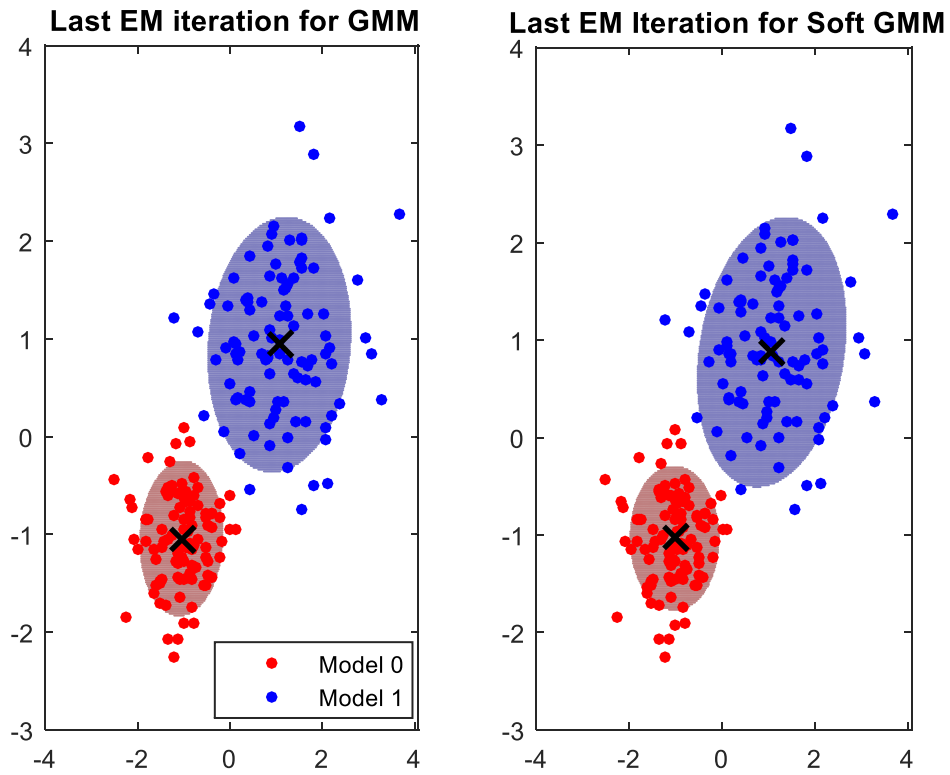

Fig. 6.2: Last EM iterations for hard GMM (left) and for soft GMM (right).

Figure 6.2 shows Soft GMMs (right) that are slightly different from Hard GMMs. Regular Model 0 does not seem very different from its softened version. However, Regular Model 1 is different from its softened version. Note that the blue blob in the right graph is slightly larger than the blue blob in the left graph, which means that the softened version of Model 1 has a mean and a variance that are different from those of the regular version of Model 1. Further, the separation between the two models for the softened GMMs is less than for the Hard GMMs. Note that the representation of the soft model depends on the data and the membership function used.

Another method to soften GMM-UBMs consists of softening the LLR by softening the probabilities $p(\mathbf{v}_t \mid \lambda_{ADHD})$ and $p(\mathbf{v}_t \mid \lambda_{UBM})$:

$$p(\mathbf{v}_t \mid \lambda_c) = \sum_{k=1}^{M} \sum_{t=1}^{T} u_{c,t} w_{c,k} g_{c,k} \left(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k} \right) \tag{6.13}$$

where $g_{c,k}$ are the pdfs of each individual Gaussian of GMM λ_c which can be expressed as

$$g_{c,k} \left(\mathbf{v}_t \mid \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k} \right) = \frac{1}{(2\pi)^{N/2} \mid \boldsymbol{\Sigma}_{c,k} \mid} \exp\left(-\frac{1}{2} \left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k} \right)^T \boldsymbol{\Sigma}_{c,k}^{-1} \left(\mathbf{v}_t - \boldsymbol{\mu}_{c,k} \right) \right) \tag{6.14}$$

Softening the LLR of GMM-UBMs has a similar effect to that of Soft KNN, where softening affects the weights of individual votes. For the case of Soft GMM-UBM, softening the LLR affects the probability of a vector $\mathbf{v}_t$ belonging to model λ_c.

The last method consists of softening both, the model parameters and LLRs. Here, not only the model is different, but also the way the decision is computed is different.

Soft GMMs and Soft GMM-UBMs have been used for speech data in the past [67, 68], but not for EEG. The methods outlined in the latter studies perform softening of the parameters and the LLRs.

6.3. Performance evaluation

For Soft KNN, performance will be measured in terms of accuracy of classification, which is defined as the number of true positives (TP) plus the number of true negatives (TN) over the total number of tests.

$$Accuracy = \frac{TP + TN}{\#tests} \tag{6.15}$$

For Soft GMM-UBM, ROC curves will be used to evaluate the performance of the models. From the ROC plots, the AUCs and EERs will be extracted, which serve to quantify how well classification was done by the model.

6.4. Experiments

In the experiments, two membership functions, u_{cj}^1 and u_{cj}^2, were used and the results were compared to those obtained when using hard labels, which will be denoted as u_{cj}^0. The experiments were conducted based on 8 subjects: 4 A and 4 NA subjects. Based on clustering and on Section

4.6.2 1 of the labels was changed from NA to A, which means that there were 5 A subjects and 3 NA subjects.

By using 4 subjects for training (2 A and 2 NA) and 4 others (3 A and 1 NA) for testing, the accuracy of classification was explored across all 30 unique permutations given the data available (5 A and 3 NA). For training and testing, AR models were computed from windows of 2 s duration, using an overlap of 50%, during the duration of ANT. Note that this approach produces many (in the range of 240 to 260) test vectors for a single test subject and for each test vector a decision is made so that a distribution of decisions results. For Hard KNN, the best classification performance resulted from using 51 nearest neighbors, as seen in Section 4.6.1. For Soft KNN, the value of K that maximizes performance will be investigated. For Hard GMM-UBM, 4 mixture components were used, and the optimal number of mixture components for Soft GMM-UBMs will be investigated as well.

6.4.1. Membership Functions

Figures 6.3 and 6.4 show the histogram of class membership weights for each individual vector of every subject using K-means clustering. The closer a weight is to 1, the higher the confidence of its label being NA. Similarly, the closer a weight is to 0, the higher the confidence of its label being A.

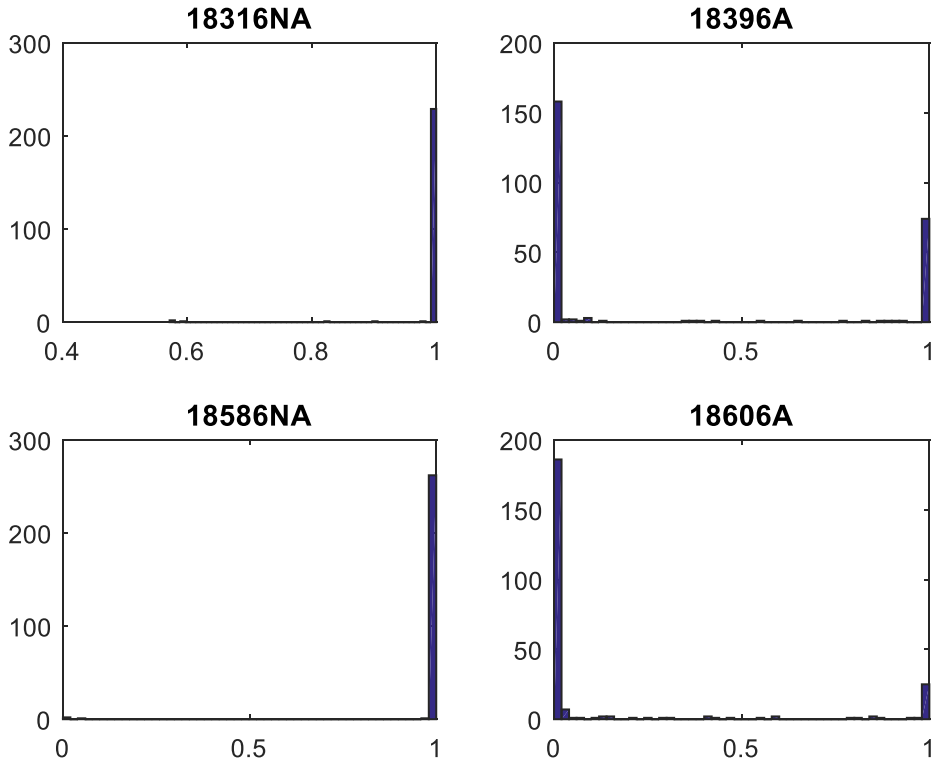

Fig. 6.3: Distribution of a posteriori probabilities of all the vectors extracted from subjects 18316NA, 18396A, 18586NA, and 18606A.

Figure 6.3 shows that the confidence of the NA labels is very high, but that of the A labels is not that high. All of the vectors extracted from subject 18316NA were clustered to the NA class, and almost all with very high confidence. Only 3/266 of its vectors were clustered to the NA class with posterior probabilities between 0.55 and 0.6. On the other hand, clustering of A subjects is not that accurate. Almost a third (82/253) of all the vectors extracted from subject 18396A were clustered incorrectly, and with very high confidence. For subject 18606A, 35/243 vectors were clustered incorrectly.

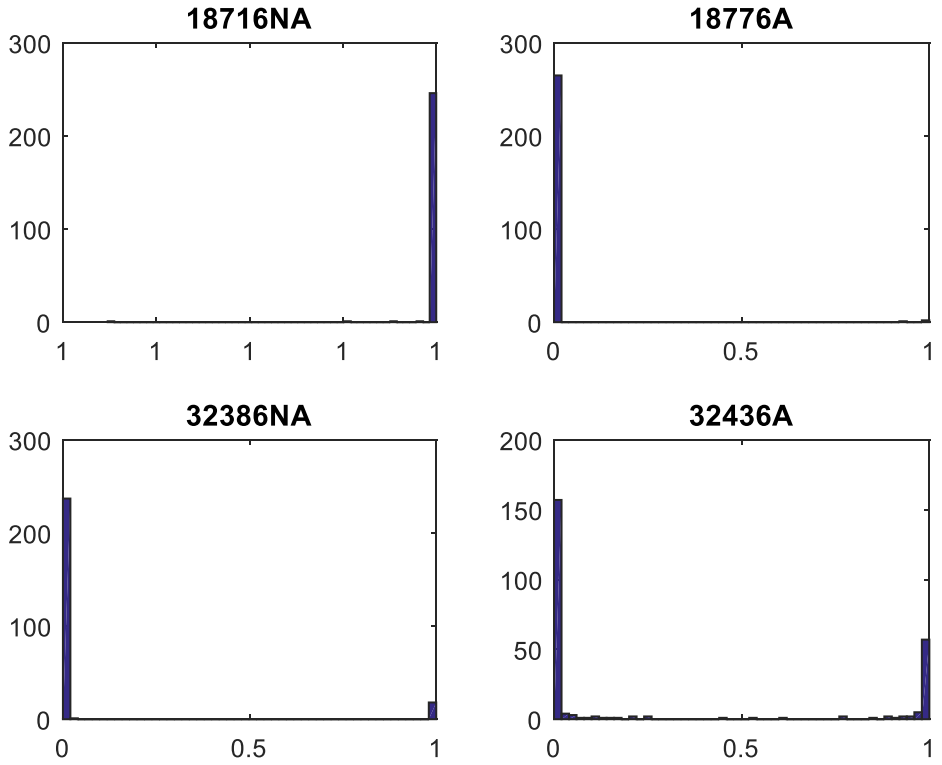

Fig. 6.4: Distribution of a posteriori probabilities of all the vectors extracted from subjects 18716NA, 18776A, 32386NA, and 32436A.

Figure 6.4 shows how the vectors from the remaining subjects were clustered. Interestingly enough, all of the vectors coming from 18716NA were clustered correctly with posterior probabilities very close to 1, if not 1. Once again, subject 32386NA appears to be an A subject, since only 18/256 vectors were clustered as NA and all the others as A. Therefore, just as in Section 4.6.2, the label of subject 32386NA will be switched to A for the next experiments. For subjects 32436A and 18776A, 74/250 and 3/268 vectors respectively were clustered incorrectly.

In summary, there were originally 751 feature vectors for the NA class and 1270 for the A class. For the NA class, 748/751 vectors were clustered correctly and with a high level of confidence. For the A class, 1068/1270 vectors were clustered correctly and 202/1270 (15.91%) were clustered incorrectly.

The second function, u_{cj}^{2}, was derived from u_{cj}^{1}. Once all the posterior probabilities were computed for every vector of every subject, for each subject *S* the mean over every vector of

posterior probabilities for subject *S* was computed (i.e. $\bar{u}^1_{0j}, \forall j \in A_S$ and $\bar{u}^1_{1j}, \forall j \in NA_S$). For all test vectors coming from the NA class, u^2_{cj} consisted of column vectors approximately equal to $[0\ 1]^T$. This occurred because 748/751 NA vectors were clustered correctly, with $u^1_{0j} \approx 0$ and $u^1_{1j} \approx 1$ for almost all values of *j*. For all of the test vectors from one A subject, u^2_{cj} consisted of column vectors approximately equal to $[1\ 0]^T$ because $u^1_{0j} \approx 1$ and $u^1_{1j} \approx 0$ for almost all values of *j*. However, for the membership vectors of the other four A subjects, the following mean posterior probabilities were obtained and used in the membership function: $[0.666\ 0.334]^T$, $[0.856\ 0.144]^T$, $[0.704\ 0.296]^T$, and $[0.929\ 0.071]^T$. Membership function u^2_{cj} was formulated to mimic a clinician providing a level of confidence in their overall diagnosis for a given subject; such a confidence level would not be more finely parsed.

The confidence values shown in Figs. 6.3 and 6.4 were used as the membership weights in the Soft KNN and Soft GMM-UBMs.

6.4.2 Setting the Value of K

To find the value of K for u^1_{cj}, an approach similar to that of Section 5.6.1 was used. Training was done with 4 subjects (2 A and 2 NA) and testing was done with 2 A and 0 NA subjects along with the membership function computed by clustering in the previous section. The features used were AR(7) extracted from windows of 2 seconds. Training and testing were done for all the odd values of K between 1 and 149. For every value of K, there were 30 combinations of subjects. The mean accuracy of classification was computed for the 30 cases of each value of K, and the results are shown in Fig. 6.5.

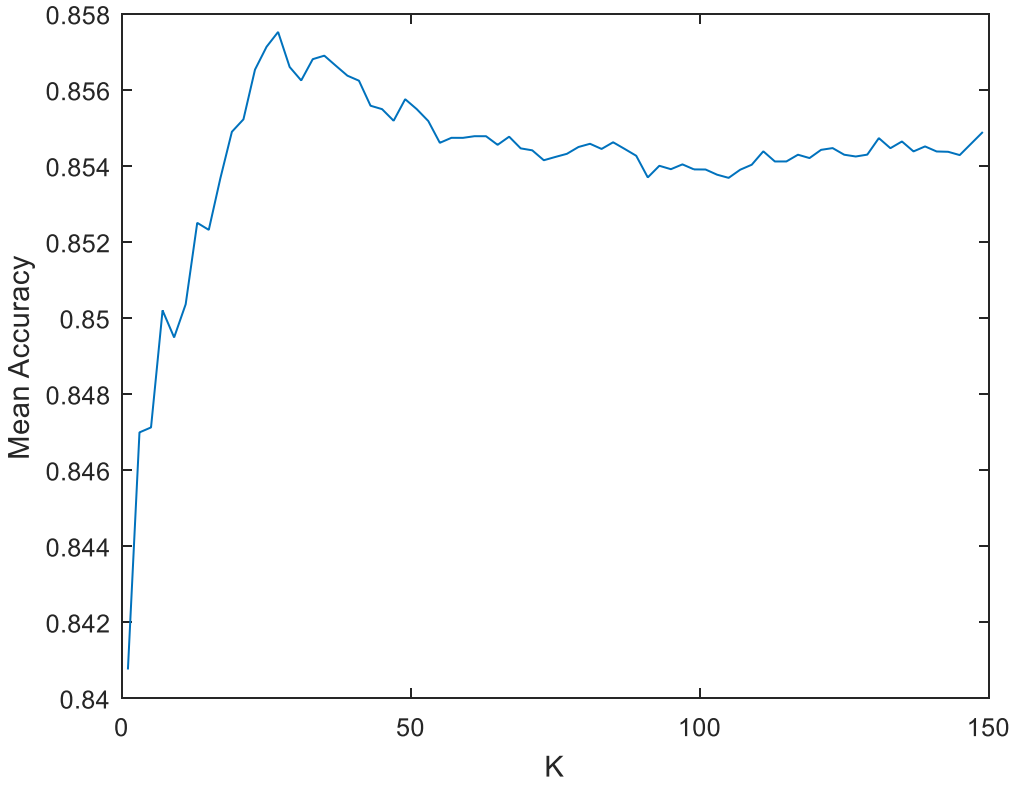

Fig. 6.5: Mean Accuracy of Classification for Different Values of K.

As seen in Fig. 6.5, the mean value of K that maximizes the mean accuracy of classification is K = 27, with a mean Accuracy of 0.8575. This value was used in the follow-on experiments. Note that for Hard KNN, K = 51, which is different from K = 27, was the value that maximized performance. This is an indication that the membership function does affect Hard KNN.

For u_{cj}^2, the value of K that maximized performance was K=3, but it was not used in the experiments. K=3 does not allow for a broad range of confidence values, and therefore, K was set to 27 as well for u_{cj}^2.

6.4.3. Soft KNN vs Hard KNN

In this section the accuracy and confidence levels for Hard KNN and Soft KNN are compared. The overall accuracy and confidence values are explored, as well the accuracy (detection rate) and confidence levels for the A and the NA classes.

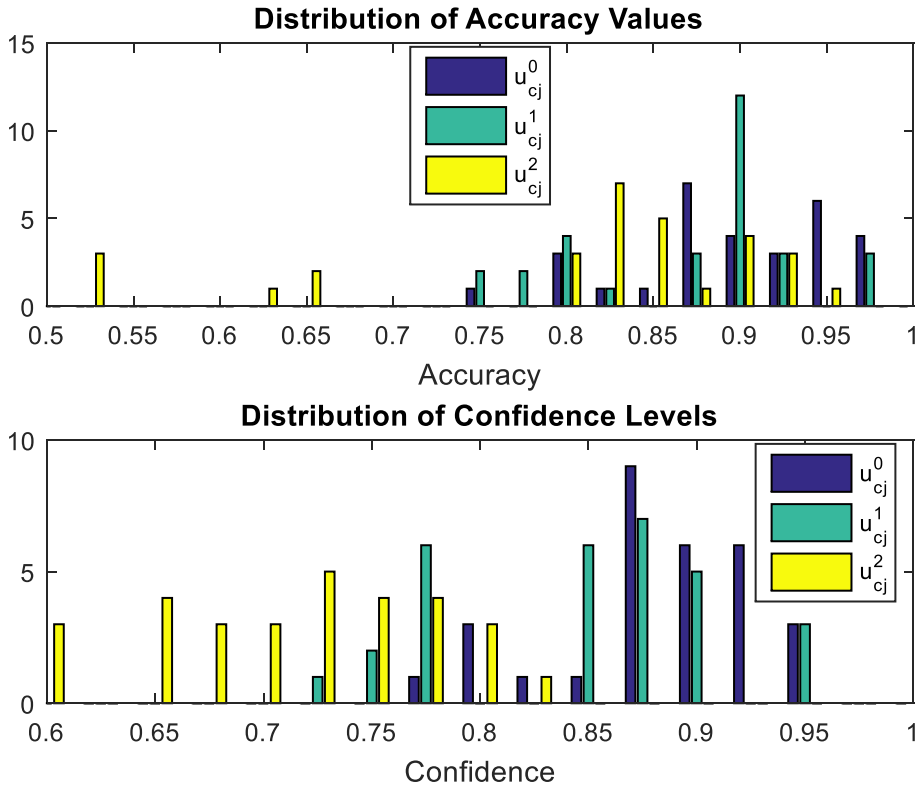

Fig. 6.6: Distribution of overall accuracy values (top) and overall confidence levels (bottom) when using Hard KNN and Soft KNN.

Figure 6.6 shows that, in terms of accuracy, u_{cj}^0 (Hard KNN) outperforms Soft KNN for the labels used. As can be seen in the top graph, using u_{cj}^1 resulted in more values below 0.9 for accuracy. For u_{cj}^1, the mean overall accuracy was 0.8730 and the minimum overall accuracy was 0.7411. For u_{cj}^0, the mean overall accuracy was 0.8963 and the minimum was 0.7452. Note that u_{cj}^0 in these experiments was trained with labels that were not only known, but also very strongly clustered to either the A class or the NA class, and the performance of u_{cj}^1 turns out to be very close to the ideal case. On the other hand, performance decreases considerably when u_{cj}^2 is used. Note that the average and worst accuracy are 0.8043 and 0.5198, where the latter can be considered a guess.

The pattern observed in the top graph can also be seen in the bottom graph. The number of lower confidence levels is higher for Soft KNN, reaching confidence of 0.7300, whereas Hard KNN reaches 0.7469. Further, the mean confidence for Soft KNN is 0.8469 whereas it is 0.8841 for Hard KNN. Lastly, for u_{cj}^2, the average and worst confidence are 0.7187 and 0.5978

A decrease in performance was expected when introducing uncertainty or softness in labeling. The hard labels used in our previous experiments (See Section 4.6.3) correspond to either one class or the other, whereas some variation is incorporated when using u_{cj}^1 or u_{cj}^2. For u_{cj}^1, performance decreased only slightly with respect to hard KNN because only 15.9% of the A vectors had soft labels. Therefore, only 15.9% of the A votes were penalized. However, u_{cj}^2 by definition penalizes, to some extent, all the votes from 4/5 of the A subjects; the confidence associated with a neighborhood comprised of training vectors with high membership is discounted.

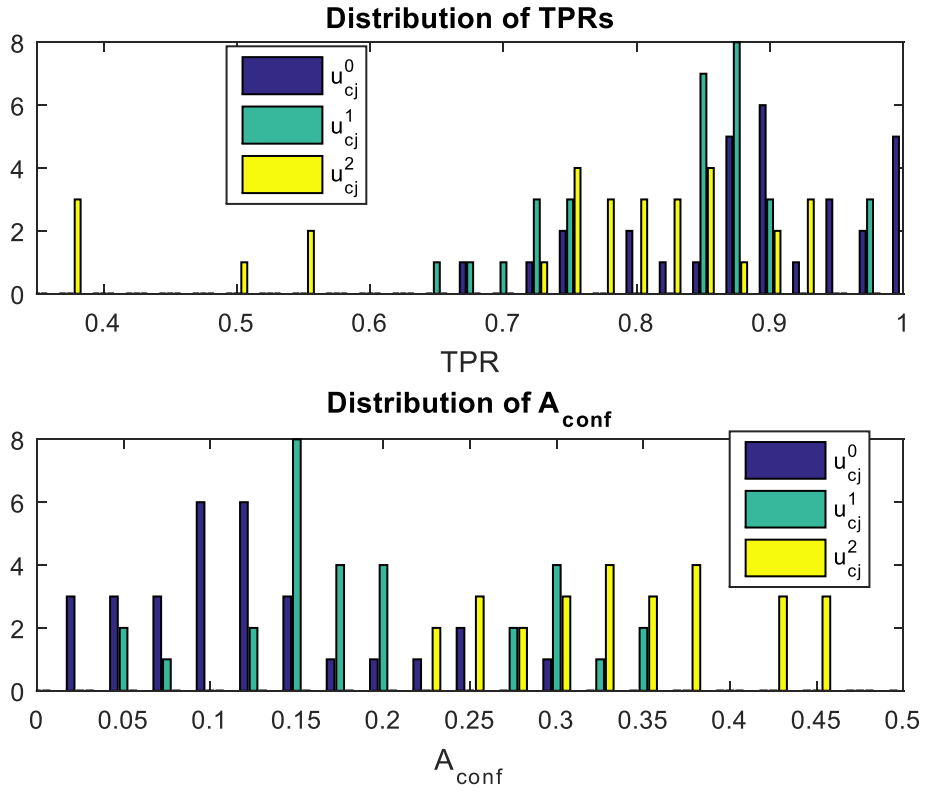

Fig. 6.7: Distribution of TPRs (top) and distribution of A_{conf} levels (bottom) when using Hard KNN and Soft KNN.

The top graph of Fig. 6.7 shows that the TPRs tend to be higher when using Hard KNN. When using u_{cj}^2, the average and worst TPRs are 0.7508 and 0.3684, which means that there are some instances where the classifier is biased towards (over-) classifying test vectors as NA. As seen previously, for u_{cj}^1 and u_{cj}^0, the results are similar. The average and worst TPR for u_{cj}^0 are 0.8877 and 0.6689. Likewise, for u_{cj}^1, the average and worst TPRs are 0.8333 and 0.6582. Thus, when using u_{cj}^1, TPRs tend to be 5% lower than when using hard labels.

The bottom graph exhibits a behavior similar to that of the top graph. For Hard KNN, most of the confidence levels are below 0.1000, 0 being the value that indicates highest confidence. However, for u_{cj}^1, there is a larger group of values above the 0.1 level of confidence. For u_{cj}^0, the average and worst confidence levels are 0.1224 and 0.2959 respectively, whereas for u_{cj}^1 they are 0.1949 and 0.3468 respectively. For u_{cj}^2, the confidence levels are always above 0.2, and the worst case has a confidence of 0.4783, which can be considered guessing.

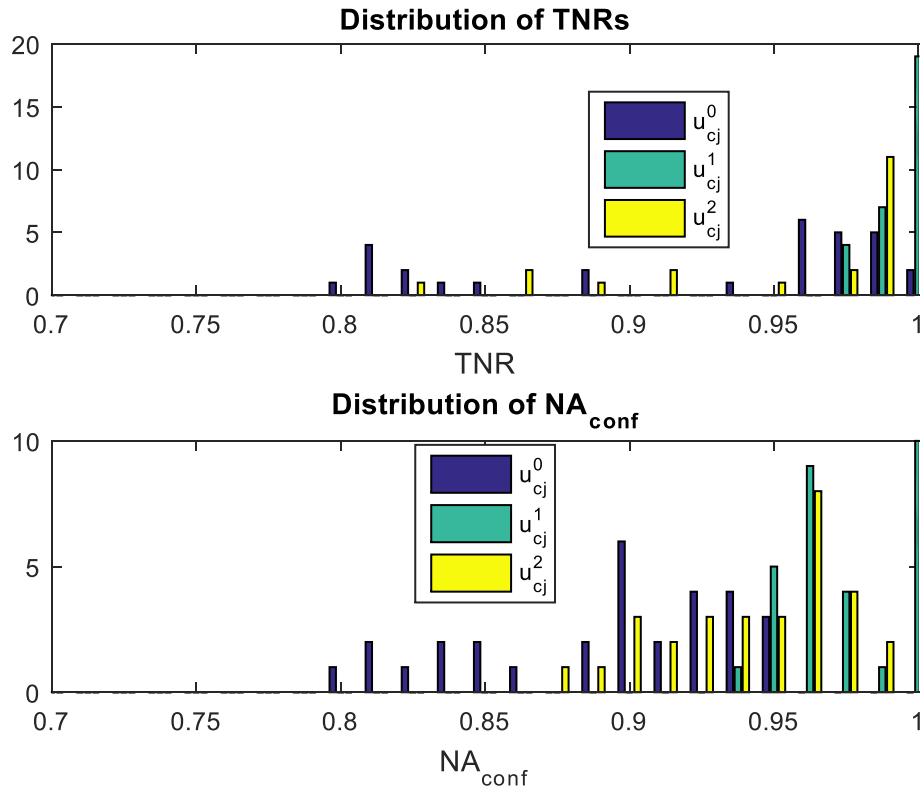

Fig. 6.8: Histogram of TNRs (top) and distribution NA_{conf} (bottom) when using KNN and Soft KNN.

Figure 6.8 shows interesting results. For u_{cj}^1, the average and worst TNRs are 0.9934 and 0.9787, whereas for u_{cj}^2 and u_{cj}^0 the average and worst are 0.9671 and 0.8280, and 0.9231 and 0.8008. Hence, when using u_{cj}^1, TNRs tend to be 7% higher than for hard labels.

The bottom graph of Fig. 6.8 shows high confidence levels for all cases. For u_{cj}^0, the mean confidence is 0.8936 and the worst confidence level is 0.7980. Unsurprisingly, the worst confidence level obtained when using u_{cj}^1 is 0.9394, and the mean is 0.9738. For u_{cj}^2, the average and worst confidence levels are 0.9482 and 0.8774. This information confirms that the membership functions introduced a bias.

The clustering performed in Section 6.4.1 provides a hint as to why this bias occurs. The histograms of Section 6.4.1 show that most of the test vectors coming from NA subjects are clustered to the NA class with a high level of confidence (i.e. weights close to 1), whereas 84% of

the test vectors coming from the A class were clustered correctly. What this means, for the Hard KNN classifier, is that a negligible number of training or testing vectors from the NA class will be surrounded by A training vectors, which implies that a small number of test vectors from the NA class will be misclassified. On the other hand, a portion of A training vectors, up to a third for the case of 18396A, will be surrounded by a large concentration of NA training vectors, which may negatively impact the accuracy of the classifier.

For Soft KNN, the vectors that are clustered incorrectly lose significance when it comes to classification. Since votes are dependent on the confidence of the labels, there is a considerable number of A vectors whose votes will favor the NA class instead of the A class because the confidence in their labels is low, whereas almost all of the votes coming from NA vectors favor only the NA class and not the A class.

6.4.4. Number of Mixture Components

The effects of softening in GMM-UBMs is assessed in this section. The following scenarios will be explored: Hard GMM-UBM (Hard), softening of the parameters (weights, means, and covariance matrices) (SP), softening of the log-likelihood ratios (LLR) (SL), and softening of the parameters and the log-likelihood ratios (SPL) for both of the hypothesis functions. Our hypothesis is that softening will decrease apparent performance.

Before creating Soft GMM-UBMs, the number of mixture components was optimized for all of the 3 scenarios of each membership functions. This was done because it is not guaranteed the optimal number of mixture components will be 4 for all those cases. To obtain the optimal number of mixture components, the approach explained in Section 5.6.1 was used. Figs. 6.9 and 6.10 summarize the results.

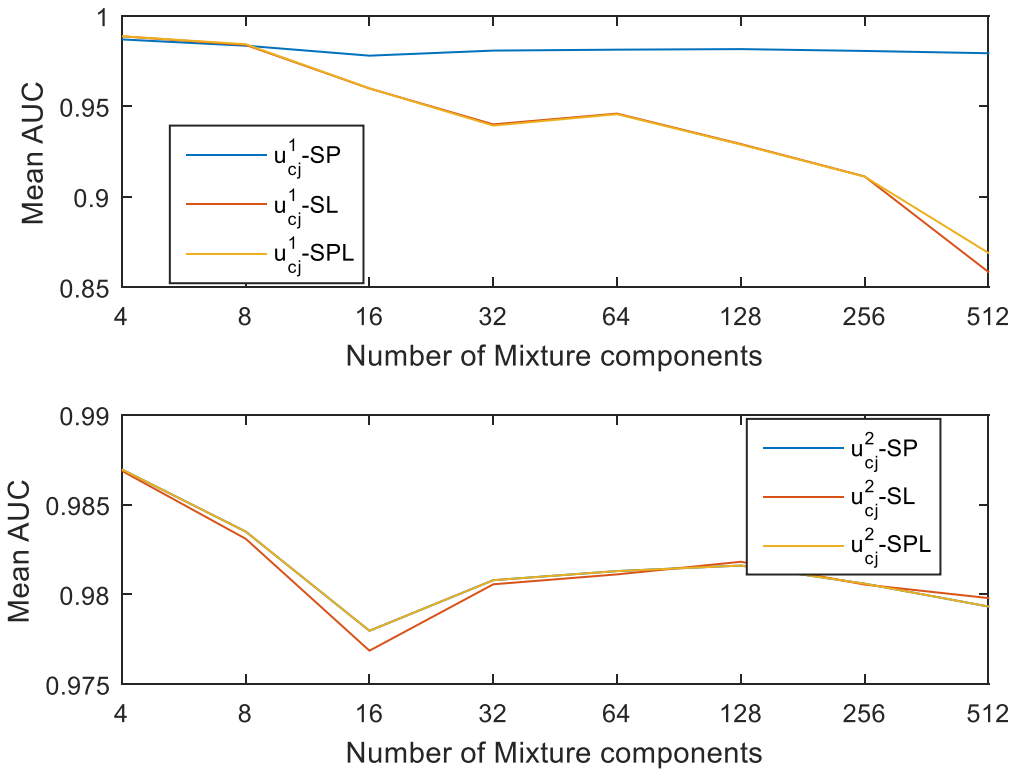

Fig. 6.9: Mean AUC for all softening scenarios involving u_{cj}^1 (top) and u_{cj}^2 (bottom).

Figure 6.9 summarizes the process for the selection of the optimal number of mixture components. For the top and bottom graphs, results for 1 and 2 mixture components are not shown because they are much lower (~ 0.7000) than the ones shown in the graphs. For u_{cj}^1 (top graph), under all three different scenarios, the number of mixture components that maximizes AUC is 4; SL and SPL are tied in first place, with mean AUC equal to 0.9880, and SP is in third place, with mean AUC of 0.9868. Observe in Fig. 3 that the AUC of SP oscillates as the number of mixture components increases, but it decreases for SL and SPL as the number of mixture components increases. The mean AUC for SL and SPL are very close for all mixture components between 4 and 256, but after the number of mixture components reaches 256, the mean AUC of SL decays faster, reaching almost 0.87. It is only at this point that it becomes visible that AUCs obtained when using SPL are larger than those obtained when using SL.

Observe that the bottom graph (illustrating the three scenarios for u_{cj}^2) exhibits a behavior similar to that of top graph (illustrating the three scenarios for u_{cj}^1). The optimal number of mixture

components appears to be 4 as well, but now SP and SPL are tied in first place, with mean AUC of 0.9870, and SL in third place with AUC of 0.9869. In fact, the graph shows a green line because the blue and yellow lines are almost coincidental. When more than 16 mixture components are used, the mean AUC oscillates for these three cases, and the global minimum appears to be when the number of mixture components is 16.

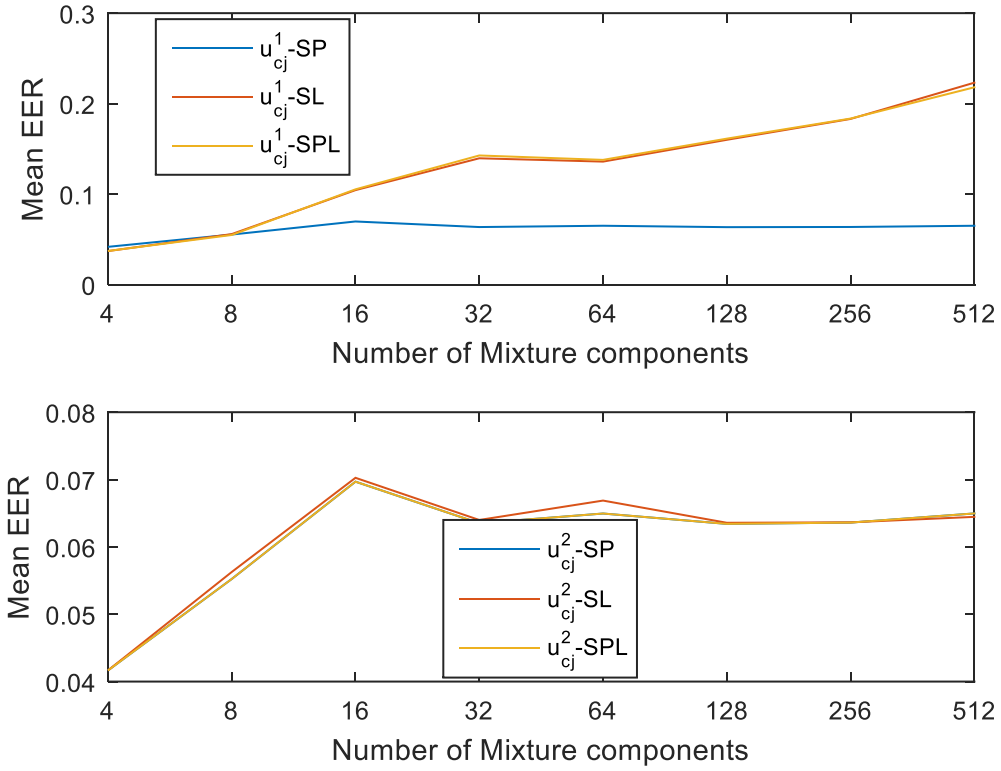

Fig. 6.10: Mean EER for all softening scenarios involving u_{cj}^1 (top) and u_{cj}^2 (bottom).

Figure 6.10 confirms that the optimal number of mixture components is 4 for both membership functions. Just as in the top graph of Fig. 6.9, SL and SPL are tied, with EER of 0.0371, and SP is last, with EER of 0.0417. For the latter, the mean EER oscillates after this point, but for SL and SPL, the mean EER degrades as the number of mixture components increases. For u_{cj}^2 (bottom graph), the mean EER for SP, SL, and SPL was 0.0418. Just as in Fig. 6.9, the EER values obtained when using SP and SPL are the same for most cases and very close for some others.

6.4.5. Soft GMM-UBM vs GMM-UBM

The follow-on experiments were conducted using 2 A and 2 NA for training and 2 A and 2 NA for testing, which led to 30 different distinct permutations. Figs. 6.11 and 6.12 show the distribution of AUCs and EERs that resulted from these combinations. Since Figs. 6.9 and 6.10 displayed values that were very similar, especially for u_{cj}^{2}, we anticipate that the distributions will not vary too much for the number of mixture components chosen.

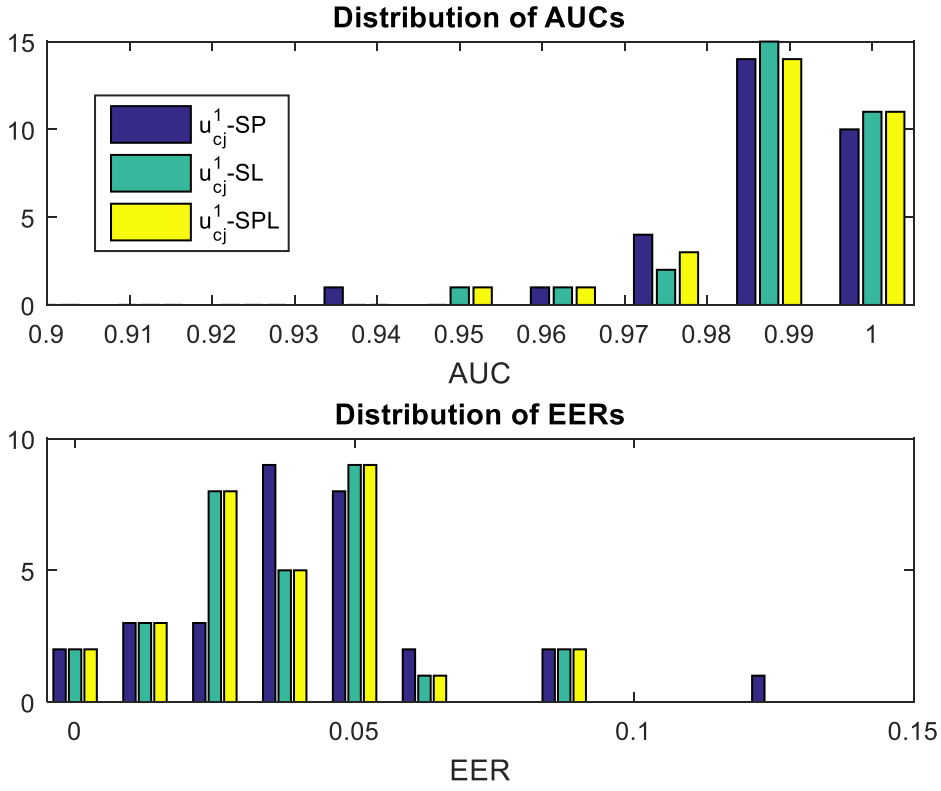

Fig. 6.11: Distribution of AUCs (top) and EERs (bottom) for all softening scenarios using u_{cj}^{1}.

Figure 6.11 shows encouraging results for all scenarios of u_{cj}^{1}. For all scenarios, the AUCs (top histogram) are highly concentrated above 0.98. In terms of AUC, the softening scheme that resulted in lowest performance is SP, with AUC of 0.987 for the average case and AUC of 0.9382 for the worst case. Note that the worst case appears to be an outlier, since the next minimum is 0.9636.

Surprisingly, the distribution of AUCs for SL and SPL are very similar. For SL 15/30 cases yielded AUCs of approximately 0.9900, whereas 14/30 yielded AUCs of approximately 0.9900

for SPL. Further, 2/30 cases yielded AUCs of approximately 0.9750 for SL, whereas 3/30 cases yielded AUCs of approximately 0.9750 for SPL. These are the only visible differences that can be obtained from the top graph. For the worst case, SP and SPL have AUC of 0.9554. For the average case, SPL has an AUC of 0.9888, and SL has an AUC of 0.9887. Although this difference is negligible, it indicates that performance is higher when SPL is used than when SL is used.

The distribution of EERs (bottom histogram) serves to confirm what was observed in the AUCs. When SP is used, EERs for the average and worst case are 0.0416 and 0.1271 respectively. The worst case happens to coincide with the one where the lowest AUC was observed in the top histogram. For SL and SPL, the distributions appear to be identical, but they are slightly different. When SPL is used, the average and worst EER are 0.0371 and 0.0884 respectively, whereas the average and worst EER for SL are 0.0369 and 0.0886. Although the difference may be negligible, this shows that the performance in the worst case is higher when SPL is used than when SL is used. On the other hand, performance in the average case is higher for SL than for SPL.

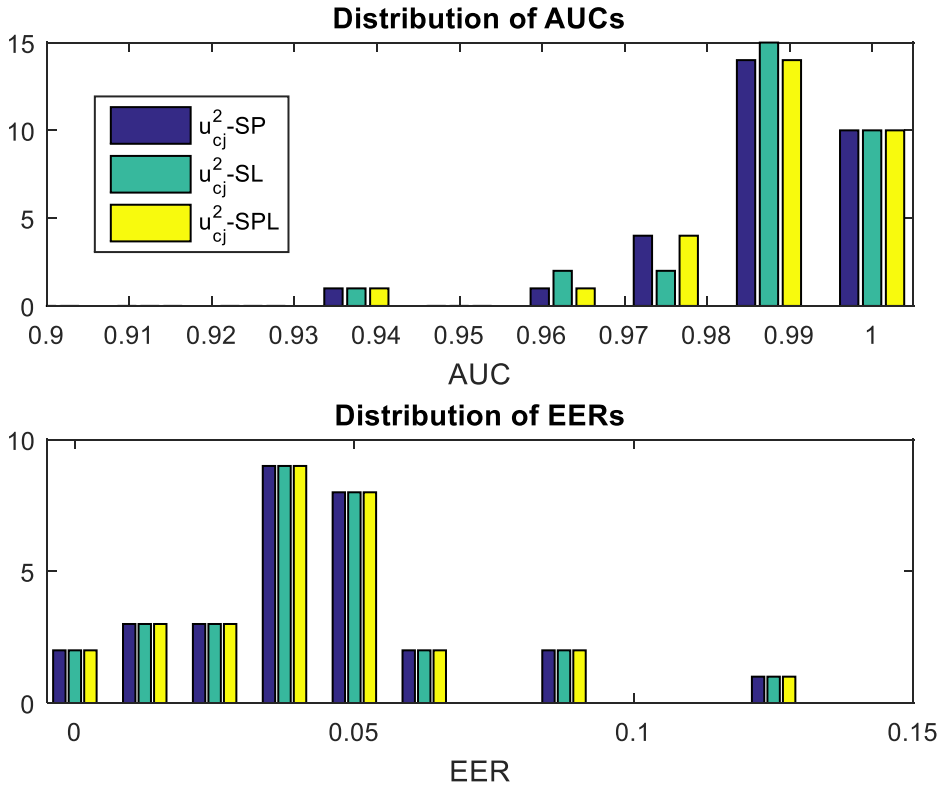

Fig. 6.12: Distribution of AUCs (top) and EERs (bottom) for all softening scenarios using u_{cj}^2.

Figure 6.12 shows encouraging results. The distribution of AUCs (top histogram) for all scenarios, when using u_{cj}^2, are very similar. In fact, SP and SPL seem to have identical distributions, and all of distributions seem to be concentrated above 0.98 AUC, with an outlier close to 0.9375. For SL, the distribution is slightly different, with AUC for the average and worst cases of 0.9869 and 0.9372 respectively. For SP, the AUCs for the average and worst cases are 0.9870 and 0.9382. Similarly, the AUCs of the average and worst cases, when using SPL, were 0.9871 and 0.9380. As for the EERs (bottom histogram), the distributions appear to be identical across all three scenarios. For the worst case, EER is 0.1271 for the 3 scenarios. The average EER is 0.0417 for SL and 0.0416 for SP and SPL.

As expected, performance did not vary too much for u_{cj}^1 and u_{cj}^2 for the 3 scenarios. For u_{cj}^2, all the results were impressive. Although the method that returned the lowest mean and worst AUC was SL, the difference in AUCs was not enough to discard SL. For u_{cj}^1, when SP is used, performance detracts slightly, but performance values are approximately equal when using $u_{cj}^1 - SL$ and $u_{cj}^1 - SPL$

Certain patterns can be found in the previous experiments. For u_{cj}^2, AUC is maximized when the parameters are softened (SP and SPL). For u_{cj}^1, the results are slightly more intriguing. Performance is lowest for SP, whose AUC for the worst case is close the AUC for the worst case of $u_{cj}^2 - SP$, $u_{cj}^2 - SL$, and $u_{cj}^2 - SPL$. However, when the decision rule is softened (SL and SPL) and u_{cj}^1, the worst AUCs and EERs are shifted, and consequently, the average case is moved closer to the ideal value (1 for AUC, 0 for EER).

Performance does not vary enough to prefer one method over the other, but some suggestions can be made. For u_{cj}^1, there is utility in discarding SP and using SL or SPL. Although performance did not vary by much between these two, SPL appears to be more tolerant to inadequate numbers of mixture components than SL (See Figs. 6.9 and 6.10). For u_{cj}^2, making a recommendation is more difficult: variations in performance metrics are in the order of 10^{-4}, and performance of the outliers was not increased. Also, if the number of mixture components is overestimated, there is a

point where the use of SL maximizes AUC and minimizes EER by a little, but before this point SP and SPL minimize EER.

In Fig. 6.13, the distribution of AUCs and EERs of u_{cj}^{0} (Hard GMM-UBM), $u_{cj}^{1} - SPL$, and $u_{cj}^{2} - SPL$ are compared. Other cases, such as $u_{cj}^{2} - SP$ or $u_{cj}^{1} - SL$ could have been used, but the difference in their distributions was negligible.

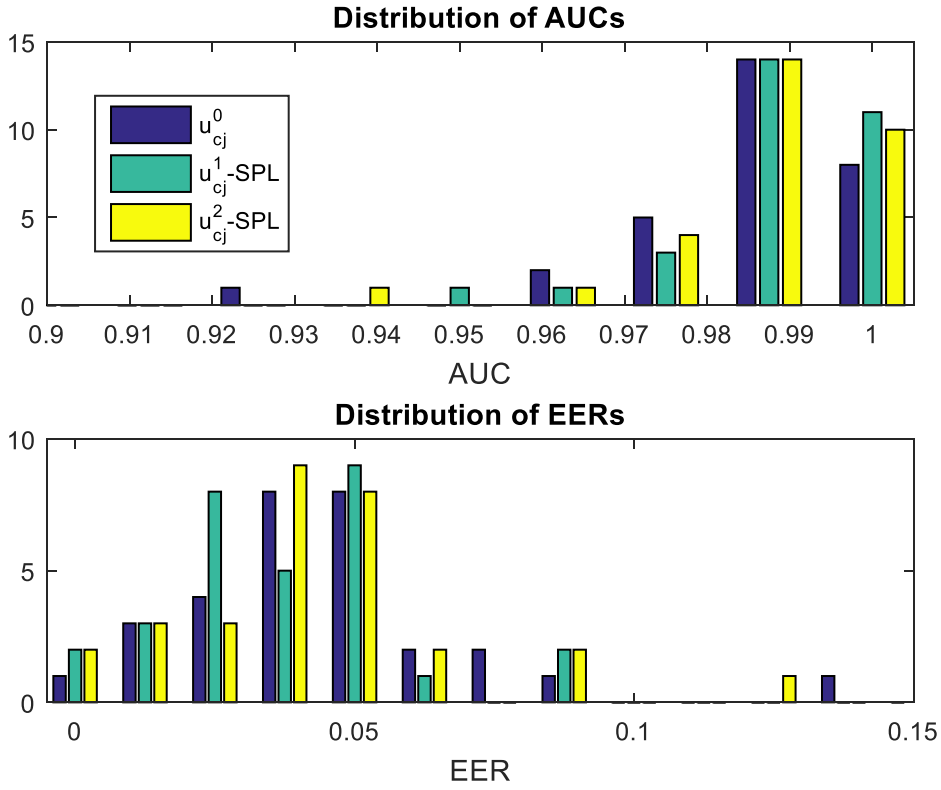

Fig. 6.13: Comparison of u_{cj}^{0}, $u_{cj}^{1} - SPL$, and $u_{cj}^{2} - SPL$ in terms of AUCs (top) and EERs (bottom).

Observe in Fig. 6.13 that using soft labels increases performance. For AUCs and EERs, u_{cj}^{0} minimizes performance of the average and worst cases. In Section 5.6.2, it was reported that Hard GMM-UBM classified A and NA subjects with AUC of 0.9858 and 0.9246 for the average and worst cases. It also reported EER of 0.0486 and 0.1336 for the average and worst cases. On the other hand, the methods presented here increase performance of both metrics for the average and worst cases. When using u_{cj}^{2}, for SP and SPL, AUC for the average and worst case is 0.9382 and 0.9870, which represent a slight improvement. As for the EERs, which represent the miss rate, the

average and worst cases when u_{cj}^2 is used are 0.0417 and 0.1271, which is another mild improvement. Lastly, when $u_{cj}^1 - SPL$, AUC for the average case does not increase by much, but for the worst case, it becomes 0.9554. Moreover, EER for the worst case of $u_{cj}^1 - SPL$ is 0.0884, which is approximately 40% smaller than 0.1336, reported for Hard GMM-UBM.

Figure 6.14 shows Detection Error Trade-off (DET) curves for the average and worst cases when u_{cj}^0, $u_{cj}^1 - SPL$, and $u_{cj}^2 - SPL$ are used. The axes of DET plots usually go from 1 to 50%, but they are zoomed in to the 1-20% region to show more detail. These plots serve to illustrates the EERs, which are found at the intercepts of the 45 degree line (red) and the curves. DET curves contain the same information as ROC plots, but ROC plots are not in log scale. Further, ROC plots illustrate AUCs, which seem to vary very little for the methods evaluated in this paper.

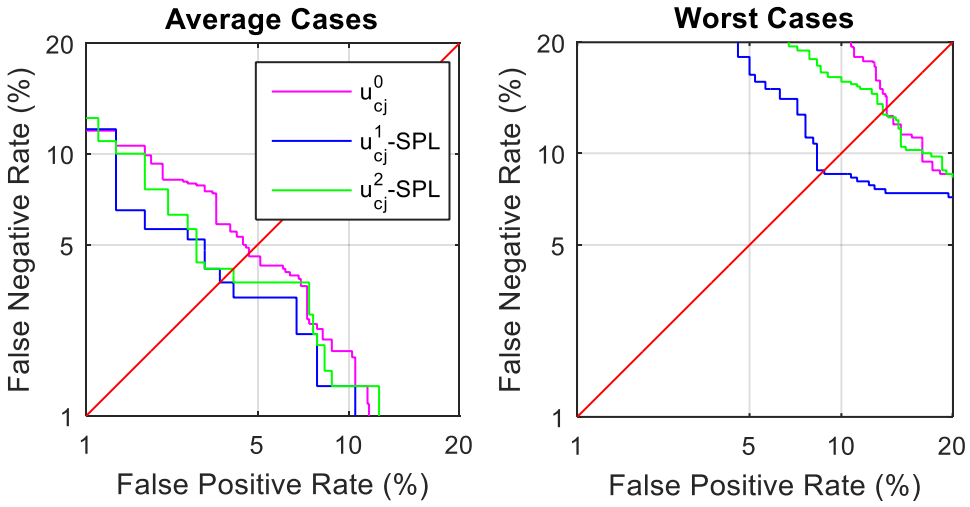

Fig. 6.14: Comparison of DET curves for the average (left) and worst (right) cases.

Figure 6.14 confirms the observations made for the EERs of these three cases (See Fig. 6.13). For the average cases (left), there is very little difference between $u_{cj}^1 - SPL$ (blue) and $u_{cj}^2 - SPL$ (green). They intersect with the 45 degree line at points that are very close. Both the green and blue curves are not smooth, even though approximately 1000 vectors were used for testing, which indicates that very few errors are made. When u_{cj}^0 is used, the intercept between the magenta curve and the red line is higher than that for the other two cases, which indicates higher number of errors.

In addition, the magenta curve is smoother, than the other 2 curves, which also indicates more errors.

For the worst cases (right) the differences are more noticeable. As discussed earlier, for the worst cases, EERs obtained when using u_{cj}^{0} and $u_{cj}^{2} - SPL$ vary by very little. Observe that u_{cj}^{0} and $u_{cj}^{2} - SPL$ intersect with the red line almost at same point. However, there seems to be a benefit of using $u_{cj}^{1} - SPL$. The blue curve intersects with the red line at almost 9%, and the curve is less smooth than the other 2, which indicates fewer misses.

As can be seen, there is utility in soft labeling and using soft labels for the classification of A and NA. Performance does not increase by much for the average case, but it does for the worst case. To maximize performance for the worst case, labeling of every 2-s time interval appears to be necessary. Nevertheless, increases in performance can be obtained by providing an overall level of confidence for a given subject, which is something that clinicians may be able to do. These results are more encouraging than those for Soft KNN (See Section 6.4.3). The results suggested that membership functions similar to u_{cj}^{2} may be detrimental to performance because they penalize one class by too much and the other by very little, which causes classification biases.

This chapter explored something that has not been attempted before: Classification of A and NA based on soft labels. For KNN, the method consisted of penalizing the votes of low certainty labels, which resulted in performance that was similar to that of the original KNN. For GMM-UBMs, it was found that the worst case improved when the parameters of the GMM-UBM were softened, but the mean performance values decreased. For this case, the effect of softening the likelihood ratio was also studied, but performance decreased whenever the likelihood ratio was found. In short, if correct, high-confidence labels are not available, softening of the algorithms can help approximate the ideal results. For KNN, softening the voting system seems adequate, and for GMM-UBM, softening the parameters only is recommended.

7. OPTIMAL CHANNEL REDUCTION

Motivated by the encouraging results described in the previous chapters, the task at hand is to explore whether there is a better combination of channels, perhaps of fewer than 5 channels, that will further maximize discrimination of A and NA subjects while also reducing computational cost. The latter is a consideration towards eventual development of an efficient portable prototype for point of care diagnostic purposes.

Since the optimal choice of channels may be different depending on the classification scheme, channel reduction will be explored for the KNN and GMM-UBM methods. AR coefficients are used as features. Datasets from 4 subjects (2 NA and 2 A) were used for training, while datasets from 4 other subjects (1 NA and 3 A) were used for testing, which resulted in 30 different combinations, as in the other chapters. The performance associated with each selection of channels is analyzed in terms of classification accuracy for KNN, as described in Section 4.5, and in terms of AUC and EER for GMM-UBM, as described in Section 5.5.

7.1.Channel Ranking

For KNN, ranking was done as follows: The EEG data from all possible 2-channel combinations was used for feature extraction. KNN classifiers were trained and tested, as explained in the previous paragraph, and the combination that maximized mean classification accuracy (across all test vectors of every case) was chosen as the best combination for the number of channels being investigated. To find a 3rd channel, all possible 3-channel combinations that include the best 2 were used in feature extraction and then used for training and classification. This process was repeated to find the fourth channel.

For GMM-UBM, the same process was executed, with the difference that the optimization objective was to maximize AUC and minimize EER.

7.2. Experiments

Unlike in the earlier chapters, feature vectors of 35-D were not used. For every test channel, AR(7) parameters were computed, and feature vectors were formed as the concatenation of the AR coefficients of all test channels. Thus, when searching for the best combination of 2 channels, 14-

D feature vectors were obtained; when searching for the best third channel to add to the best 2-channel combination, 21-D feature vectors were used; when searching for a fourth channel to add to the latter combination, 28-D feature vectors were used.

Just as in the previous chapters, training was done with 4 subjects (2 A and 2 NA) and testing was done with the other 4 (3 A and 1 NA). Consistent with the previous chapters, the KNN algorithm was performed for k=51 and 4 mixture components were used for GMM-UBM.

7.2.1. Best Channel Combinations for KNN

For this channel ranking scheme, the accuracy of classification with all 2-channel combinations was investigated first. Once the combination that maximized mean accuracy (across all test vectors, of all test subjects, for all 30 combinations) was found, the investigation focused next on which channel(s) could be added thereto to obtain the best 3- and 4-channel combinations.

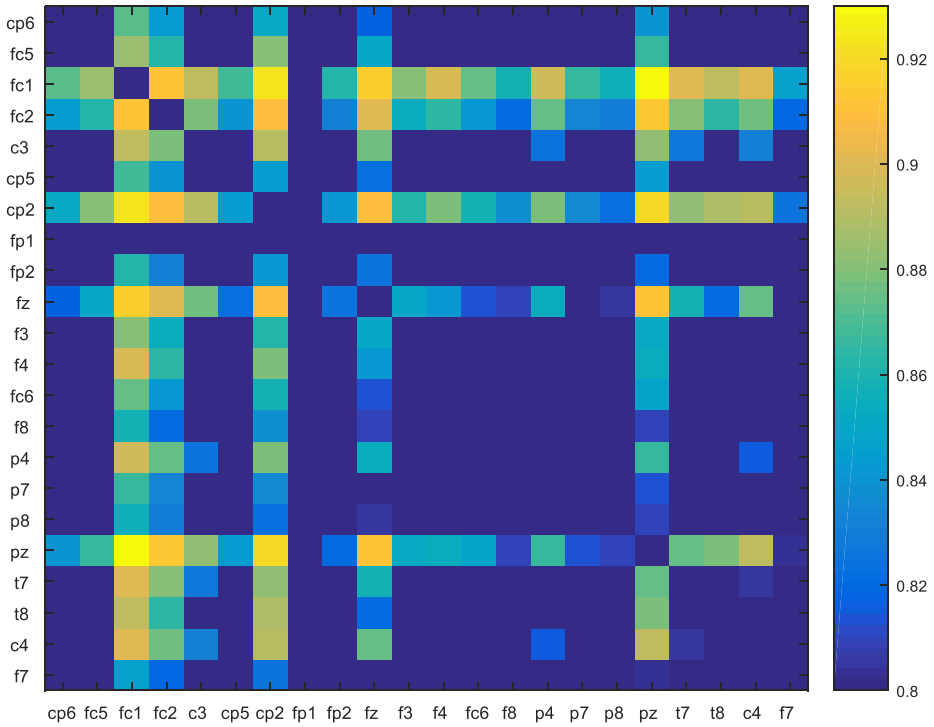

Fig. 7.1: Mean accuracy for all 2-channel combinations.

Figure 7.1 shows the accuracy of classification for all 2-channel combinations, for visual differentiability zoomed in to the range from 0.8 to 0.93 with color. As a result, in Fig. 7.1 many entries are blue or dark blue, meaning that the accuracy of classification was below 0.86. In this

experiment, the best classification performance was realized when pair Fc1-Pz was used, which achieved a classification accuracy CDF (cumulative distribution function) characterization of {0.8933 - 0.9312 - 0.9637}, which are the 5th percentile, the mean, and the 95th percentile respectively. The latter percentiles provide an idea of how concentrated the accuracy (viewed as a random variable) is about the mean. The Fc1-Pz pair is followed by Fc1-Cp2, with classification performance of {0.8646 - 0.9238 - 0.9645}. Other combinations, such as Cp2-Pz, Fc1-Fz, and Fc2-Pz ranked high as well (third, fourth, and fifth respectively), but achieved mean accuracy of less than 0.92. Since Fc1 and Pz yielded the highest accuracy of classification, the Fc1-Pz pair was used in the follow-on experiments.

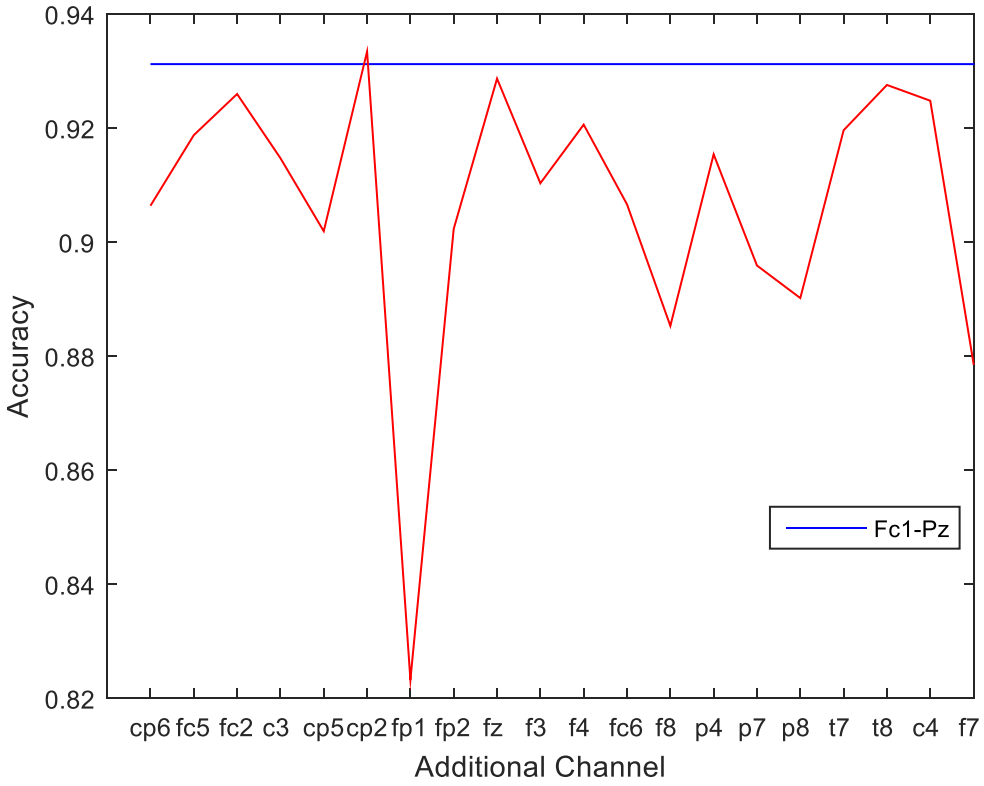

Fig. 7.2: Accuracy of 3-channel combinations that include Fc1-Pz.

Figure 7.2 shows the top 3-channel combinations that include Fc1-Pz. As can be seen, the combination that yields the highest accuracy is Fc1-Pz-Cp2, which was then used in further experiments. For Fc1-Pz-Cp2 the CDF values of classification performance are {0.8988 – 0.9334 - 0.9690}.

The second best performance results from the selection Fc1-Pz-Fz, with accuracy of performance CDF values of {0.8831 - 0.9286 - 0.9581}. Figure 7.2 shows interesting results because Fc1-Cp2 ranked second in the 2-channel experiment, and Pz, Fz, and Fc2 show up in the third, fourth, and fifth best 2-channel combinations. It is also worth noting that mean accuracy does not increase by much when adding Cp2: an increase from 0.9312 to 0.9334.

Figure 7.3 summarizes the 4-channel combinations that include Fc1-Pz-Cp2.

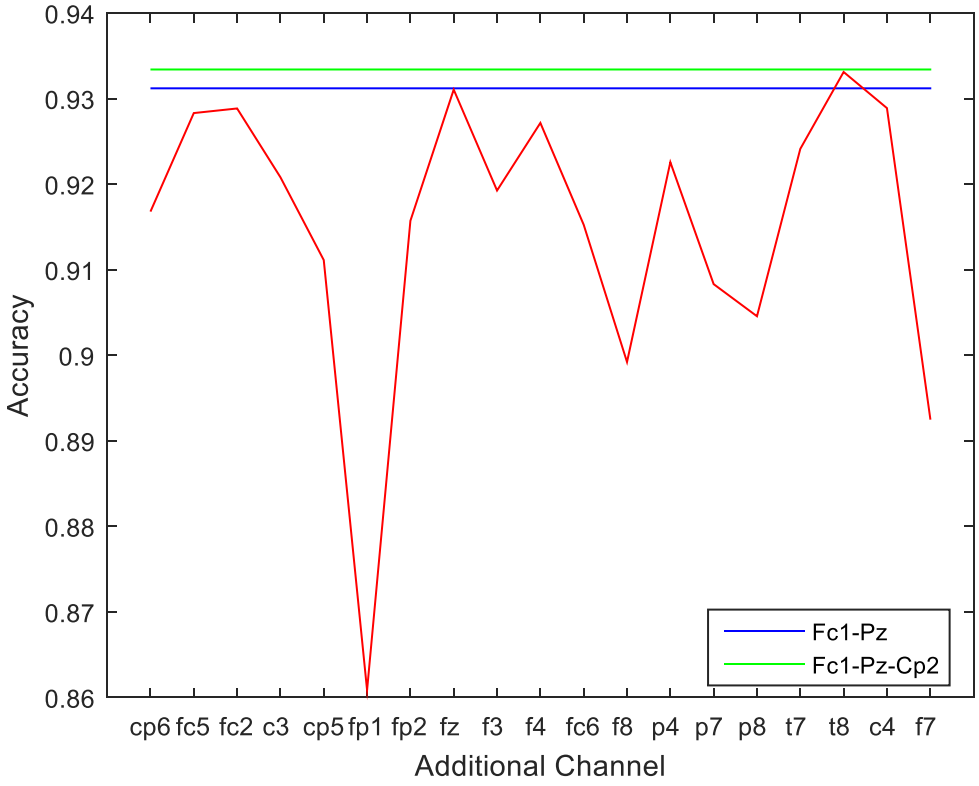

Fig. 7.3: Accuracy of all 4-channel combinations that include Fc1-Pz-Cp2.

As seen, the combination that yields the highest 4-channel performance is Fc1-Pz-Cp2-T8, with classification CDF characteristics of {0.8996 - 0.9331 - 0.9775}. Note that while this is the best 4-channel combination, accuracy is about the same (slightly higher 5^{th} and 95^{th} percentile, but slightly lower mean) as for its 3-channel counterpart, which suggests that a 3-channel combination would be more favorable in the feature domain used in this study. However, if computational complexity is to be minimized, the Fc1-Pz combination should be considered, given that adding Cp2 and T8 will only add 0.22% of mean accuracy.

7.2.2. Best Channel Combinations for GMM-UBM

In this section, we report on the investigation into the best 2-, 3-, and 4-channel combinations for GMM-UBM. The procedure followed is similar to that for KNN, except that the best combinations are chosen in terms of AUC and EER performance.

Figure 7.4 shows the performance of all 2-channel combinations. In Fig. 7.4 the values above the diagonal represent the AUCs and the values below the diagonal represent 1 minus the EERs (Complementary EER). To find the best combinations, AUCs should be maximized and EERs should be minimized, meaning that 1 – EER should be maximized as well.

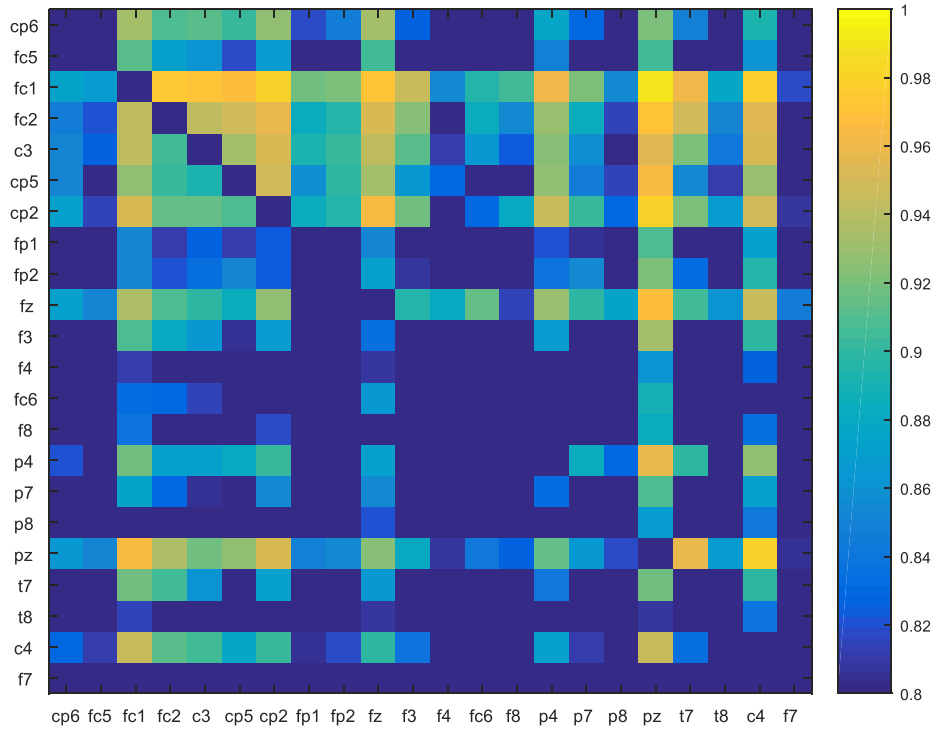

Fig. 7.4: AUCs (above diagonal) and 1-EERs (below diagonal) of all 2-channel combinations.

In Fig. 7.4, many elements are blue and dark blue because the image was zoomed to the range from 0.8 to 1. Just as for Fig. 7.1, Fc1-Pz provides the best performance: Mean AUC of 0.9899 and mean EER of 0.0357. The 5th and 95th percentiles of the AUC of the Fc1-Pz combination are 0.9730 and 0.9960 respectively, and the 5th and 95th percentile of EER for this combination are 0.0091 and 0.0700 respectively. The 95th percentile of the AUCs indicates that only 5% of the AUCs exceed the 95th percentile, and the 5th percentile indicates that 95% of all the AUCs exceed the 5th percentile, i.e. it is of interest to have a high 5th percentile for AUC. Similarly, for the EERs,

5% of all EERs exceed the 95th percentile and 95% of all EERs exceed the 5th percentile, so in this case it is of interest to have a low 95th percentile. Interestingly enough, the second best 2-channel combination appears to be Cp2-Pz, which was also highly ranked with KNN. Cp2-Pz achieves AUC CDF values of {0.9577 - 0.9806 - 0.9982} and EER CDF values of {0.0109 - 0.0491 - 0.0882}. Pz-C4 achieves a slightly higher AUC, but at the expense of EER, with CDF values of {0.0182 - 0.0536 - 0.1222}. Combinations involving Fc1, Fc2, Pz, and Cp2 are in the top 10 2-channel combinations, but their mean AUC are below 0.9800 and their mean EER above 0.0500.

Figure 7.5 shows the effect of adding one channel to Fc1-Pz. In Fig. 7.5, the mean AUCs fluctuate between 0.9700 and 0.9900. Similarly, the mean EERs fall between 0.0300 and 0.0900.

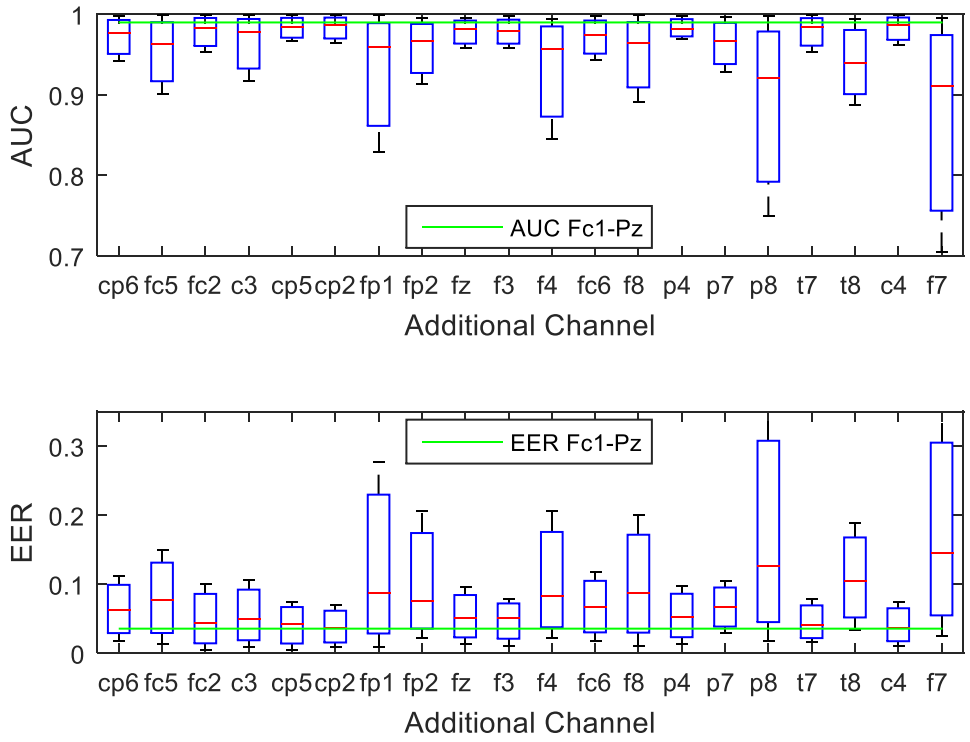

Fig. 7.5: AUCs and EERs of all 3-channel combinations that include Fc1-Pz.

The best 3-channel combination in this experiment is Fc1-Pz-Cp2, with AUC CDF values of {0.9643 - 0.9867 - 0.9990} and EER CDF values of {0.0086 - 0.0367 - 0.0700}. The second best 3-channel combination is Fc1-Pz-C4, with AUC CDF values of {0.9622 - 0.9865 - 0.9992} and EER CDF values of {0.0109 - 0.0373 - 0.0745}. The mean EERs for all other combinations exceed 0.0400 and the corresponding mean AUCs are below 0.9850. This experiment shows that several 3-channel combinations involving Fc1 and Pz achieve high AUCs and low EERs. Looking at the

mean (red lines), in Fig. 7.5, it is not advisable to combine Fc1-Pz with T8, P8, or F7. Looking at the worst case, P8, F7, and Fp1 should not be used.

This experiment suggests that a 2-channel combination may achieve higher performance than 3-channel combinations when using GMM-UBM with AR model features. The mean AUC and mean EER of Fc1-Pz-Cp2 are 0.9867 and 0.0367 respectively. On the other hand, the mean AUC and mean EER of Fc1-Pz are 0.9899 and 0.0357, which indicates that on average performance deteriorates when a third channel is added to Fc1-Pz. In addition, the AUCs and EERs of Fc1-Pz-Cp2 are more spread than those of Fc1-Pz. Since feature extraction and classification are less computationally demanding for the 2-channel combinations, which also happens to improve performance for the feature set (AR parameters) and classification scheme (GMM-UBM) of this section, leads to the strong suggestion that Fc1-Pz be used.

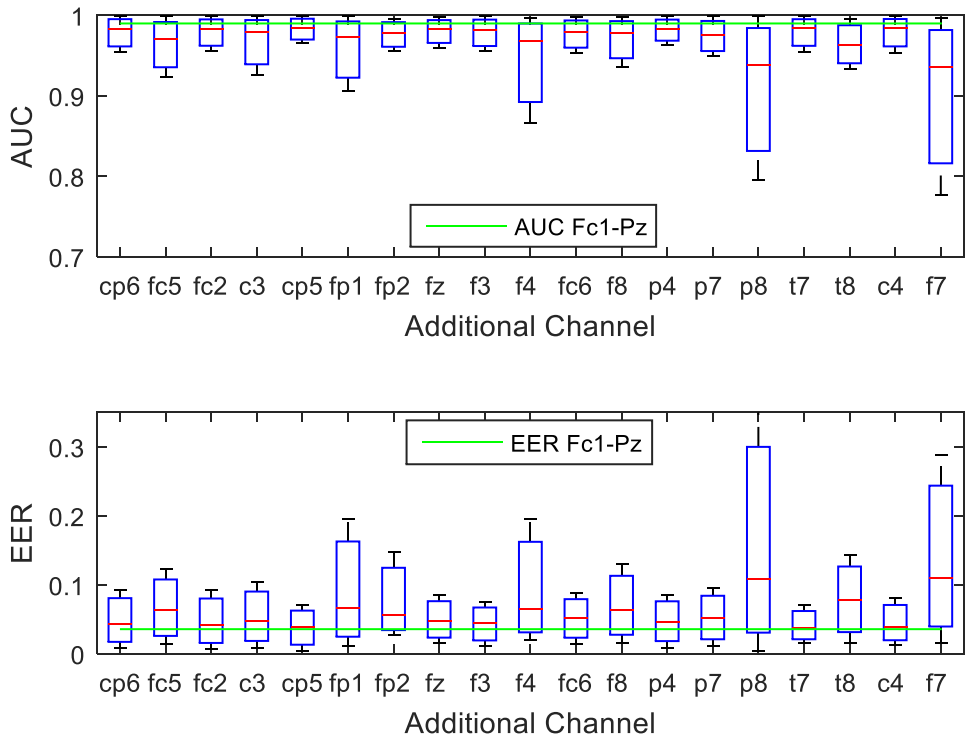

Fig. 7.6: AUCs and EERs of all 4-channel combinations that include Fc1-Pz-Cp2.

Figure 7.6 exhibits a behavior similar to that observed in Fig. 7.5, but performance degrades more. The best 4-channel combination, Fc1-Pz-Cp2-T7, has a mean AUC of 0.9844 and a mean EER of 0.0367. For this combination, the 5^{th} and 95^{th} percentiles of the AUCs are 0.9546 and 0.9988 respectively, and the 5^{th} and 95^{th} percentiles of the EERs are 0.0159 and 0.0705. Although

the mean EER did not change and the AUC is slightly smaller than that for the Fc1-Pz-Cp2 combination, computational complexity is not favorable. When using Fc1-Pz, feature vectors are 14-D, whereas they are 28-D when using Fc1-Pz-Cp2-T7. Besides, mean AUC is 0.9899 when using Fc1-Pz only. As a result, this experiment suggests that the number of channels be kept at 2.

Figure 7.7 illustrates a common ROC obtained when training and testing GMM-UBMs using pair Fc1-Pz. The closer the ROC curve comes to approximating the top-left corner the better the performance (AUC approaches 1).

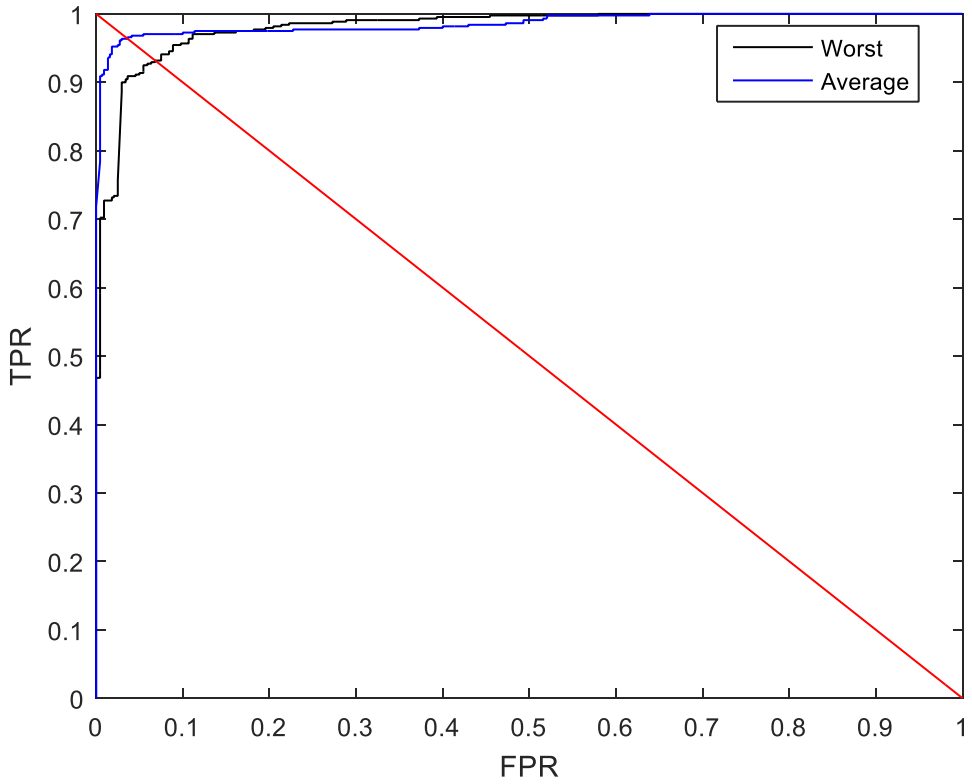

Fig. 7.7: ROC when pair Fc1-Pz is used in GMM-UBM.

Of all 30 test case permutations, each producing its own ROC, the worst case (green) and the averaged (blue) ROC are shown.

The area under the ROC curve is the AUC, and the intersection between the ROC curve and the red line, which is where the false acceptance (FP) and false rejection (FN) rates are equal, is the EER. The AUC for the average ROC shown is 0.9860 and its EER is 0.0364. What may be even more meaningful in practice is that for the worst case ROC, AUC is 0.9805 and EER is

0.0700, meaning that for all 30 test case configurations AUC is higher than 0.9805 and EER is 7% or less. Each of these ROC curves is based on using between 240 and 260 test vectors from each test subject, i.e. incorporating results from 1000 to 1100 test vectors.

An alternative way of presenting EER in a more detailed fashion is by way of detection error tradeoff (DET) curves, in which false negative rate (FNR) is plotted versus false positive rate (FPR) on logarithmic axes. Figure 7.8 shows the best, average, and worst EER cases obtained using GMM-UBM with pair Fc1-Pz.

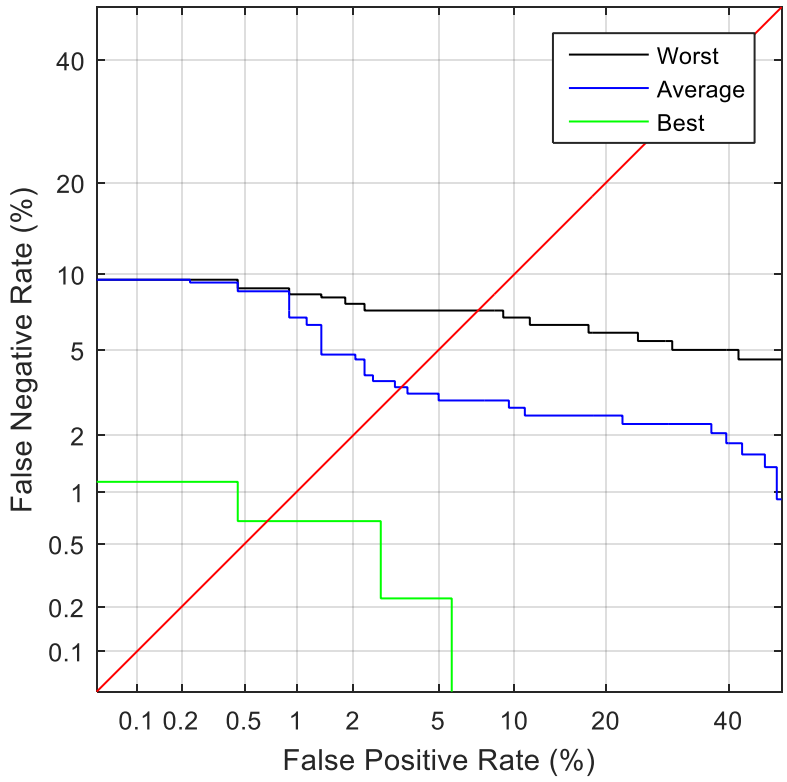

Fig. 7.8: DET curves when pair Fc1-Pz used in GMM-UBM.

EER for each test case are found at the intersections with the red line. Note that the best DET curve is very close to the bottom left corner, as happens when very few test vectors are misclassified; for this subject configuration, AUC was 0.9996 and EER was 0.0091. The worst case DET curve indicates that for an EER of 7% or less, the True Positive/Acceptance rate is at least 93%.

Of the methods explored here, GMM-UBM using a 2-channel combination yields the best results. A mean AUC of 0.9899 was found using GMM-UBM, which slightly outperforms 0.97 using SVMs [4]. Both studies used similar datasets: 8 subjects in our study vs 10 [4]; approximately 5 minutes of EEG recordings sampled at 512 Hz in our study vs approximately 2 minutes of EEG recordings sampled at 256 Hz [4]; lastly, windows of 2 s are used for feature extraction in our study vs windows of 1 s [4].

To provide a more comprehensive view of performance, AUC and EER are reported in terms of mean as well as 5^{th} and 95^{th} percentiles. For GMM-UBM the 5^{th} percentile of AUC was higher than 0.97 and the 95^{th} percentile was higher than 0.99, indicating AUC to be highly concentrated near 1.

In this chapter, the objective was to maximize performance of KNN and GMM-UBM when using AR(7) for the classification of A and NA. For KNN, 2, 3, and 4-channel combinations were found, and the best combination was Fc1-Pz-Cp2, with a CDF of {0.8988 – 0.9334 - 0.9690}, which is higher than the values reported in Chapter 4 (75 – 100%) for the 5-channel combination. For KNN, there was also a 2-channel combination, Fc1-Pz, that achieved a local maximum, but it was not as high as that of Fc1-Pz-Cp2. For GMM-UBM, classification performance was enhanced even more, to the point where the average GMM-UBM makes very few mistakes. The best combination for GMM-UBM was Fc1-Pz, with a CDF of {0.9730 – 0.9899 - 0.9996}, which outperforms the results in Chapter 5 for the 5-channel combination. Interestingly enough, most of the channel combinations overlap. For instance, pair Fc1-Pz maximized the performance of GMM-UBMs and also greatly improved the performance of KNN. The fact that most of the combinations matched suggests that there is something in those areas of the brain that occurs during ANT, and this response elicits AR(7) that are very different for A and NA subjects.

8. CONCLUSIONS

In this work, several machine learning algorithms were explored for the classification of A and NA subjects using AR coefficients as parameters. The experiments were conducted on a dataset collected and provided by our collaborator from the Psychology Department at Virginia Tech. The dataset consisted of 8 subjects, 4 A and 4 NA, whose EEG data was recorded as they performed several activities. After carefully studying the dataset using the algorithms disclosed earlier, one of the NA labels was switched to A. The algorithms used in this work were KNN, GMM-UBM, K-means, Soft KNN, and Soft GMM-UBM.

When using KNN and training with 4 subjects (2 A and 2 NA), classification performance was between 75% and 100% and confidence of classification was between 77% and 97%. The classifiers were found to be biased towards classifying test vectors as NA. Nevertheless, these results were encouraging because KNN performs no processing on the data in order to increase performance.

An algorithm that increases performance by trying to separate classes is GMM-UBM with expectation maximization. In this algorithm, a likelihood ratio is optimized so that the distance between the A and the NA subjects becomes maximal. When using GMM-UBM, the worst performance values were an AUC of 0.9285 and an EER of 0.1336. However, the average performance values were AUC of 0.98 and EER of 0.045.

Contrary to many works found in the literature, this work suggests that eyes-closed EEG data undermines discrimination. Activities that require attention proved to be the type of activities that maximize discrimination, whereas eyes-closed EEG data appears to be a contaminant that moves detection performance towards that of guessing.

In this work, algorithms that take into account the confidence of labels were used also. Concretely, KNN and GMM-UBM were softened for this purpose. When compared with their hard decisions, using labels that are perfect/approximately correct, performance seems to be slightly lower, which means that soft algorithms can approximate ideal models depending on the data and labels. Further, performance for the worst case scenario appears to increase when soft methods are used.

In order to reduce runtime and possibly increase performance, combinations involving fewer than 5 EEG channels were investigated for KNN and GMM-UBM. Surprisingly, most of these combinations were common across both algorithms, which may suggest that there is something special about these regions. In fact, when using one of these combinations for GMM-UBM, performance for the worst case was an AUC of 0.9730 and EER of 0.0700.

Although interesting results have been reported, there are other avenues that may yield interesting results. Neural Networks for the classification of A and NA should be explored because, even though they tend to overfit, they can perform operations on input data (either features or raw data) and combine the resulting data to boost classification performance; neural networks do so by performing multiplications, additions, and other operations as the data is propagated through the layers of the network. Also, it would be interesting to find exactly what kind of stimulus is the one that helps maximize discrimination the most. In this study, the 5-minute ANT block proved to be the most useful, but the ANT block does not continuously provide stimuli that require attention.

REFERENCES

[1] (2016, 3-Oct-2016). *Data and Statistics | ADHD | NCBDDD | CDC* [Online]. Available: http://www.cdc.gov/ncbddd/adhd/data.html

[2] *Diagnostic and Statistical Manual of Mental Disorders (DSM-5®)*. Washington, DC: American Psychiatric Association, 2013.

[3] R. L. Kennedy, "The prognosis of sequelae of epidemic encephalitis in children," *American Journal of Diseases of Children,* vol. 28, pp. 158-172, 1924.

[4] S. Stryker, "Encephalitis lethargica: The behavior residuals," *Training School Bulletin,* vol. 22, pp. 152-157, 1925.

[5] J. F. Lubar, "Discourse on the development of EEG diagnostics and biofeedback for attention-deficit/hyperactivity disorders," *Biofeedback and Self-regulation,* vol. 16, pp. 201-225, 1991.

[6] C. Bradley, "The behavior of children receiving benzedrine," *American journal of Psychiatry,* vol. 94, pp. 577-585, 1937.

[7] A. A. Strauss and L. E. Lehtinen, *Psychopathology and education of the brain-injured child*, 1947.

[8] *Diagnostic and Statistical Manual of Mental Disorders (DSM)*. Washington, DC: American Psychiatric Association, 1987.

[9] *Diagnostic and Statistical Manual of Mental Disorders (DSM)*. Washington, DC: American Psychiatric Association, 1994.

[10] S. M. Snyder, H. Quintana, S. B. Sexson, P. Knott, A. Haque, and D. A. Reynolds, "Blinded, multi-center validation of EEG and rating scales in identifying ADHD within a clinical sample," *Psychiatry research,* vol. 159, pp. 346-358, 2008.

[11] (2016, 03-Oct-2016). *ADHD Behavior Treatment for Preschoolers | ADHD | NCBDDD | CDC* [Online]. Available: http://www.cdc.gov/ncbddd/adhd/treatment.html

[12] "ADHD: Clinical Practice Guideline for the Diagnosis, Evaluation, and Treatment of Attention-Deficit/Hyperactivity Disorder in Children and Adolescents," *Pediatrics,* 2011.

[13] S. E. Nissen, "ADHD drugs and cardiovascular risk," *New England Journal of Medicine,* vol. 354, pp. 1445-1448, 2006.

[14] J. B. Cullen, "The impact of fiscal incentives on student disability rates," *Journal of Public Economics,* vol. 87, pp. 1557-1589, 2003.

[15] E. Dhuey and S. Lipscomb, "Funding special education by capitation: Evidence from state finance reforms," *Education,* vol. 6, pp. 168-201, 2011.

[16] T. E. Elder, "The importance of relative standards in ADHD diagnoses: evidence based on exact birth dates," *Journal of health economics,* vol. 29, pp. 641-656, 2010.

[17] D. L. Schomer and F. L. Da Silva, *Niedermeyer's electroencephalography: basic principles, clinical applications, and related fields*: Lippincott Williams & Wilkins, 2012.

[18] V. L. Towle, J. Bolaños, D. Suarez, K. Tan, R. Grzeszczuk, D. N. Levin, *et al.*, "The spatial location of EEG electrodes: locating the best-fitting sphere relative to cortical anatomy," *Electroencephalography and clinical neurophysiology,* vol. 86, pp. 1-6, 1993.

[19] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of modern Physics,* vol. 65, p. 413, 1993.

[20] K. J. Murphy and J. A. Brunberg, "Adult claustrophobia, anxiety and sedation in MRI," *Magnetic resonance imaging,* vol. 15, pp. 51-54, 1997.

[21] J. F. Schenck, "The role of magnetic susceptibility in magnetic resonance imaging: MRI magnetic compatibility of the first and second kinds," *Medical physics,* vol. 23, pp. 815-850, 1996.

[22] H. Begleiter, "Evoked Potential Primer," *Journal of Clinical Neurophysiology,* vol. 3, pp. 270-271, 1986.

[23] S. J. Luck, *An introduction to the event-related potential technique*: MIT press, 2014.

[24] W. O. Tatum, "Ellen R. Grass lecture: Extraordinary eeg," *The Neurodiagnostic Journal,* vol. 54, pp. 3-21, 2014.

[25] E. AlEissa and S. Bendabi. (2016, 28-Oct-2016). *First Adult Seizure Workup*. Available: http://emedicine.medscape.com/article/1186214-workup#c9

[26] N. Ishida, K. Kasamo, Y. Nakamoto, and J. Suzuki, "Epileptic seizure of El mouse initiates at the parietal cortex: depth EEG observation in freely moving condition using buffer amplifier," *Brain research,* vol. 608, pp. 52-57, 1993.

[27] B. Abibullaev and J. An, "Decision support algorithm for diagnosis of ADHD using electroencephalograms," *Journal of medical systems,* vol. 36, pp. 2675-2688, 2012.

[28] M. Ahmadlou and H. Adeli, "Wavelet-synchronization methodology: a new approach for EEG-based diagnosis of ADHD," *Clinical EEG and Neuroscience,* vol. 41, pp. 1-10, 2010.

[29] V. J. Monastra, *Unlocking the potential of patients with ADHD: A model for clinical practice*: American Psychological Association, 2008.

[30] A. S. Rowland, C. A. Lesesne, and A. J. Abramowitz, "The epidemiology of attention-deficit/hyperactivity disorder (ADHD): a public health view," *Mental retardation and developmental disabilities research reviews,* vol. 8, pp. 162-170, 2002.

[31] C. W. Anderson, E. A. Stolz, and S. Shamsunder, "Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks," *IEEE Transactions on Biomedical Engineering,* vol. 45, pp. 277-286, 1998.

[32] C. Guger, G. Edlinger, W. Harkam, I. Niedermayer, and G. Pfurtscheller, "How many people are able to operate an EEG-based brain-computer interface (BCI)?," *IEEE transactions on neural systems and rehabilitation engineering,* vol. 11, pp. 145-147, 2003.

[33] Q. Wang and O. Sourina, "Real-time mental arithmetic task recognition from EEG signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 21, pp. 225-232, 2013.

[34] M. Gaub and C. L. Carlson, "Gender differences in ADHD: a meta-analysis and critical review," *Journal of the American Academy of Child & Adolescent Psychiatry,* vol. 36, pp. 1036-1045, 1997.

[35] C. A. Mann, J. F. Lubar, A. W. Zimmerman, C. A. Miller, and R. A. Muenchen, "Quantitative analysis of EEG in boys with attention-deficit-hyperactivity disorder: Controlled study with clinical implications," *Pediatric neurology,* vol. 8, pp. 30-36, 1992.

[36] R. J. Chabot, H. Merkin, L. M. Wood, T. L. Davenport, and G. Serfontein, "Sensitivity and specificity of QEEG in children with attention deficit or specific developmental learning disorders," *CLINICAL ELECTROENCEPHALOGRAPHY-CHICAGO-,* vol. 27, pp. 26-34, 1996.

[37] V. J. Monastra, J. F. Lubar, M. Linden, P. VanDeusen, G. Green, W. Wing, *et al.*, "Assessing attention deficit hyperactivity disorder via quantitative electroencephalography: an initial validation study," *Neuropsychology,* vol. 13, p. 424, 1999.

[38] V. J. Monastra, J. F. Lubar, and M. Linden, "The development of a quantitative electroencephalographic scanning process for attention deficit–hyperactivity disorder: Reliability and validity studies," *Neuropsychology,* vol. 15, p. 136, 2001.

[39] S. Koehler, P. Lauer, T. Schreppel, C. Jacob, M. Heine, A. Boreatti-Hümmer, *et al.*, "Increased EEG power density in alpha and theta bands in adult ADHD patients," *Journal of neural transmission,* vol. 116, pp. 97-104, 2009.

[40] S. K. Loo and S. Makeig, "Clinical utility of EEG in attention-deficit/hyperactivity disorder: a research update," *Neurotherapeutics,* vol. 9, pp. 569-587, 2012.

[41] C. A. Magee, A. R. Clarke, R. J. Barry, R. McCarthy, and M. Selikowitz, "Examining the diagnostic utility of EEG power measures in children with attention deficit/hyperactivity disorder," *Clinical Neurophysiology,* vol. 116, pp. 1033-1040, 2005.

[42] I. Buyck and J. R. Wiersema, "Resting electroencephalogram in attention deficit hyperactivity disorder: developmental course and diagnostic value," *Psychiatry research,* vol. 216, pp. 391-397, 2014.

[43] S. K. Loo and M. Arns, "Should the EEG–Based Theta to Beta Ratio Be Used to Diagnose ADHD?," *The ADHD Report,* vol. 23, pp. 8-13, 2015.

[44] M. D. Liechti, L. Valko, U. C. Müller, M. Döhnert, R. Drechsler, H.-C. Steinhausen, *et al.*, "Diagnostic value of resting electroencephalogram in attention-deficit/hyperactivity disorder across the lifespan," *Brain topography,* vol. 26, pp. 135-151, 2013.

[45] A. Mueller, G. Candrian, J. D. Kropotov, V. A. Ponomarev, and G.-M. Baschera, "Classification of ADHD patients on the basis of independent ERP components using a machine learning system," *Nonlinear Biomedical Physics,* vol. 4, p. 1, 2010.

[46] K. Sadatnezhad, R. Boostani, and A. Ghanizadeh, "Classification of BMD and ADHD patients using their EEG signals," *Expert Systems with Applications,* vol. 38, pp. 1956-1963, 2011.

[47] E. M. Thomas, A. Temko, G. Lightbody, W. P. Marnane, and G. B. Boylan, "A gaussian mixture model based statistical classification system for neonatal seizure detection," in

2009 IEEE International Workshop on Machine Learning for Signal Processing, 2009, pp. 1-6.

[48] S. J. Orfanidis, *Optimum Signal Processing*: Collier Macmillan, 1988.

[49] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, pp. 716-723, 1974.

[50] D. M. Simpson, A. F. C. Infantosi, J. Junior, A. Peixoto, and L. de Sousa Abrantes, "On the selection of autoregressive order for electroencephalographic (EEG) signals," in *Circuits and Systems, 1995., Proceedings., Proceedings of the 38th Midwest Symposium on*, 1995, pp. 1353-1356.

[51] P. M. Djuric and S. M. Kay, "Order selection of autoregressive models," *IEEE Transactions on signal processing*, vol. 40, pp. 2829-2833, 1992.

[52] M. I. Posner and S. E. Petersen, "The attention system of the human brain," DTIC Document 1989.

[53] M. R. Rueda, J. Fan, B. D. McCandliss, J. D. Halparin, D. B. Gruber, L. P. Lercari, *et al.*, "Development of attentional networks in childhood," *Neuropsychologia*, vol. 42, pp. 1029-1040, 2004.

[54] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*: Wiley, 2010.

[55] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *International conference on database theory*, 1999, pp. 217-235.

[56] T. Koenig, D. Studer, D. Hubl, L. Melie, and W. Strik, "Brain connectivity at different time-scales measured with EEG," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 360, pp. 1015-1024, 2005.

[57] P. Raman and A. A. (Louis) Beex, "Using LSF features for speaker verification in noise," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 697-701.

[58] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, pp. 827-832, 2015.

[59] P. Nguyen, D. Tran, T. Le, X. Huang, and W. Ma, "EEG-based person verification using multi-sphere SVDD and UBM," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013, pp. 289-300.

[60] D. A. Reynolds, "Automatic speaker recognition using Gaussian mixture speaker models," in *The Lincoln Laboratory Journal*, 1995.

[61] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, pp. 19-41, 2000.

[62] C. H. You, H. Li, and K. A. Lee, "Relevance factor of maximum a posteriori adaptation for GMM–NAP–SVM in speaker and language recognition," *Computer Speech & Language*, vol. 30, pp. 116-134, 2015.

[63] "MSR Identity Toolbox," ed: Microsoft Corporation.

[64] G. V. Tcheslavski and A. A. (Louis) Beex, "Phase synchrony and coherence analyses of EEG as tools to discriminate between children with and without attention deficit disorder," *Biomedical Signal Processing and Control*, vol. 1, pp. 151-161, 2006.

[65] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, pp. 580-585, 1985.

[66] C. Thiel, "Classification on soft labels is robust against label noise," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2008, pp. 65-73.

[67] D. Tran, T. Van Le, and M. Wagner, "Fuzzy Gaussian mixture models for speaker recognition," in *ICSLP*, 1998.

[68] H. N. Pinheiro, T. I. Ren, G. D. Cavalcanti, T. I. Jyh, and J. Sijbers, "Type-2 fuzzy GMM-UBM for text-independent speaker verification," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 4328-4331.