Summer 7-19-2021

Common Spatial Pattern Detection of Concept Semantic Relatedness Using Combined Event-Related Potentials and Frequency Spectrum Features

Michael Doyel

Common Spatial Pattern Detection of Concept Semantic Relatedness Using Combined Event-Related Potentials and Frequency Spectrum Features

A Thesis

Submitted to the Faculty

of

Rose-Hulman Institute of Technology

by

Michael Joseph Doyel

In Partial Fulfillment of the Requirements for the Degree

of

Master of Science in Biomedical Engineering

June 2021

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Final Examination Report

Michael Doyel
Name

Biomedical Engineering
Graduate Major

Thesis Title Common Spatial Pattern Detection of Concept Semantic Relatedness Using Combined Event-Related Potentials and Frequency Spectrum Features

DATE OF EXAM: July 19, 2021

EXAMINATION COMMITTEE:

	Thesis Advisory Committee	Department
Thesis Advisor:	Alan Chiu	BBE
	William Weiner	BBE
	Jennifer O'Connor	BBE
	Mark Brandt	CHEM
	Yongjin Kim	ECE

PASSED X

FAILED ____________

Defense Report

Abstract

Doyel, Michael Joseph

M.S.B.E.

Rose-Hulman Institute of Technology

June 2021

Common Spatial Pattern Detection of Concept Semantic Relatedness Using Combined ERP and Frequency Spectrum Features

Thesis Advisor: Dr. Alan Chiu

One of the chief contributing factors to slowing down BCI spellers for users with profound disabilities is backtracking to delete mistakes or correct certain selections. The ability to design an EEG-based strategy to identify the desire to make corrections in the BCI speller would enhance the user experience and bit rate of the device. Past efforts suggested that Common Spatial Patterns (CSP) may show promise in helping detect and classify semantic violations in reading despite CSP not being widely used in event-related potential (ERP) applications. Semantic violations in EEG often exhibit deflections in the N400 and P600 region coinciding with the violation. This research aims to create a CSP model that can improve the classification accuracy of semantic violations in reading by incorporating neural oscillation information. Visual stimuli consisting of 150 pairs of nouns from *Maguire et al.* (2010) and *Calvo et al.* (2018) were presented, where the first word served as the Primer and the second word served as the Target. EEG signals from 14 channels were parsed to obtain the average N400 potential, the average P600 potential, and signal power in the alpha and theta bands, creating a 56-dimension (14 channels by 4 feature types) feature space. The CSP algorithm was implemented to improve orthogonality in the feature space, and the feature space dimension was reduced to 2. Three types of classification strategies Linear Discriminant

Analysis (LDA), Naïve Bayes (LB), and K-Nearest Neighbor (KNN), were implemented. Graphical analysis by CSP showed that while individual features did not appear separable, a higher dimension dataset including all four feature types does demonstrate separability. The 10-fold validation results showed that two-class models optimized for individual subjects achieved accuracies ranging from 50% to 66% with LDA, 78% to 98% with KNN, and 76% to 100% with NB. Examining the mixing matrices during the dimension reduction step in CSP suggested that alpha frequency band and EEG locations P7, F4, F3, AF4, and FC6 consistently play a critical role in the success of these classifiers across the different subjects.

Acknowledgments

I extend my extreme gratitude to each of the following:

Dr. Chiu for his willingness to help guide me through this thesis, for his humor and flexibility during all unexpected circumstances. Thank you for all of your late night responses and video calls where you helped me sort out any issues that I had!

Andre Adam for being a great and supportive friend, constantly answering my MATLAB questions and helping verify all of my code.

Hannah Bach and Shogo Honda for their infectious excitement for the original project; while we didn’t get to implement everything quite as planned, I am still very grateful for the laughs and brainstorming sessions we had.

Kate Holland for always providing a source of entertainment and distraction throughout the past year.

Table of Contents

Contents

List of Figures

Figure **Page**

List of Tables

List of Abbreviations

AAC – Augmentative & Alternative Communication

SGD – Speech Generating Device

ERP – Event-Related Potential

EEG - Electroencephalography

BCI – Brain-Computer Interface

CNS – Central Nervous System

SNR – Signal to Noise Ratio

CSP - Common Spatial Patterns

LDA – Linear Discriminant Analysis

QDA – Quadratic Discriminant Analysis

NB – Naïve Bayes

KNN – K Nearest Neighbor

1. Introduction

Augmentative and Alternative Communication (AAC) devices are used to reduce the daily stress and burden for those who suffer from an expressive communication disorder whether congenital or acquired [43]. In particular, high-end AAC's referred to as Speech Generating Devices (SGD) are of great use to patients who require an AAC solution that can be operated with limited motor function, while still allowing the patient to retain a certain level of communicative complexity [43]. With an estimated 2 million Americans alone benefitting from the use of AAC devices, the ability to improve the flexibility and adaptability of these devices is paramount to increasing the quality of life for those who use this technology [44][4].

One of the current issues that results in significant time delays in the use of SGDs is the detection and correction of semantic violations in the generated text. A sentence such as, "He likes avocado and red onions on his socks", is grammatically correct by syntax and would be difficult to detect and correct without cycling through a complete set of commands in order to be able to delete the semantic error. For Event-Related Potential (ERP) based Brain-Computer Interface (BCI) spellers, the rapid detection of semantic errors and their streamlined correction is a key step to increasing device speed and precision, which ultimately leads to an increase in quality of life [5][6].

The N400 ERP was identified as early as 1980 when *Kutas et al.* determined that semantic incongruities in sentences could be correlated with a strong negative deflection of the potential in an Electroencephalography (EEG) reading 400ms after the onset of the stimulus [7]. Since its discovery it has been the subject of many different research articles and studies in conjunction with many positive deflection ERPs such as the P300 and P600, however, many studies conducted have had differing and sometimes conflicting results when measuring semantic errors with these ERPs

[8], [9]. For this reason, spectral analysis is being used as an additional layer of verification in the experiments in order to account for subject attention in the tasks that may have some effect on the attenuation of these two ERPs [9]–[11].

The purpose of this thesis is to analyze event-related potentials in EEG data that correspond to detected semantic violations in text. This includes the analysis of ERP data in the N400 and P600 regions, as well as alpha and theta brainwaves that may correspond to the alertness of the individual being tested.

With the inclusion of the P600 ERP and the alpha and theta brainwaves, it is believed that the model shown by *Calvo et al.* can achieve enhanced efficiency when these variables are accounted for either by accounting for lapses in attention or by analysis of later onset semantic recognition in the P600 region [12]. It is hoped that integrating many of these different variables can help inform the limits in which spatial coherence analysis can be employed to improve the detection and resolution of (BCI) speller technology.

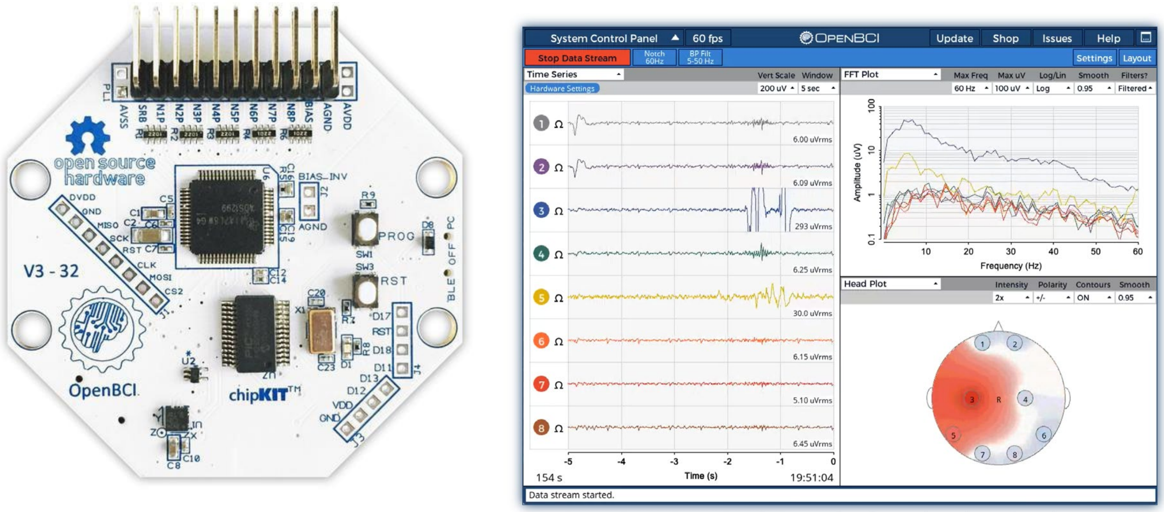
2. Background Information

2.1 Brain-Computer Interfaces

The burden of illness placed on people living with degenerative disease and those who have suffered traumatic spinal cord injury is large and often restricts the level of independence and communication ability these people may have [43][44]. Brain-Computer Interfaces, or BCIs, provide an avenue for reconciling this loss of function and augmenting the user's ability to communicate with the world through physical means. BCI devices can function in various roles while also utilizing many different brain wave signals to accomplish their goals. Inputs can include EEG signals measured from electrodes on the scalp, EOG signals which measure the movement of the eyes, ECoG implantations in the scalp, and even measurements of pupil size oscillation[1][13]. These inputs then undergo filtering, algorithmic feature extraction, and classification to isolate meaningful data which is then implemented in the BCI hardware to generate a real-world output [14][1][15]. Because this type of technology focuses on creating outputs based on brainwaves from the central nervous system (CNS), it provides alternate solutions for accomplishing tasks that may otherwise require vocal or musculoskeletal function [11]. For example, in a tetraplegic patient with no voluntary control over speech, BCIs can provide a rapid and non-invasive method for artificial speech synthesis [16]. Additionally, BCIs have even shown promise in more complex motor control of prostheses and robotic implementations that can further patient ability for independence and self-care [14].

However, despite their many uses, BCI devices are not without their downsides. A major roadblock for many BCI technologies is the ability to translate the results found in a lab setting to a real-world scenario, as lab environments are often carefully controlled and may not have the variety of stimuli that a person would be subject to in a normal use case [17]. Furthermore, these devices can be costly and oftentimes require extensive training periods and calibration for each subject, leading to slower access and more cumbersome use. Depending on the type of BCI, it may not be financially or technologically feasible for a consumer to use, such as devices utilizing fMRI, for other devices such as ECoG, users may be more wary of the invasive nature of these implementations. Additionally, it is also important that BCI devices be robust enough to survive sustained long-term use and potential rough handling by users without failure, while also meeting criteria for safety and functionality in technically challenging environments [1].

Fortunately, with the rapid expansion of research in the BCI field, many different groups are offering solutions that help to tackle some of these concerns [14]. OpenBCI is one such group that is aiming to reduce the cost of BCI systems by providing BCI hardware capable of research-grade data collection for a fraction of the normal price. In addition, the open source nature of the OpenBCI mission allows for collaboration in the BCI community, and they even provide an open-source GUI, shown in Figure 1 below, capable of tracking live EEG data and interacting with matlab and other programming interfaces. Feasibility studies have demonstrated that BCIs constructed with the OpenBCI hardware can be used in an uncontrolled setting to produce results similar to that of current clinical-grade BCI systems. Since these implementations cost only around 10-20% of the typical price of a clinical BCI, further advances that improve network stability and assist in detection of measurement artifacts could help bring BCI devices to many more people. However, it is important to note that these systems do not have any type of medical or FDA approval, so the current usage in patients is strictly limited [19].



Vert Scale 200 uV
Window 5 sec
Max Freq 60 Hz
Max uV 100 uV
Log/Lin Log
Smooth 0.95
Filters? Filtered
Intensity 2x
Polarity +/-
Contours ON
Smooth 0.95

Data stream started.

Figure 1. Cyton Board and OpenBCI GUI. Advances is BCI technology continue to make research and accessibility a greater reality for many more people. Low-cost tools such as this data collection board and data streaming GUI continue to foster innovation in the BCI field.

2.2 Electroencephalography

Electroencephalography (EEG) is a non-invasive technique that allows for the measurement of electrophysiological activity in the brain. This measurement is done through the placement of electrodes on specific regions of the head through either a scalp cap or headset. The standardized placements for these electrodes are known as the 10-20, 10-10, and the 10-5 systems where the 20, 10, and 5 represent the percent subdivision between landmarks on the skull where an electrode is placed [1][20]. The 10-20 system was one of the first standardized electrode placements and consists of 21 total placement positions labeled by the area of the head they measure from and the hemisphere from which they are measuring [21]. The 10-10 system includes the electrodes from the 10-20 division while adding additional subdivisions for a total of 74 electrodes. The same is true of the 10-5 system shown in Figure 2, which brings the total number of electrodes in this system to 142 and provides a high level of granularity to the measurements. The error associated with any one of these systems is directly related to the number of electrodes, but also to the model of the head being used. Electrodes that are misplaced in respect to their theoretical models will generate deviant data, and it is therefore crucial that electrode placement on each individual subject is carefully controlled to ensure quality data is produced [1][22].

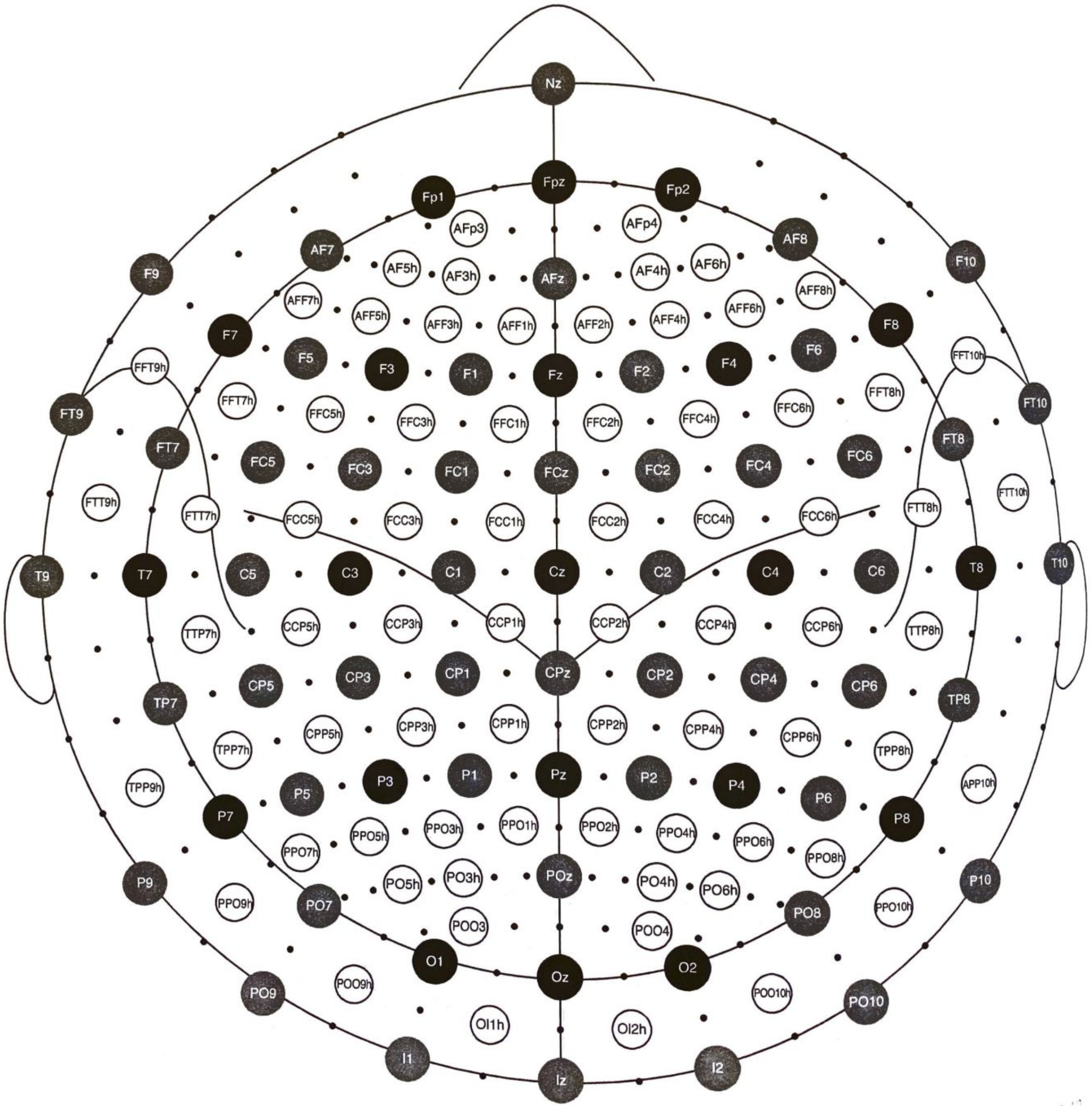


Figure 2. Complete diagram of a 10-5 EEG system illustrating the typical electrode placement on the scalp of the subject [1]. The 10-5 electrode setup contains the electrode locations for the 10-20 system in the solid black, the 10-10 system in the fuzzier black, and the 10-5 system in the additional white circles.

Because EEG data are typically in the realm of microvolts and millivolts, it is necessary to amplify these signals into larger values for data processing, and in most cases a signal will run through an analog-to-digital converter (ADC) in order to enable processing and storage of the signal. As an EEG signal is a continuous-time signal, it must be sampled to be useful in application and to avoid aliasing, which is a distortion to the original signal due to an improper sampling frequency. Sampling at the Nyquist frequency allows for a near-perfect reconstruction of the signal and avoid aliasing as shown in Figure 3, because most activity in EEG scalp recordings lies at 80Hz and below, a sampling rate of 160Hz is sufficient, but depending on the types of filtering used a lower value may be appropriate [1][23].

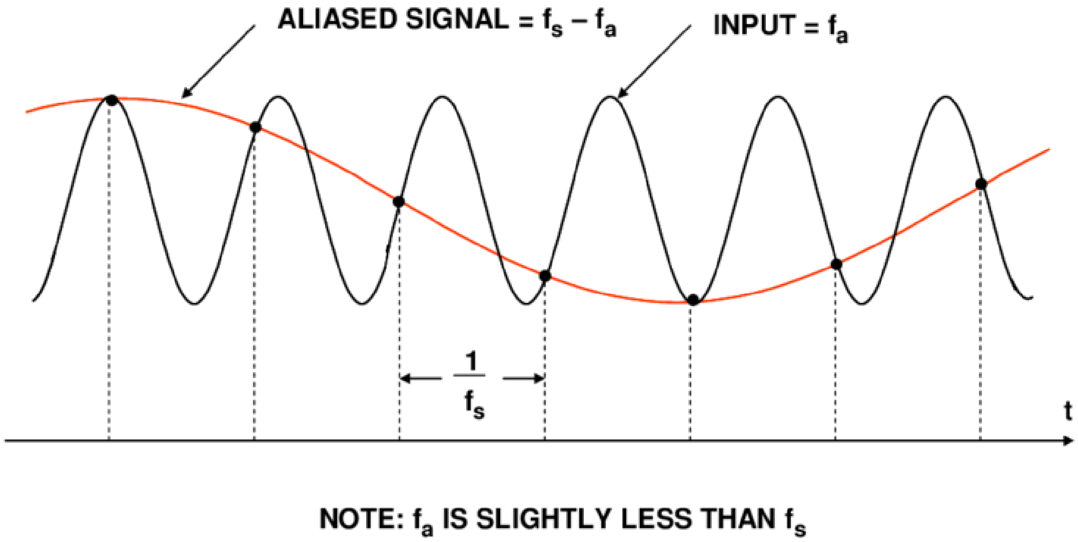

ALIASED SIGNAL = $f_s - f_a$
INPUT = f_a
$\frac{1}{f_s}$
t
NOTE: f_a IS SLIGHTLY LESS THAN f_s

Figure 3. Demonstration of the negative effects of aliasing on time-domain data [2]. You can see that as the original signal as sampled and replotted at an improper sampling frequency, the reconstructed signal becomes distorted and loses much of the info contained in the original signal.

After sampling the data, an EEG destined for use in a BCI device will need to undergo a few more steps before it can be used to create meaningful outputs. Since EEG signals with frequencies above 40Hz tend to have a low signal to noise ratio (SNR), these components are generally filtered out with the use of a bandpass filter ranging from 0.5Hz to 40Hz that also removes some amplification artifacts. For EEG data sampled at a higher rate than the Nyquist frequency it may be useful to decimate the signal to the Nyquist rate for more efficient data processing. It is also best practice to normalize a signal by subtracting the mean value of the signal from itself [9][10][24]. At this point the signal can then be processed through spatial filtering to extract event-related potentials and other useful data while also providing a metric to analyze and assess data.

2.3 Event-Related Potentials & Rhythms

One of the key ways in which EEG data is analyzed is through event-related potentials (ERPs). An ERP is an electrical response in the brain that correlates to stimuli presented in the environment and is generally measured through the use of EEG [25]. These time-domain signals allow for researchers to correlate associations and actions with EEG readings for use in research, and their non-invasive nature has made them exceedingly popular in a wide array of studies [8], [26]–[29]. ERPs generally fall within preselected time windows and derive their names from these windows and the direction in which the EEG signal deflects from their mean. For example, an N400 component is an ERP signal component that usually produces a negative voltage peak within the time frame of 350-500ms after the onset of a stimulus. Furthermore, these specific ERPs are usually elicited by specific events such as semantic or grammatical incongruities in sentence structure while reading, allowing their presence to be used as a form of detection for these incongruities in EEG data[30].

The N400 and P600 ERPs are of particular interest in studies of language comprehension as both of these have been shown to relate to violations in syntax as well as semantic incongruity in language [26], [31], [32]. Semantic violations are known to elicit a P600 response that has been attributed to a mental repair of incorrect sentence structure, however, they have also been shown to appear in some contexts with N400 peaks. N400 peaks generally occur when an unexpected change occurs in a sentence such as “He went to the barbershop for a salad” as opposed to the expected “He went to the barbershop for a haircut.”. The expectancy of a word in a given context can also change the magnitude in which the N400 component appears [7], [26], [33][34]. These two ERPs can be seen as they appear in the time domain in Figure 4 below.

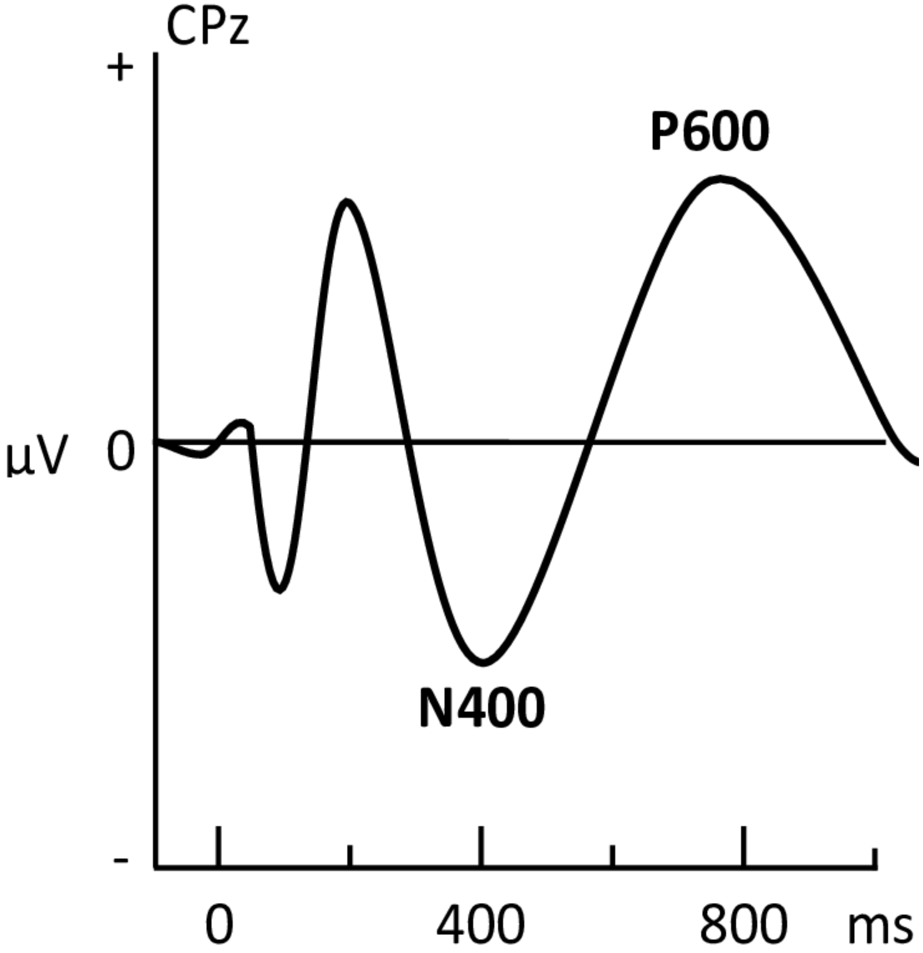


Figure 4. N400 and P600 ERP time-domain responses to a presented stimulus. Both of these ERPs are tied to language comprehension and understanding and may serve as markers for semantic understanding [3].

From the perspective of BCI, signals such as these give a physical metric for which assistive devices can be designed around and improved. A current issue for users who rely on ERP-based BCI spellers is the ability of these devices to provide rapid and accurate communication [5][35]. Feature detection of ERP signals in conjunction with other metrics such as power spectral density in the theta and alpha bands related to attentiveness could allow for better design and error correction in these devices that can be tailored to the specific errors associated with individual ERPs.

Neural rhythms are repetitive electrical activities that appear on EEG and relate to different types of neural activity in the brain. Many of these rhythms have been shown to correlate with attention and focus as well as restfulness, which make these rhythms good candidates as features for EEG analysis as they can be used to determine whether a subject was focused on the task and examine how task results may differ based on this focus. Unlike ERPs which occur in the time domain of an EEG signal, neural rhythms are features that occur in the frequency domain of an EEG signal. Alpha and theta waves are of particular note as their presence indicates levels of both alertness and relaxation, which are key control variables in EEG research [36], [37].

Alpha rhythms are those that occur between 8-12 hertz in the frequency domain of an EEG signal measured on the occipital lobe. In general, alpha waves are not highly prevalent in most EEG readings of awake and alert subjects as they are at their strongest when a subject is in a restful state with their eyes closed[36]. However, high levels of alpha waves in an EEG experiment can indicate confounding variables such as drowsiness of a subject or lack of focused attention on a task. Measuring this activity is important in adjusting the parameters of a classification filter based on the attention that a subject may be allocating to the task at hand [37]. Theta rhythms are similar to alpha rhythms though they occur in the range of 4-8 hertz and are generally strongest around the hippocampus. Studies have linked these theta rhythms not only to alertness and activation of the hippocampus, but also to semantic memory and translation of thought to motion, both of which are useful in a two class setting examining semantic relationships in language [38].

2.4 Common Spatial Patterns

Common Spatial Patterns or CSP is a mathematical method used to maximize the variance between multivariate signals through rotation on the axis. By inputting a multi-channel time signal into a CSP algorithm, it is possible to classify the most variant of these channels and use them to maximize the discrimination achieved [39]. This allows for the analysis of large quantities of channel data while simultaneously helping to remove unnecessary channels that don’t vary in any meaningful way [15], [24], [40]. CSP, in its most basic sense, is a rotational separation of two sets of data that projects both vector spaces in their most orthogonal position. Based on this transformation, it becomes possible to sort each point into either class one or class two depending on their location from the origin point[15], [24]. Because this axial rotation around the origin ignores the overall mean of each data set, it makes CSP a useful tool for analyzing data where there is significant overlap in the mean, such as applications of extracting event-related potentials from signal noise as the CSP algorithm can increase the variance of these ERPs to help distinguish them from the background signal[39].

2.5 Classification Algorithms

Classification algorithms are algorithms that can be used to sort data into specific classes and groupings based on information available in the data. Naïve Bayes (NB), is a classification algorithm based on Bayes Theorem. It makes its predictions based on the probability of events occurring given a subset of variables. In this case, that subset of variables is the feature inputs from our EEG and CSP analysis. K-Nearest Neighbor (KNN), is an algorithm that classifies data points based on their nearest neighbor just as the name suggests. The K value gives the weighting for how many neighbors are looked at, so K = 3 will classify a point based on the 3 nearest neighbors. An example of this classification style for a 3 class dataset can be seen in Figure 5 below.

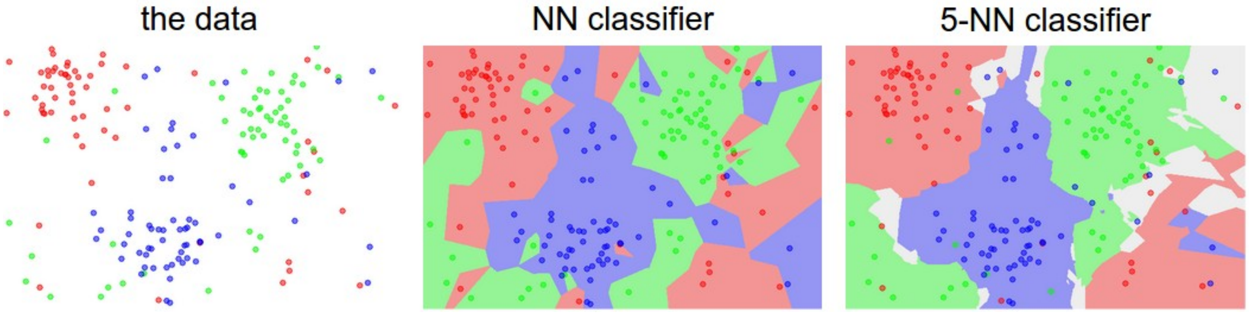


Figure 5. KNN classification for a 3 class data set. This figure illustrates the original data spread, a classification method using only the first nearest neighbor, and a set that looks at the five closest neighbors to each point [45].

Discriminant Analysis functions by creating a decision surface that partitions and separates the data based on the training set. Two types of discriminant analysis are shown in Figure 6 below, these are the quadratic and linear discriminant analysis (QDA & LDA).

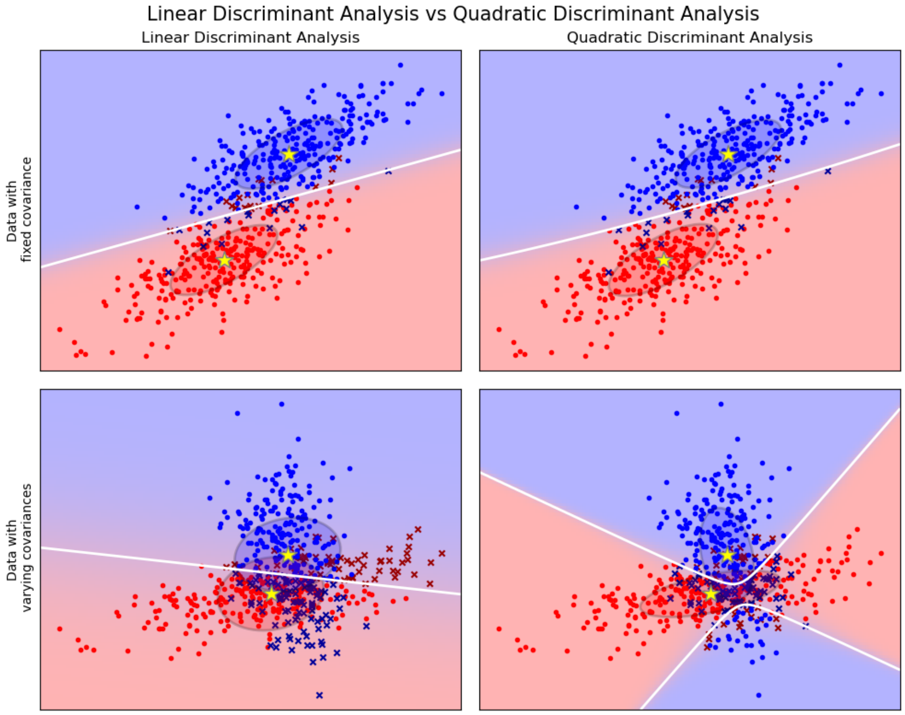


Figure 6. Examples of LDA and QDA analysis of two datasets. It is apparent that the QDA decision surface looks very similar to the separation achieved by a CSP filtering algorithm [46].

2.6 Overview of *Calvo et al.* Experiment

The dataset used in this study was initially collected by *Calvo et al.* for their research into classifying EEG data based on semantic groupings. The EEG collected for their study was based on semantic response to groupings of concrete nouns such as Oven-Sink and Cow-Milk. Eighteen postgraduate students from the Center of Computing research at the National Polytechnic Institute in Mexico City served as the subjects for the experiment. A total of 150 word pairs were presented to each subject in blocks of 30, with a word pair appearing on screen in size 70 Arial font for 5 seconds. The left and right arrow keys would be pressed by each subject to indicate whether they thought the word pairs were semantically related or not. EEG data would be recorded using the Emotic EPOC headset with a 10-20 electrode system using 14 channels at AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, and AF4. The sampling bandwidth for this experiment was 0.2 -

45 Hz. Information about the arrow keystrokes, timing data, stimulus onset, and reset periods were saved to a separate event file to be used with the data in processing[12].

3. Methods

3.1 Overview of Original Methods

In order to examine the relationship between ERPs in EEG data and semantic violations in reading, an experiment was designed around eliciting ERP responses using sentences containing a semantic violation in the last word. Unfortunately, due to the COVID-19 pandemic, it was not possible to implement this experimental method, but it is described in detail here for potential future applications. Twenty randomly assorted college-aged students, both male and female, would have observed sentence sets presented via rapid serial visual presentation (RSVP) and would have pressed a button to indicate whether or not they believed that the sentence made semantic sense. For the presentation, 60 sentences pairs were created at a middle school reading level for presentation. The sentence pairs can be found in Appendix B of this work. These sentence pairs included one sentence that was normal and would not generally elicit an ERP response, while the second sentence was an exact copy with one word replaced so as to create a semantic violation in the sentence. Each of these sentences was structured so that the semantic violation would fall at the end of the sentence and would always be a noun so as to control for any artifacts that may have arisen with other word types in the presentation. The timing between words during the RSVP protocol was also designed to be slightly randomized to alleviate the effects of a subject anticipating words as they came on screen [41], [42].

EEG data during the experiment would have been recorded using the OpenBCI 3D printed headset with electrodes placed in the international 10-20 format. The OpenBCI Cyton and Daisy boards would have assisted in streaming the live data to the OpenBCI data collection GUI where they could be recorded and monitored in real-time. These data could have then been subsequently migrated to Matlab for more extensive analysis of both the ERPs generated and the alpha and theta components that govern alertness in subjects.

Complete descriptions of the processes used to find subjects, testing environments and equipment, changes to the project due to COVID-19, and data processing methods can be found in the following sections. Section 3.2 details how subjects would have been selected for this project, IRB review and protections for these subjects, as well as subject compensation. Section 3.3 outlines in detail the original setup of the experimental area and what environmental and subject variables we attempted to control for in the experiment. Section 3.4 covers how this project changed due to the COVID-19 pandemic and how this thesis was adapted due to changes in lab accessibility and remote learning. Finally, section 3.5 covers the process used for analyzing collected data and extracting specific ERPs and neural oscillations to assess the feasibility of detecting semantic violations.

3.2 Subject Selection & Compensation for Planned Experiments

Before the beginning of subject selection, all research assistants and those working on the project were required to complete the Biomedical CITI training to obtain IRB approval for research. Research subjects would have completed a pre-experiment survey that would allow for exclusion based on emotional stress, neurological function, and vision as these are critical elements that could potentially skew EEG data. The research excluded the participation of minors as well as other protected groups unable to give voluntary informed consent.

Subject participation in this research would have been completely voluntary and recruiting would be carried out through standard methods such as word of mouth and normal advertising.

Each subject would be required to complete an informed consent document and would have a subject ID assigned to them. These IDs would have then be used when referencing the data and the informed consent forms would have been stored separately in order to retain the anonymity of the test subjects. Testing subjects would have also filled out a survey prior to the experiment that gauged emotional state and exhaustion of the subject, as well as collecting data on other factors such as visual impairment, age, handedness, biological sex, existence of neurological disorder, and comfort with the experimental setting and procedures. After completing the experiment and finishing an exit survey, test subjects would have been compensated with a $25 gift card.

3.3 Testing Environment for Planned Experiments

The testing environment was designed to control for as many outside variables as possible that could have interfered with the collection of clean EEG data. The subject would have sat in a copper mesh enclosure like the one seen in Figure 7 that would act as a faraday cage to isolate the sensitive electrodes within from any outside noise.

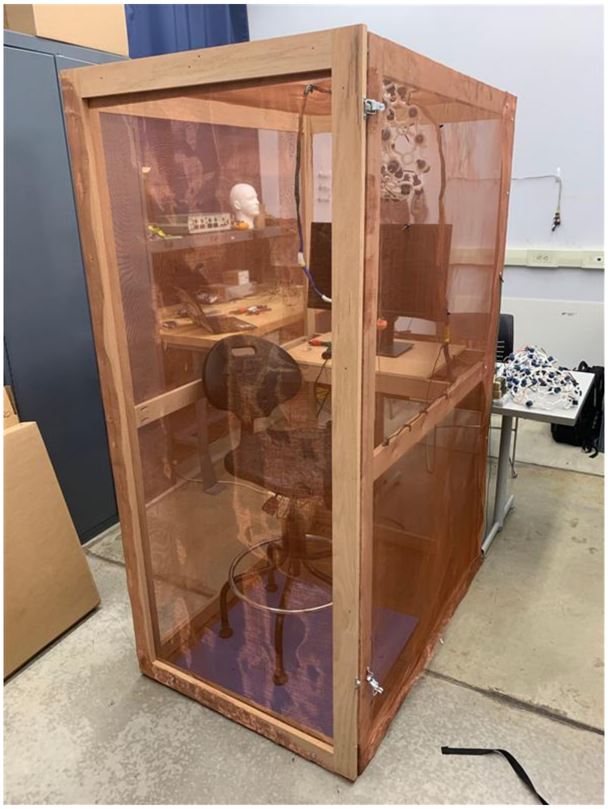


Figure 7. Faraday Cage Setup with monitor system in the BCI research lab. Subjects would have sat in this enclosure while wearing one of the two headsets seen in the photo. The RSVP protocol would have been implemented on the screen in the enclosure with data live streaming to researchers in the room.

The room would have been kept darker to help maintain subject attention on the screen and reduce distractions that may have been present in the environment. The screens that the subjects viewed would be set to a fixed viewing angle, and the RSVP protocol would be adjusted to have the ideal presentation angle on the screen for viewing. Research assistants would have been with the subject at all times of testing to view the data and administer the test, however these assistants would have remained out of view and as quiet as possible to ensure that they did not create a distraction for the research subjects. Additionally, with 120 sentences total, sentences would have been given to the test subjects in blocks of 30 with breaks of 5 minutes in between to minimize the fatigue that subjects may have faced throughout the entire experiment. For each sentence presented, there would have been a period of time after the completion of the sentence where the subject was to press one of two buttons indicating whether the sentence seemed to make semantic sense. Data from each subject would have been streamed to the OpenBCI GUI where it could be recorded and viewed in real-time to monitor for any issues with the headset or controls that could invalidate the experiment.

3.4 Changes in Research Methods

With school and lab closures during the COVID-19 pandemic, the scope and focus of this thesis changed significantly from the procedures outlined above. Because it became impossible to gather data from subjects in real-time, and since the equipment for EEG recording is quite specialized, the project shifted goals and instead focused on analyzing currently available data. For this, we used the data obtained by *Calvo et al.* in their paper "Measuring concept semantic relatedness through common spatial pattern feature extraction on EEG signals." [12]. These data sets were chosen for a variety of different reasons. Firstly the research for which this data was acquired was also aimed at measuring the ability to detect changes in EEG signals based on semantic clues in read text, similar to the initial experiment proposed above for our own research. Secondly, the research methodology established by *Calvo et al.* was similar to the methodology of the original experimentation project. Subjects were university students, the electrode number and placements were the same, the presentation of the word pairs was done with RSVP, and the testing was done autonomously with the subjects using a button to select whether or not they believed the words to be semantically related. Finally, *Calvo et al.* used common spatial pattern filtering to classify their EEG signals based on P300 and N400 components of the waveform. However, they did not extend this classification to include P600, alpha, and beta components, which also hold relation to semantic violations and alertness respectively.

With this in mind, the new aim of this thesis was adjusted to look at whether or not the inclusion of P600, alpha, and beta components of EEG would allow for more accurate detection of semantic violations and act as an improvement over the original data. By adding additional dimensions to the common spatial pattern analysis, we believe that we can further increase the variance between signals with a semantic violation and those without one to better separate these cases from each other. We also aim to look at the feasibility of implementing common spatial pattern classification in EEG research to determine if it is realistic to use this type of classification or if a different type of classification is needed. This aim arose from the fact that *Calvo et al.* achieved high levels of accuracy in their classification despite using common spatial pattern classification on raw data sets in some instances, despite this not generally being a use for common spatial pattern classification. We plan to confirm that the results achieved are not due to errors created by noise introduced in the signal.

3.5 Analysis of Data Provided by *Calvo et al.*

As mentioned before, EEG data from *Calvo et al.* was collected in a very similar manner to the original experimental setup of this project. Light and sound were controlled for in the environment, the experiment was autonomous, and stimuli were presented on a screen in a similar manner with a subject pressing a button to relay whether they believed the words to be semantically related or not. Because the RAW data from this experiment were published publicly online, we were able to access the data to test the ability to further refine the detection protocol laid out by *Calvo et al.* For our analysis, the two of the key components necessary are the timing data and subject responses mentioned in the introduction. A workflow for the entire study can be seen in Figure 8.

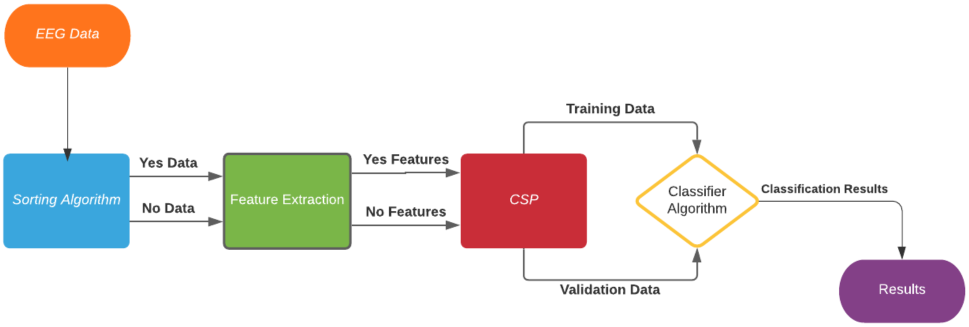


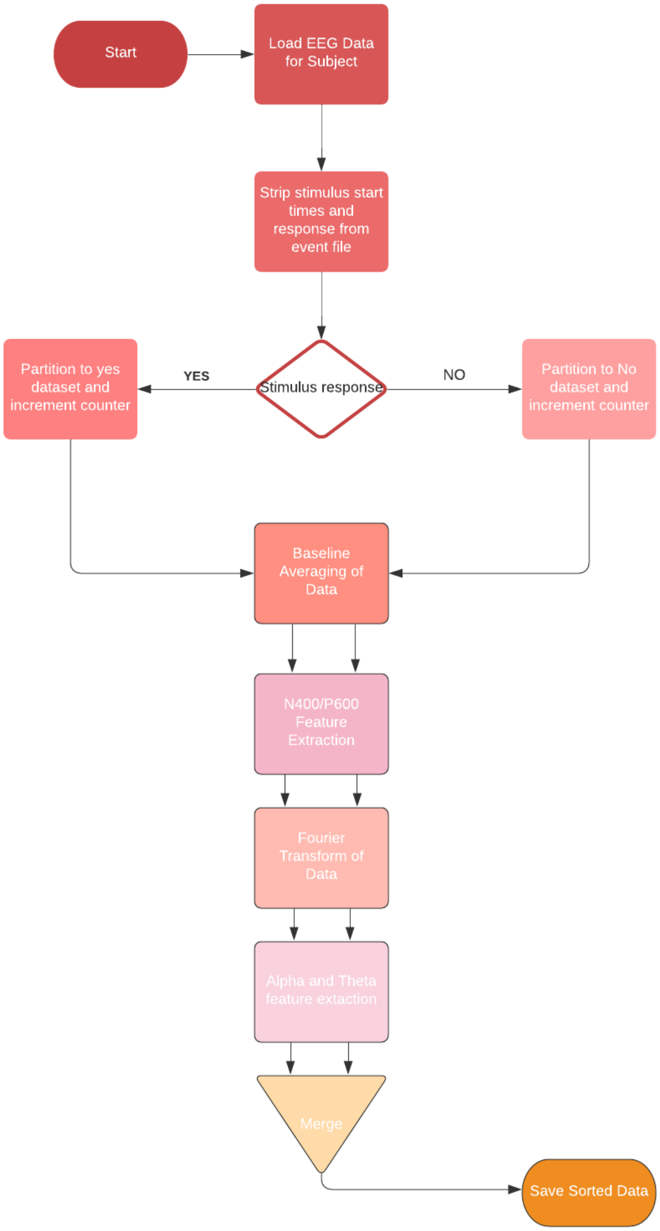
Figure 8. Workflow of the overall thesis process. Steps include sorting and feature extraction, CSP filtering of the extracted features, and classification of CSP results for analysis.

Because the aim of this work is to analyze whether ERP and other EEG components can be used to classify semantic violations a large portion of the collected EEG data is unnecessary. In order to streamline the data used and to isolate the relevant time periods for each stimulus a MATLAB program, found in Appendix C, was created to read in the event file for a given subject and strip off the matrix position where that stimulus began. This was done by creating a separate vector start(k) in a simple for loop that inserts the start time in the third column of every second row into the vector. Furthermore, a simple vector type(k) can be created in the same manner that records the yes and no data recorded in the fifth column of the event file.

Using the start(k) and type(k) vectors, it is possible to sort through each of the five blocks of data and strip out each event. Each of the 14 channels is sorted, taking data from 13 samples before the event onset, and 127 samples after the event onset to capture the window required for normalization and analysis. The channel data were concatenated across all five blocks so that each channel had a dataset consisting of all 150 trials. Additionally, these datasets were split based on whether or not the subject indicated a semantic relation between each word pair. This was determined using the string value in the provided event files and sorting each data set according to that value. Two counters were used in an if statement to increment based on whether a yes or no result occurred.

After all the channels had been sorted, each channel had the first 13 values in each row averaged into a mean to baseline average each signal against the background noise present during the experiment. After baseline averaging of signals, each channel of yes and no data were plotted on a subject by subject basis to look at individual differences in channel amplitude to see if N400 and P600 regions are appreciably different in the data sets. Baseline corrected data were then be separated into two 3D matrices, YesBlock and NoBlock, for the extraction of features from each trial and for the condensing of each feature into the appropriate vector space for the CSP algorithm. The large blocks allow for easy access to each trial stimulus for all 18 subjects, while the N400 and P600 blocks allow for CSP of the time signal. It is notable that N400 takes the average value from 58 to 84 samples of the signal while P600 takes the average value from 90 to 128 samples. These windows correspond with the time period in which the N400 and P600 components should reach their peak respectively and account for the sampling rate of 128 hertz as well as the 13 samples of baseline left in front of each stimulus signal.

For the theta and alpha wave components, the time-series data was converted into the frequency domain with a Fourier transform in order to extract the useful information. Because all the data was partition into the 3D vectors YesBlock and NoBlock, an fft was performed selecting the second dimension to ensure the proper transformation. Finally, using a nested for loop, the double-sided power spectrum of the function was determined and alpha and theta frequency data was ascertained from this power spectrum and sorted into the proper vectors. After this sorting, the overall 14 channel matrices for each feature, N400, P600, alpha, and theta, were plotted in a histogram to compare the average values of the data sets between the yes and no responses. For data sets with a significant overlap of the mean, CSP processing was a good way to make the data more separable for classification. An overall workflow for this data processing can be found in Figure 9 below.



Start
Load EEG Data for Subject
Strip stimulus start times and response from event file
Stimulus response
YES
NO
Partition to yes dataset and increment counter
Partition to No dataset and increment counter
Baseline Averaging of Data
N400/P600 Feature Extraction
Fourier Transform of Data
Alpha and Theta feature extaction
Merge
Save Sorted Data

Figure 9. Workflow for the separation and sorting of EEG data provided by *Calvo et al.*

3.6 Implementation of CSP

After the processing and sorting of the four features was completed, the data was ready to be input into the CSP algorithm. To understand how the CSP algorithm works, an example of CSP with a sample signal has been provided. Given a pair of three-channel EEG signals $\mathbf{M_x}$ and $\mathbf{M_y}$, a CSP algorithm maximizes the variance between these signals when the channels otherwise appear to be inseparable as they are in the data shown in Figure 10.

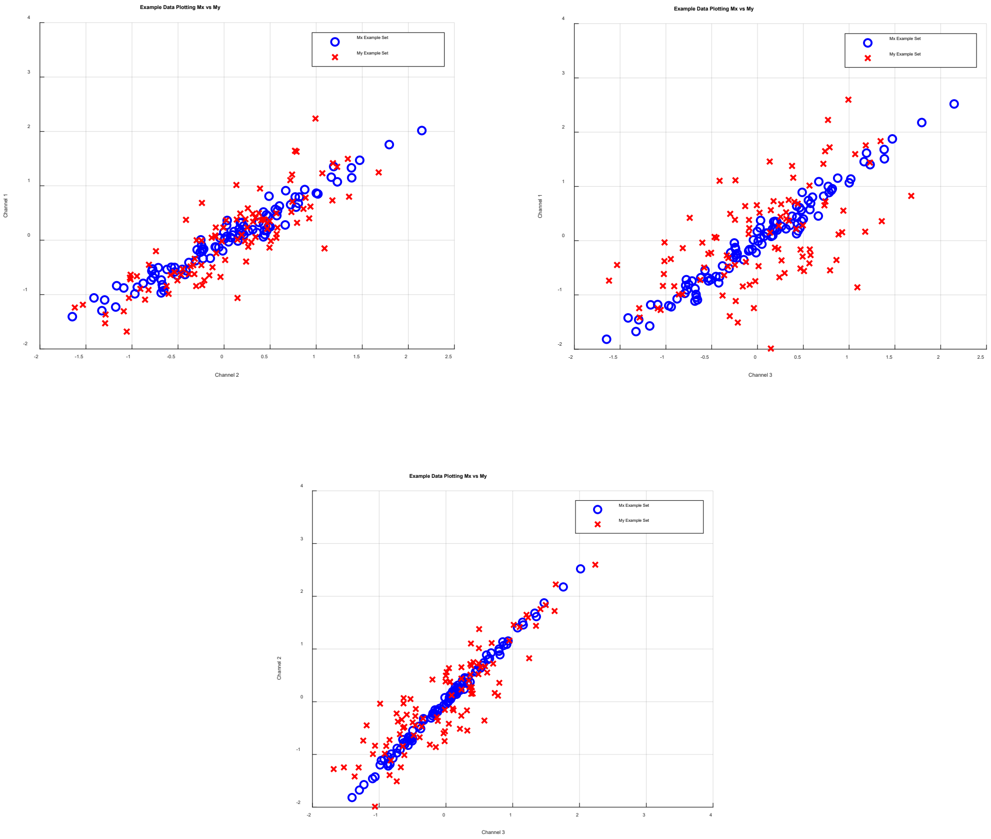


Figure 10. Example plotting of the channels of data M_x *and* $M_{y.}$ *These data have mean values with significant overlap, making CSP a viable option for their separation.*

First, we compute the covariance matrices for each signal and then create a composite of both signals using the following equations:

$$R_x = \frac{M_x * M\prime_x}{\sum_{i=1}^{n} m_{11} + m_{22} + \dots m_{nn}} \tag{1}$$

$$R_{sum} = R_x + R_y \tag{2}$$

This R_{sum} value can be decomposed into eigenvalue and eigenvector components Λ_{sum} and $U_{sum,}$ respectively. When combined with the covariance matrices, a Whitening Transform can be used to create two signals with common eigenvectors S_1 and S_2:

$$W = \sqrt{\Lambda_c^{-1}} U_c \tag{3}$$

$$S_x = W * R_x * W' \quad S_y = W * R_y * W' \tag{4}$$

Where

$$S_x = B\Lambda_x B' \qquad S_y = B\Lambda_y B' \qquad \Lambda_x + \Lambda_y = I \tag{5}$$

In this case, with I as the identity matrix, the corresponding eigenvectors of S_x and S_y will always sum to one. This means that larger eigenvector values in S_x lead to smaller corresponding eigenvector values in $S_{,y}$ which makes the eigenvectors particularly useful for classification in a two-class problem. Using the projection matrix MCSP:

$$MCSP = B' * W \tag{6}$$

each trial can be decomposed into the resulting mapping matrix Z.

$$Z_x = MCSP * M_x \qquad Z_y = MCSP * M_y \tag{7}$$

The final result is a pair of matrices Z_x and Z_y that are mapped orthogonally to one another from the origin, point as illustrated below in Figure 11 [24].

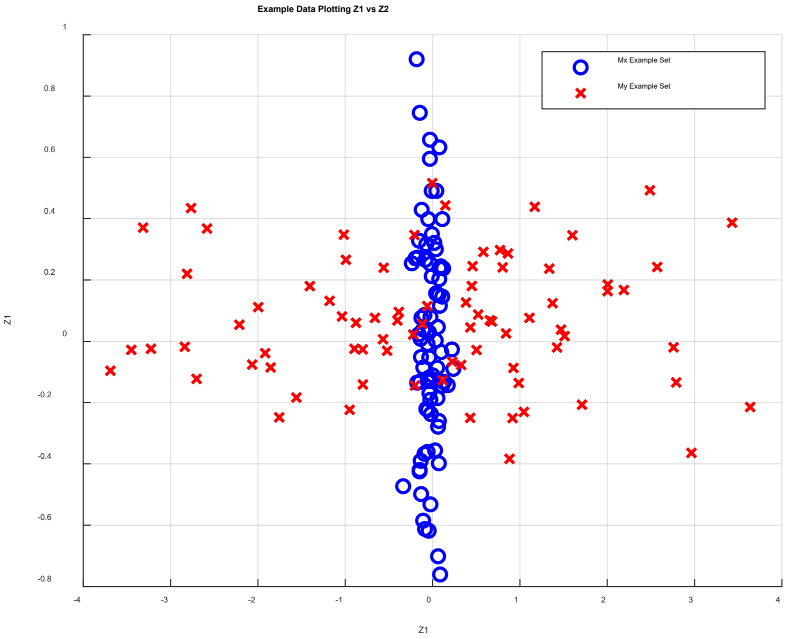


Figure 11. Plot of CSP Transformed Data from M_x and M_y. As opposed to the original data set, this data is orthogonal on the X and Y axis and shows a much greater separation than before.

This result can be replicated with higher dimensional matrices such as the 14xN matrices containing the ERP data for N400 and P600 as well as theta and alpha wave data where N is the number of responses for each class of answer from the subject. For this analysis the variance between yes and no responses for N400, P600, alpha, and theta waves was calculated to determine if any of these variables were sufficiently separable on their own. In addition, these four matrices were also combined into matrices Q and F as 56xN to observe if additional discrimination was possible with a combination of different ERPs and neural rhythms. In addition to looking at the two most distinct eigenvectors generated by the CSP algorithm, the entire eigenvector space was also viewed to determine if the inclusion of more dimensions would allow for better sorting for each data class. The implementation workflow for this CSP algorithm is shown in Figure 12.

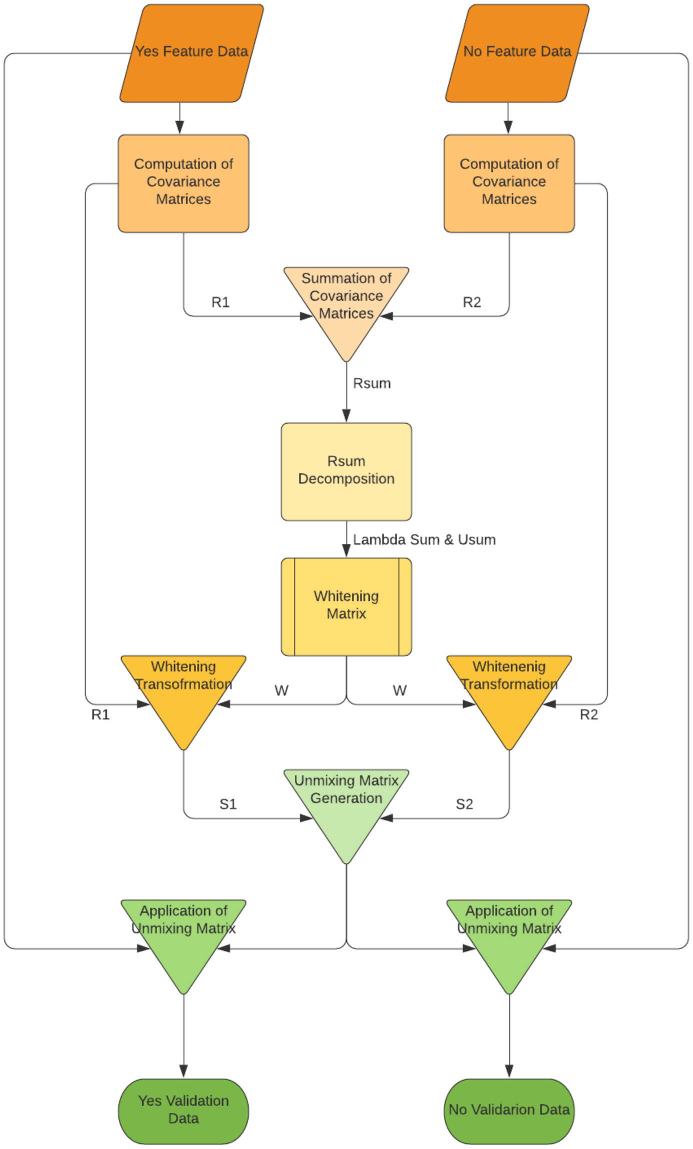

Yes Feature Data
No Feature Data
Computation of Covariance Matrices
Computation of Covariance Matrices
R1
Summation of Covariance Matrices
R2
Rsum
Rsum Decomposition
Lambda Sum & Usum
Whitening Matrix
Whitening Transofrmation
W
W
Whitenenig Transformation
R1
R2
S1
Unmixing Matrix Generation
S2
Application of Unmixing Matrix
Application of Unmixing Matrix
Yes Validation Data
No Validarion Data

Figure 12. Workflow for the implementation of CSP filtering for extracted feature data.

Validation was done by implementing a linear and quadratic discriminant analysis algorithm in MATLAB using the fitcdisc function. Processed data was split into ten sections, and a ten-fold cross-validation method was used to verify the data sorting and check overall separability. For comparison, the data was also classified using the ten-fold cross-validation method with a Naïve Bayes (NB) classifier using the fitcnb function, and a K-Nearest Neighbor (KNN) classifier using fitcknn. After completion of this step, the entire original dataset was band--pass filtered between 0.5 and 40Hz and a CSP algorithm and ten-fold cross-validation was done for this filtered data. The implementation of this validation is illustrated in Figure 13 below.

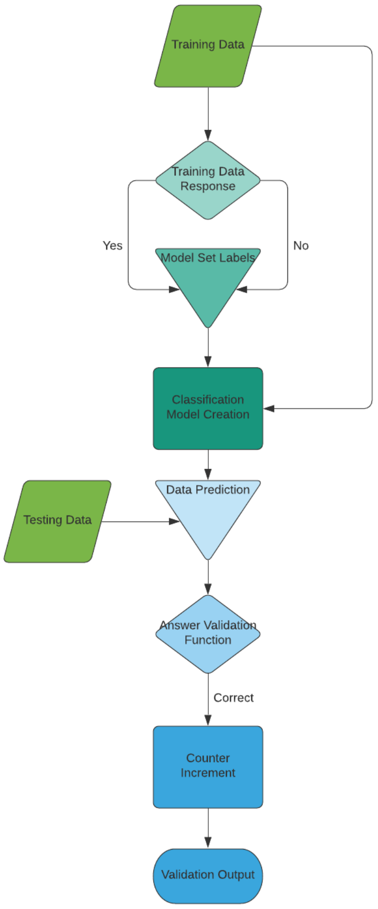


Figure 13. Workflow for the implementation of classifier algorithms on filtered CSP data. Each classifier could be easily adjusted so that all four could be used on the data.

4. Results

4.1 Time-Series Amplitude Analysis

The collected data from each subject was input into a sorting algorithm to separate the data into two classes, yes and no. For each subject, the number of yes and no responses was dependent on their personal trial responses, and as such, the number of answers in each class varies between subjects. After the initial sorting of all 150 trials into two matrices, their time-series was plotted and viewed to determine if separability can be achieved when using singular features such as N400 and P600 ERPs similar to Figure 14 below. While individual trials vary wildly, looking at the average yes and no responses across multiple subjects indicates that there may be features that allow for separation of the data. Averages across the time domain data show differences in response between yes and no classes both across the entire signal and in the N400 and P600 regions. The regions where N400 and P600 ERP peaks are likely to occur are shaded in dark grey, and these ERPs would most typically appear in a signal where the subject indicated that the two words were not semantically related.

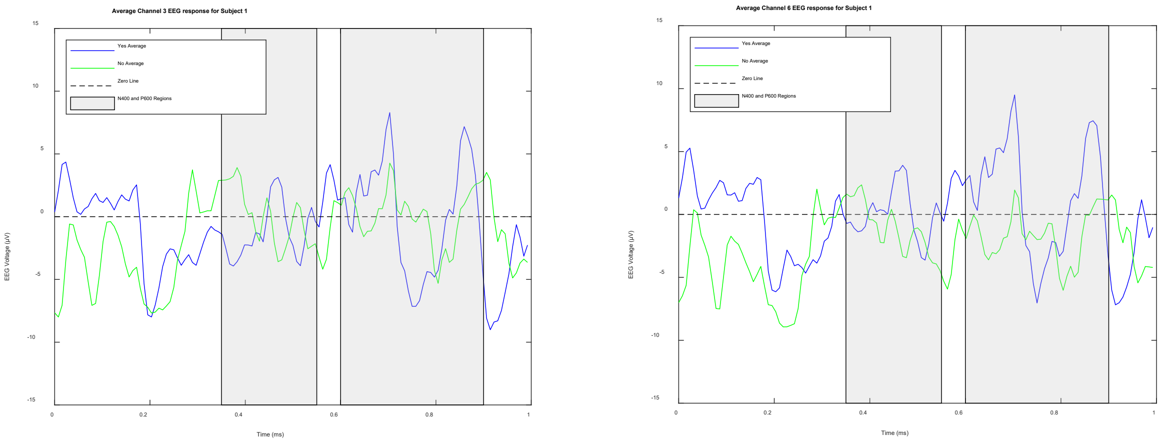


Figure 14: *Illustration of differing amplitudes in the EEG signals across multiple channels for one subject. There is wide variability in the response, but it is clear that yes and no responses have some differences. Some shaded regions also appear to show deflections that are possibly indicative of the N400 and P600 ERPs.*

In the averaged signals, it is possible to see some deflections in the N400 and P600 ranges that could possibly be attributed to a semantic response in the no category. However, while the time-series data shows differences in the signals depending on the subject response, this does not necessarily indicate that the data will be easily separable outright as there are significant areas of overlap in many data sets. In order to test whether or not the data contained in the time domain can be used for classification, a simple comparison of the mean of each feature can be made to determine their overlap.

4.2 Histogram Mean Analysis

One of the simplest methods available of separating two classes of data is the comparison of their mean. Using histograms, it is possible to compare the response values across all trials to determine if the mean is separable enough to be considered for classification purposes, as shown in Figure 15. For this study, all of the subject histograms displayed relatively similar results that indicate the data is inseparable.

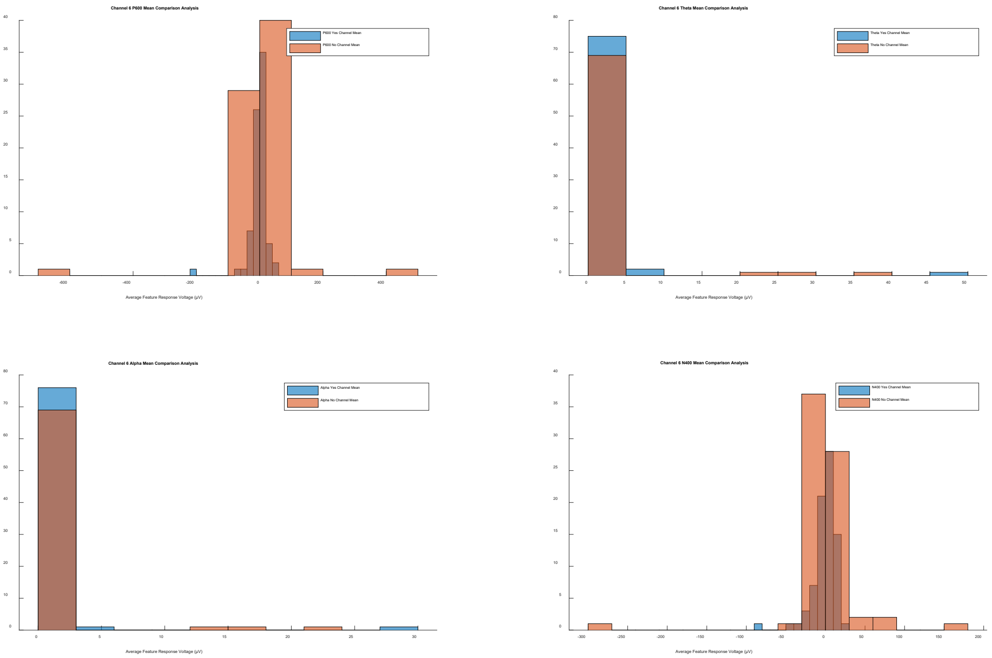


Figure 15: *Histogram of average voltage values of four separate features from Subject 1 Channel 6. The overlap in mean of these features shows that they are inseparable by mean and will likely need to be processed in a way that ignores the data mean.*

Across all subjects, channels, and features, the average value of each data set had significant overlap, which indicates a relatively similar mean for the data. This precluded the use of classification methods that depend on the separation of the mean of two data sets. However, a method that performs a transform of the data is still a viable option for classifying these two data sets. After band-pass filtering the data from 0.5 - 40Hz the mean of the two data sets remains inseparable as indicated by Figure 16, though the filtered data does show better overlap with the removal of signal noise.

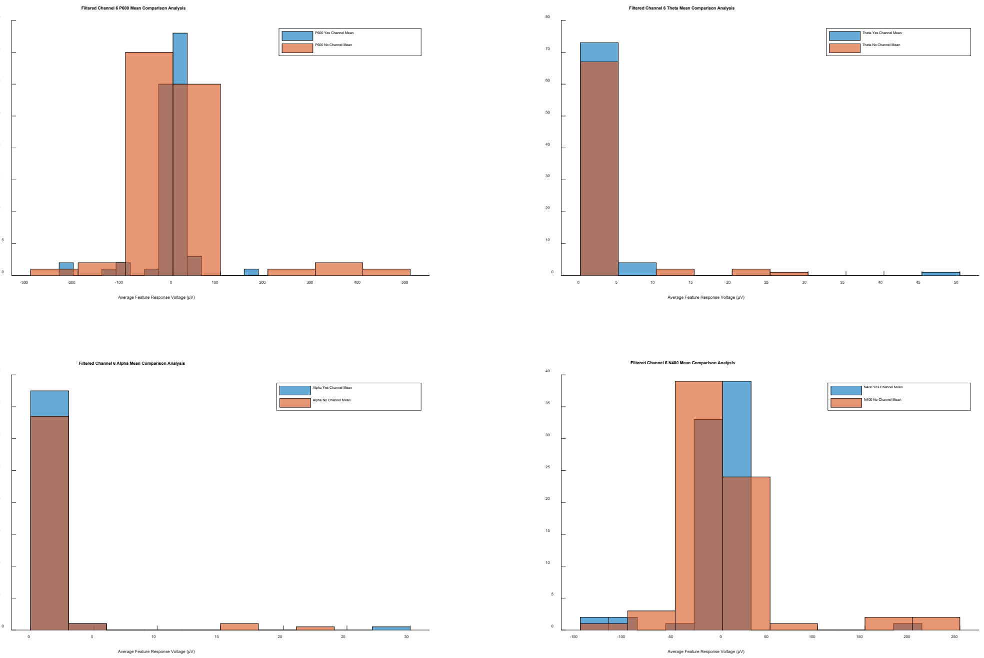


Figure 16: *The filtered data for Subject 14 still has an inseparable mean, which indicates that the mean overlap was likely not due to additional signal noise.*

4.3 Common Spatial Pattern Analysis

After verifying the inseparable means of the single feature RAW and filtered EEG data, CSP became a good option for maximizing the variance between the datasets as it allows data with overlapping means to be separated by projecting them orthogonally against each other. Each of the four data features for each subject was run through the CSP algorithm to determine whether or not a single feature could be used to classify the data. These results are illustrated graphically below in Figure 17 using the two features of greatest variance for each feature.

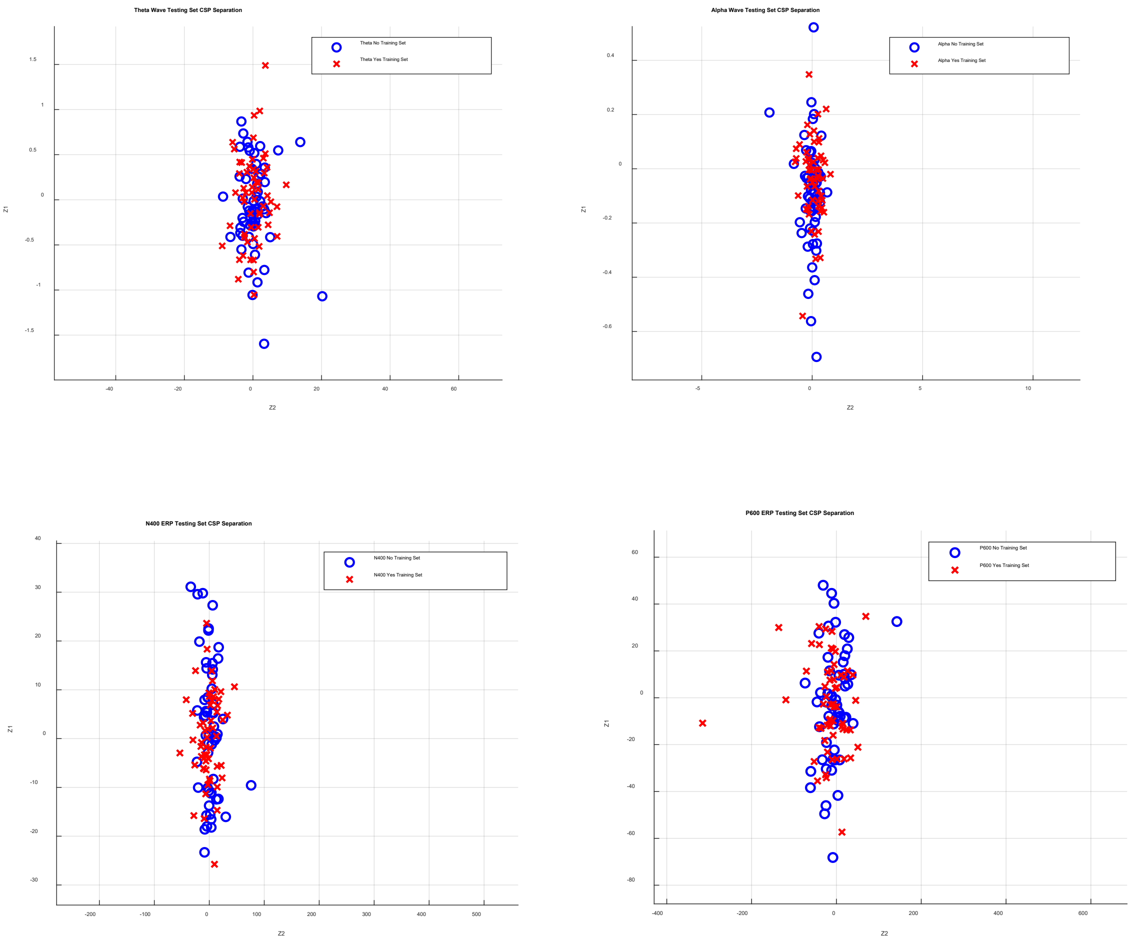


Figure 17: *The four features selected for analysis are all shown here to be individually inseparable. To create separation with these features, they will need to be combined into a higher dimensional dataset.*

For these four features, CSP could not generate appreciable separation and variance between the two data classes, which indicates that each feature alone doesn't provide enough granularity for response classification. Running validation tests using the unmixing matrix generated by these cases verifies that the single features are not separable enough by themselves, as illustrated in Figure 18. These trends hold true for both the RAW EEG data set and the filtered EEG data set.

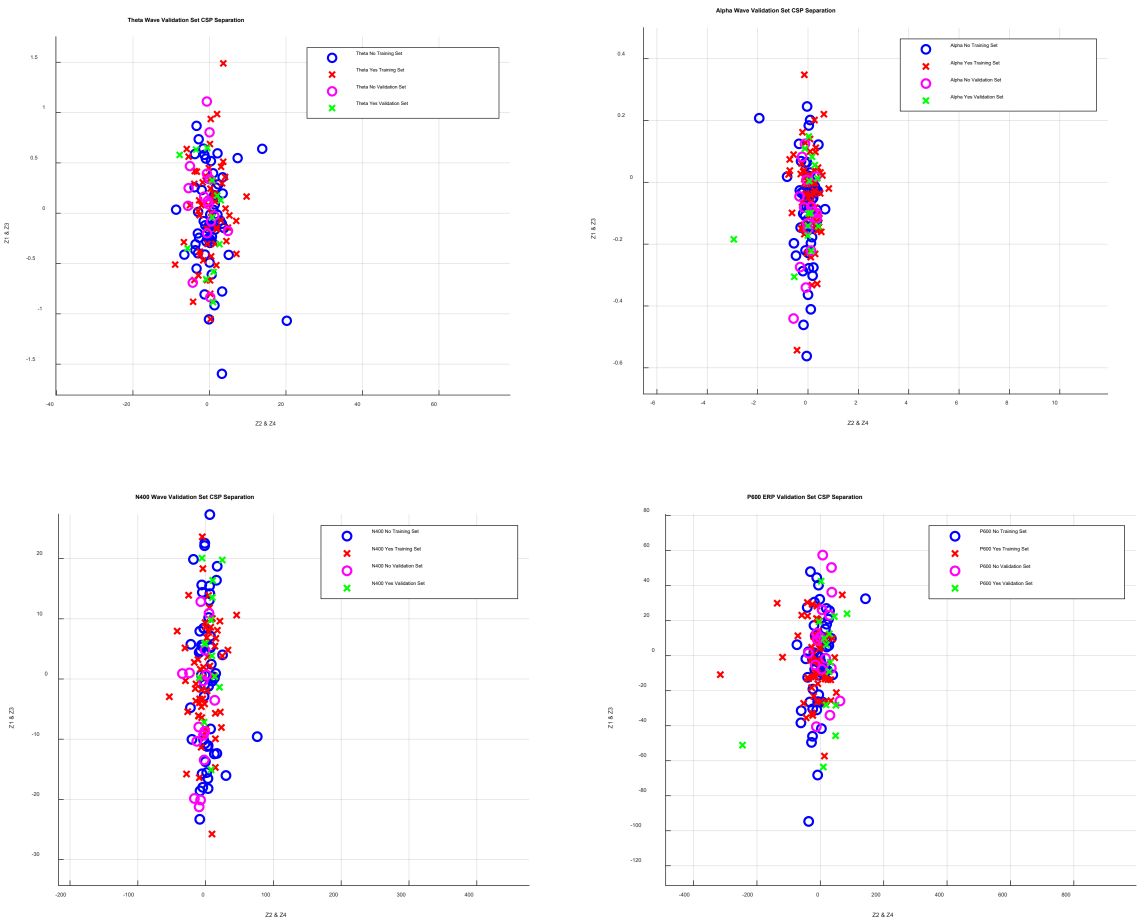
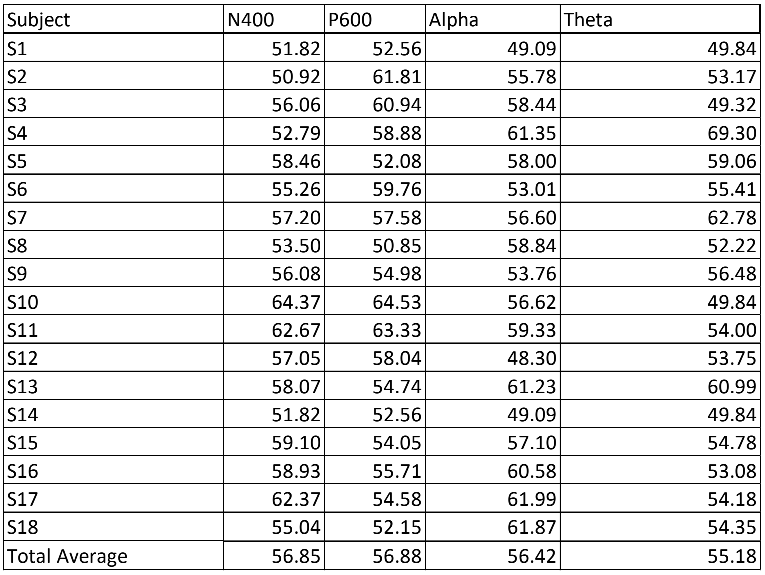


Figure 18: *Validation data overlayed onto the CSP transformation confirms that this data is still linked. Each feature validation set is clumped within the general mean of the training set.*

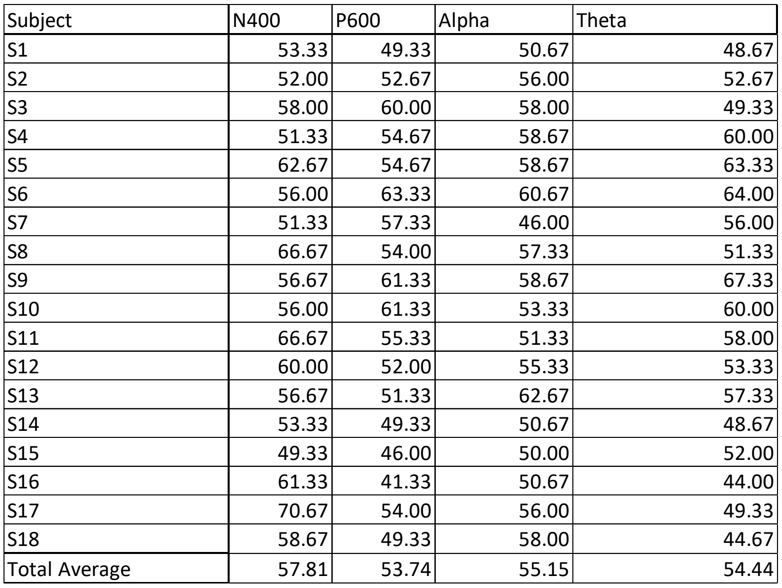
Tables 1 and 2 below show the results of using both Naïve Bayes and K-Nearest Neighbor classifiers on each distinct feature. While one subject did reach a classification rate as high as 70% with single feature classification, the overall trend for most subjects was only a moderate increase over random chance.

Table 1. Naïve Bayes Analysis with Ten-Fold Validation for all 18 research subjects. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation. Each subject had the four individual features of interest classified with this method, with Subject 4 having the highest classification rate at 69.30%. Overall rates indicate single features are not robust enough for accurate classification.



Subject	N400	P600	Alpha	Theta
S1	51.82	52.56	49.09	49.84
S2	50.92	61.81	55.78	53.17
S3	56.06	60.94	58.44	49.32
S4	52.79	58.88	61.35	69.30
S5	58.46	52.08	58.00	59.06
S6	55.26	59.76	53.01	55.41
S7	57.20	57.58	56.60	62.78
S8	53.50	50.85	58.84	52.22
S9	56.08	54.98	53.76	56.48
S10	64.37	64.53	56.62	49.84
S11	62.67	63.33	59.33	54.00
S12	57.05	58.04	48.30	53.75
S13	58.07	54.74	61.23	60.99
S14	51.82	52.56	49.09	49.84
S15	59.10	54.05	57.10	54.78
S16	58.93	55.71	60.58	53.08
S17	62.37	54.58	61.99	54.18
S18	55.04	52.15	61.87	54.35
Total Average	56.85	56.88	56.42	55.18

Table 2. K-Nearest Neighbor Analysis with Ten-Fold Validation for all 18 research subjects. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation. Each subject had four individual features of interest classified separately, with subject 17 having the highest classification rate at 70.67%. While this rate was considerably greater than random chance, it also appears to be a one-off anomaly in the data, though it could indicate the strong semantic responses present during semantic violations.



Subject	N400	P600	Alpha	Theta
S1	53.33	49.33	50.67	48.67
S2	52.00	52.67	56.00	52.67
S3	58.00	60.00	58.00	49.33
S4	51.33	54.67	58.67	60.00
S5	62.67	54.67	58.67	63.33
S6	56.00	63.33	60.67	64.00
S7	51.33	57.33	46.00	56.00
S8	66.67	54.00	57.33	51.33
S9	56.67	61.33	58.67	67.33
S10	56.00	61.33	53.33	60.00
S11	66.67	55.33	51.33	58.00
S12	60.00	52.00	55.33	53.33
S13	56.67	51.33	62.67	57.33
S14	53.33	49.33	50.67	48.67
S15	49.33	46.00	50.00	52.00
S16	61.33	41.33	50.67	44.00
S17	70.67	54.00	56.00	49.33
S18	58.67	49.33	58.00	44.67
Total Average	57.81	53.74	55.15	54.44

To generate a dataset that would be more separable than any single feature, all four features were combined into a single feature space. This new feature space comprising fourteen channels from four features has a total of 56 dimensions for analysis. With this new higher dimensional data set, it was possible to obtain separation between the two classes for all 18 like that shown in Figure 19. The data here were highly orthogonal, and it was clear that each class was aligned on a separate axis.

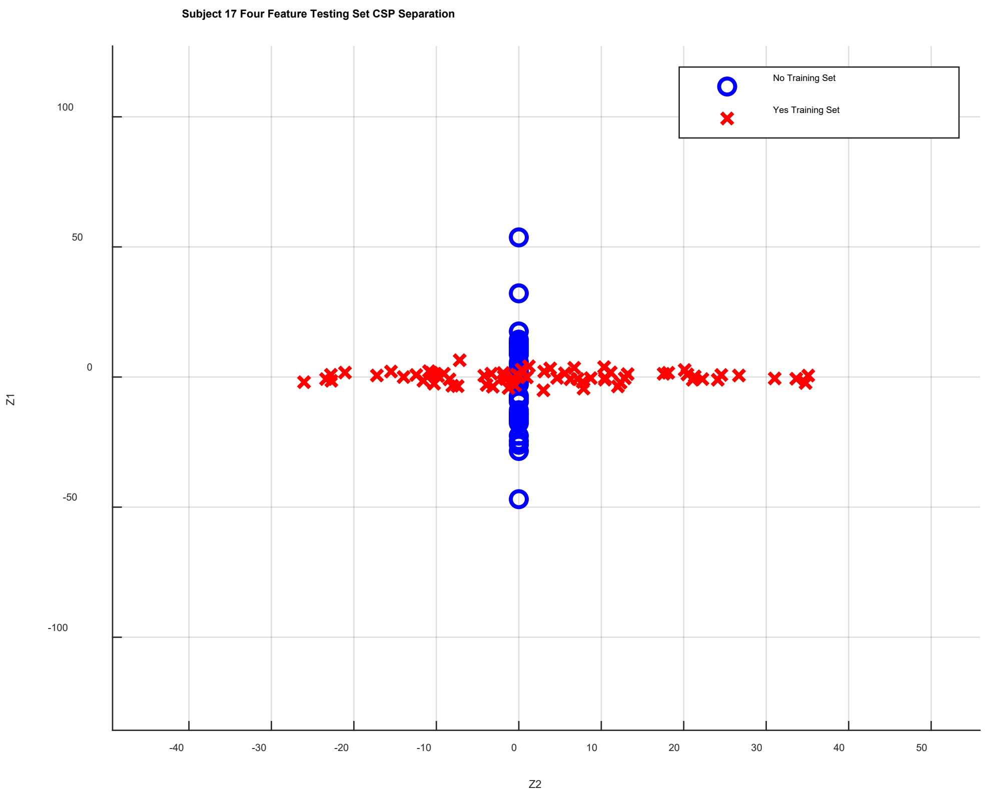


Figure 19: *The four features combined create two data sets that can be projected orthogonally to each other after CSP processing for each individual subject.*

Using the unmixing algorithm to project the validation data set onto the subject set shows more promising results than the single figure trials. The validation set projection in Figure 20 shows a clear separation from the mean when plotted over the training set.

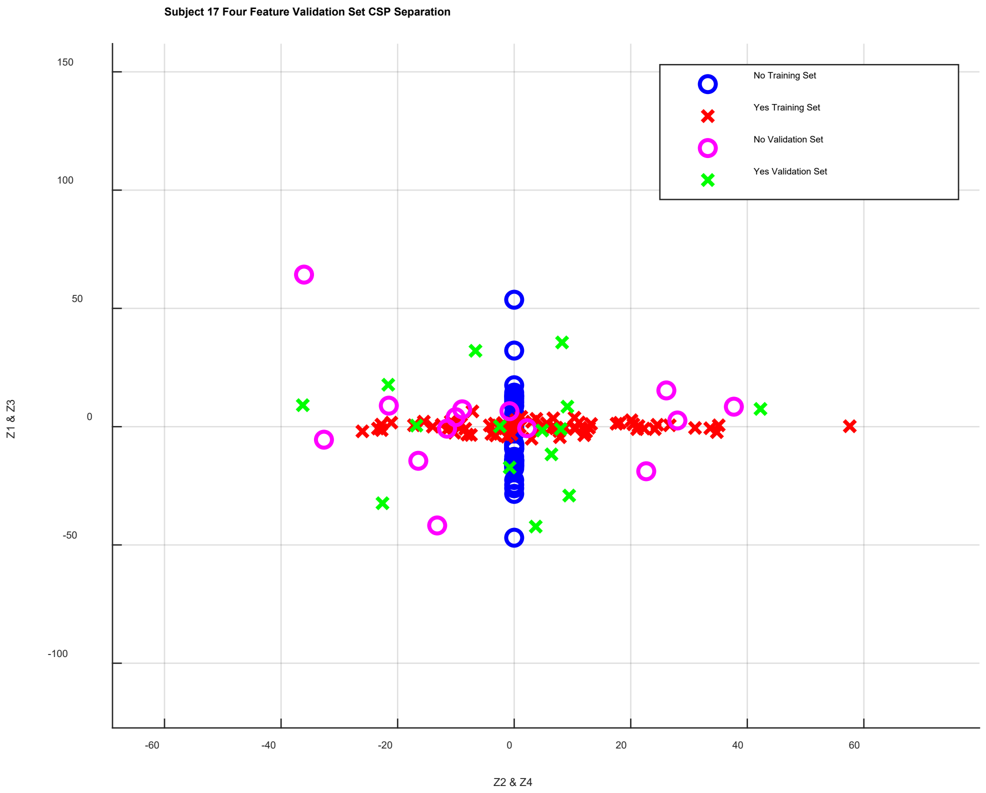


Figure 20: *Validation set data plotted over the original CSP transformation shows that an unmixing matrix generated from this transform does separate the data to some degree for independent subjects.*

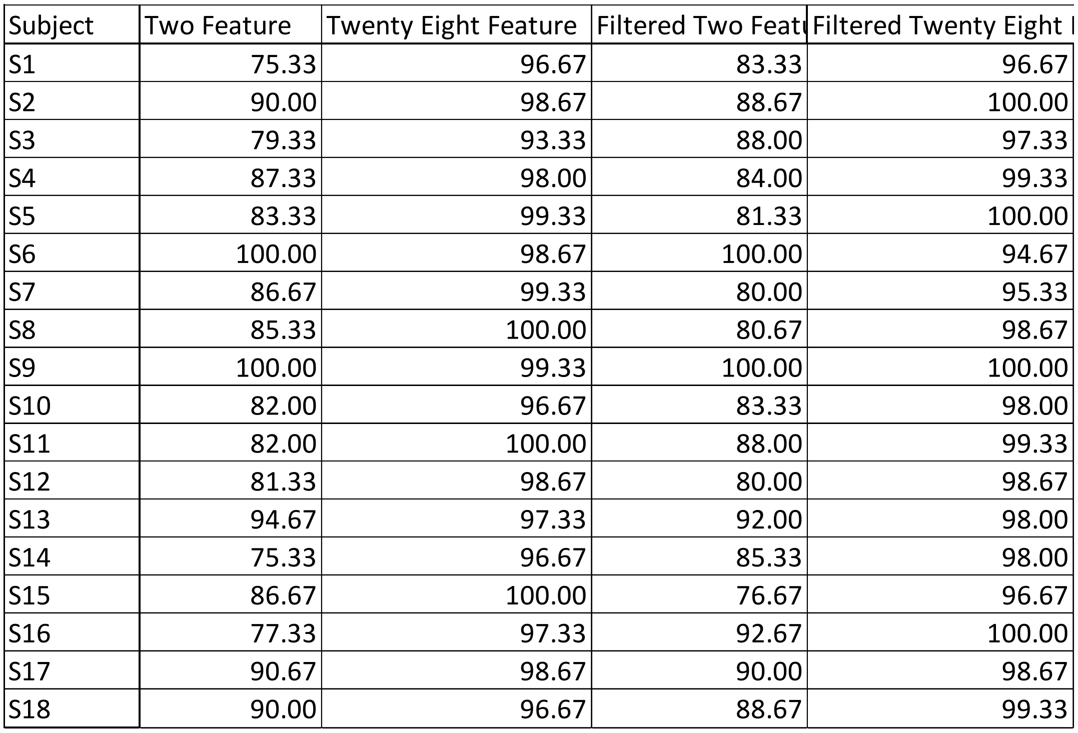
For higher-dimensional sets, it was not possible to look at the spread of data to visually assess whether or not separation and proper classification occurs during the validation step, so instead, a linear discriminant algorithm was created to classify the results of each set. This algorithm was used in a ten-fold cross-validation of all 18 subjects, and it allowed for the examination of 2, 28, and 56 features of RAW and filtered data for each subject. After the ten-fold validation process the average value and maximum value of the correct answers were found for each subject. The values shown in Table 3 indicate the classification percentage for LDA for the given feature sets.

Table 3. Linear Discriminant Analysis with Ten-Fold validation for all 18 subjects and six feature groupings. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation. LDA was overall an ineffective method at classifying the data, performing worse in some cases than using single features for classification.

Subject	Two Feature	Twenty Eight Feature	Fifty Six Feature	Filtered Two Feature	Filtered Twenty Eight Feature	Filtered Fifty Six Feature
S1	66.67	65.33	54.00	50.67	57.33	46.00
S2	50.00	62.00	54.00	50.67	60.00	47.33
S3	53.33	58.00	48.00	56.00	60.00	50.00
S4	51.33	58.67	41.33	47.33	52.00	43.33
S5	53.33	51.33	48.00	43.33	52.67	52.00
S6	62.67	61.33	50.67	64.00	63.33	46.00
S7	59.33	64.67	51.33	58.00	54.67	48.00
S8	47.33	52.67	40.00	49.33	50.00	39.33
S9	66.67	64.67	55.33	65.33	66.00	52.67
S10	43.33	47.33	42.67	52.67	50.67	42.67
S11	49.33	60.67	59.33	52.00	61.33	50.00
S12	50.67	60.00	54.00	54.00	59.33	49.33
S13	59.33	62.67	51.33	54.00	64.67	52.67
S14	66.67	65.33	54.00	56.67	68.67	79.01
S15	56.67	68.67	52.67	58.67	44.67	41.33
S16	62.00	50.00	38.00	56.00	50.00	43.33
S17	56.00	66.67	54.67	57.33	65.33	49.33
S18	59.33	56.00	57.33	48.00	60.00	48.00

Overall, LDA does not appear to be an effective way to classify data that has been modified through CSP. Subjects 1 and 6 had the highest classification rate at 66.67% with two features using RAW data, while many other subjects barely achieved classification at a rate higher than random chance. However, while linear discriminants were ineffective, quadratic discriminants showed much better results, as outlined below in Table 4.

Table 4. Quadratic Discriminant Analysis with Ten-Fold validation for all 18 subjects and four feature groupings. Due to the nature of CSP, the 56 feature covariance matrices become too similar to perform QDA on these sets. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation.



Subject	Two Feature	Twenty Eight Feature	Filtered Two Feat	Filtered Twenty Eight
S1	75.33	96.67	83.33	96.67
S2	90.00	98.67	88.67	100.00
S3	79.33	93.33	88.00	97.33
S4	87.33	98.00	84.00	99.33
S5	83.33	99.33	81.33	100.00
S6	100.00	98.67	100.00	94.67
S7	86.67	99.33	80.00	95.33
S8	85.33	100.00	80.67	98.67
S9	100.00	99.33	100.00	100.00
S10	82.00	96.67	83.33	98.00
S11	82.00	100.00	88.00	99.33
S12	81.33	98.67	80.00	98.67
S13	94.67	97.33	92.00	98.00
S14	75.33	96.67	85.33	98.00
S15	86.67	100.00	76.67	96.67
S16	77.33	97.33	92.67	100.00
S17	90.67	98.67	90.00	98.67
S18	90.00	96.67	88.67	99.33

QDA heavily outperforms LDA, with all subjects achieving high levels of classification accuracy. These results are very competitive with the results achieved by Naïve Bayes and K-Nearest Neighbor classification found below in Tables 5 and 6, respectively.

Table 5. Naïve Bayes with Ten-Fold validation for all 18 subjects and six feature groupings. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation. For higher-level feature sets NB appears to have extraordinarily accurate classification.

Subject	Two Feature	Twenty Eight Feature	Fifty Six Feature	Filtered Two Feature	Filtered Twenty Eight Feature	Filtered Fifty Six Feature
S1	76.55	100.00	100.00	82.75	98.61	99.31
S2	90.75	100.00	100.00	88.14	100.00	100.00
S3	79.65	99.25	99.25	88.75	100.00	100.00
S4	87.01	100.00	100.00	84.33	100.00	99.28
S5	83.75	99.30	99.30	81.71	100.00	100.00
S6	100.00	100.00	100.00	100.00	100.00	100.00
S7	89.08	100.00	100.00	85.00	100.00	100.00
S8	85.83	100.00	100.00	80.44	100.00	100.00
S9	100.00	100.00	100.00	100.00	100.00	100.00
S10	83.07	99.31	99.31	84.24	100.00	100.00
S11	82.00	100.00	100.00	88.00	100.00	99.33
S12	81.70	100.00	100.00	80.63	100.00	100.00
S13	93.97	100.00	100.00	93.39	98.44	100.00
S14	76.55	100.00	100.00	85.40	99.34	100.00
S15	88.37	100.00	100.00	79.25	100.00	100.00
S16	77.53	100.00	100.00	92.21	100.00	100.00
S17	91.04	100.00	100.00	93.59	100.00	100.00
S18	90.22	98.03	98.73	88.11	99.37	99.37

Table 6. K-Nearest Neighbor with Ten-Fold validation for all 18 subjects and six feature groupings. Values in the table represent the percentage of correct classifications based on the classification error of the 10-fold validation. The accuracy of KNN appears to increase for 28 features but sharply declines when using all 56 features.

Subject	Two Feature	Twenty Eight Feature	Fifty Six Feature	Filtered Two Feature	Filtered Twenty Eight Feature	Filtered Fifty Six Feature
S1	86.67	90.00	55.33	88.67	84.00	44.67
S2	91.33	88.00	50.00	83.33	94.00	56.67
S3	79.33	84.00	50.67	85.33	92.00	54.67
S4	84.67	90.67	43.33	85.33	88.67	49.33
S5	78.67	92.67	57.33	77.33	93.33	56.67
S6	98.00	82.00	60.00	95.33	91.33	64.67
S7	78.67	83.33	61.33	90.67	90.00	53.33
S8	86.00	92.67	52.00	85.33	92.00	45.33
S9	89.33	92.67	58.67	83.33	90.67	62.67
S10	82.67	84.00	48.00	81.33	84.00	47.33
S11	86.67	92.67	56.67	86.00	88.67	56.67
S12	88.67	89.33	56.00	77.33	90.00	48.67
S13	91.33	89.33	60.67	92.00	91.33	61.33
S14	86.67	90.00	55.33	84.67	89.33	70.99
S15	82.67	85.33	61.33	79.33	87.33	48.67
S16	79.33	88.67	50.67	92.00	91.33	42.67
S17	88.00	92.67	58.00	90.00	88.00	65.33
S18	85.33	87.33	60.00	84.00	88.67	53.33

Looking across all 18 subjects and four classifications methods with two features, KNN was the best method for 3 subjects, filtered KNN was the best method for 2 subjects, NB was the best method for 2 subjects with filtered NB being best for 4. QDA was the best classifier for 2 while filtered QDA was the best method for 2. Finally, there was no subject for which LDA or filtered LDA was the best classifier. The remaining slots were all ties that went between two or more different classification methods. These results can be seen in Figure 21 below. Figure 22 contrasts the results in this study with those achieved by *Calvo et al.*.

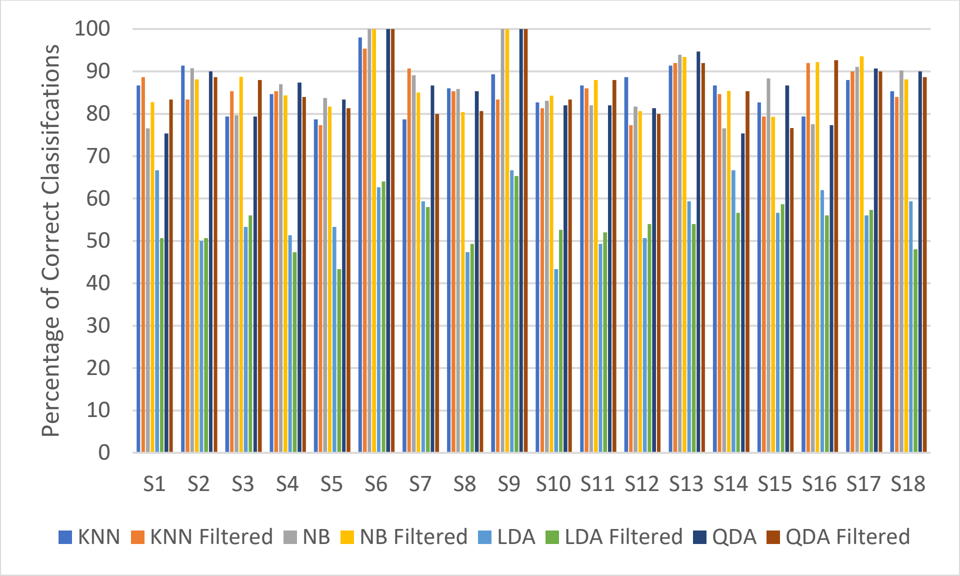


Figure 21: *Classification percentages show that all of the cases except LDA provide a significant increase over random chance in detecting possible semantic errors in reading.*

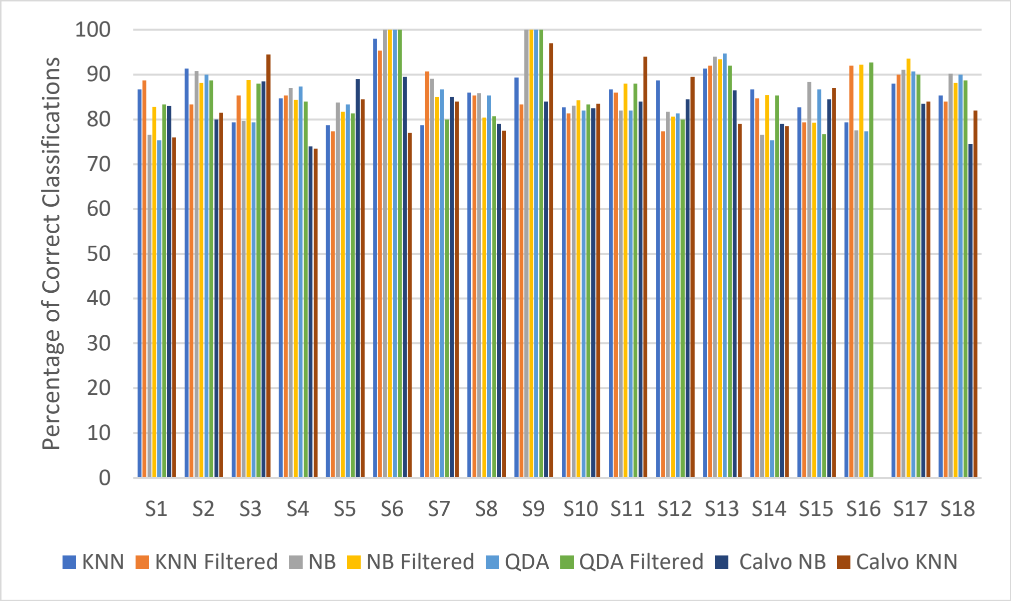


Figure 22: *Comparison of best classifiers with the best results obtained by Calvo et al. The methods and classifiers outlined in this paper outperform the Calvo NB and KNN in 12 out of 17 cases. Subject 16 is excluded as this subject was not included in the data from the original paper.*

5. Discussion

The primary goal of this research was to examine the viability of using CSP to classify semantic responses using N400, P600, theta and alpha waves. Increasing the accuracy of a BCI is important in improving the quality of life for the end-user, and an algorithm that would allow for the easy detection and correction of semantic violations would be a significant boon for those living with advanced disabilities. Based on preliminary research presented by *Calvo et al.* it appeared promising that CSP could allow for high accuracy classification of semantic violations even though CSP is not generally used in conjunction with event-related potentials. The findings given here appear to support the results by *Calvo et al.*. Ten-fold validation of CSP data provided by all 18 subjects shows that for many different combinations of features, CSP and classification can properly sort the results at a rate significantly better than random chance. Classification accuracy does seem to be related to the subject, as different subjects have classification accuracies as different as twenty percent.

To gauge the effectiveness of CSP, the original dataset was partitioned into 150 segments and then each segment had four features extracted from it. These four features were then put into a CSP algorithm to see if they were individually separable. Some of the features did have separability and classification at a slightly higher rate than random chance, however, this rate was not enough to be useful. This does indicate, however that many of these different features may play a small part in categorizing semantic violations as they occur. A combination of these different features responsible for categorizing semantic responses led to a very separable dataset with high classification accuracy. With the combination of four features with 14 channels of data each, the final result was a 56 feature matrix for use in the CSP algorithm. However, the use of all 56 features does not provide an increase in performance for all classifiers, and for the case of KNN 56 features actually reduced the effectiveness of the classifier. This may be due to the fact that all 56 feature eigenvalues are ordered from smallest to largest in CSP, and as you approach the center, the eigenvalues become similar in magnitude and, therefore, less separable. Adding in these similar magnitude eigenvalues likely decreases the effectiveness of the classifier algorithm as it creates an unmixing function where similar data values for both classes begin to overlap.

It was believed that the orthogonality gained from CSP would allow for the use of LDA, QDA, NB, or KNN for classification. LDA was ineffective as a classifier as it was unlikely to find a bound between the data that would not split both datasets in half. QDA, on the other hand, was able to create quadratic surfaces that encapsulated each class of data and led to high accuracy classification. Across all 18 subjects there was no singular greatest classifier, KNN worked best for 3 subjects, filtered KNN for 2 subjects, NB for 2 subjects, filtered NB for 4 subjects, QDA for 2 subjects, and filtered QDA for 2 subjects, with LDA and filtered LDA remaining as the best classifier for no subjects. Aside from LDA, all subjects regardless of classification type were able to achieve results appreciably surpassing random chance, indicating that with proper feature selection, CSP may be a method that is useful for a wide variety of subjects. When comparing to *Calvo et al.*, the methods outlined here outperform their CSP method for 12 out of 17 subjects with subject 16 excluded, though it is unclear how many features and dimensions *Calvo et al.* analyzed in their work. Additionally all of these cases were completed with ten-fold validation instead of the random single sampling completed by *Calvo et al.* Additional analysis of the unmixing algorithms showed that the electrodes responsible for the highest weighted eigenvalues were P7, F4, F3, AF4, and FC6. F3 and P7 are of particular interest because these two electrodes are located in the general vicinity of Broca’s area and Wernicke’s area, respectively, areas of the brain associated with language comprehension and function. The remaining three electrode locations can be found on the frontal right hemisphere, which could indicate the use of some problem solving and motor function impulses that allowed each subject to select and press the button to decide the class of each stimulus.

Despite the promising data results, there are a few different issues that still need to be addressed within the dataset and analysis. Firstly, the analysis achieved a high rate of success when classifying the provided EEG data within the stimulus range, however, three of the electrode locations that were important in creating separation in CSP are also associated with motor planning and function. This indicates that it is probable that these locations were responding to a subject's preparation and pressing of the button indicating whether the groups were semantically related or not, as opposed to a response based solely on semantic reactions to the word groupings. Further experiments looking to control for this would be well served by finding a different way to label these sentences outside of the collected EEG data. Some alternatives could include running the test twice, once without buttons and once with them to obtain labels and compare the two datasets. Secondly, another issue with this analysis is that it is based on averaging time windows to obtain singular values for N400 and P600 features. While this is a general way in which this feature analysis is done, different methods could better preserve the feature data and lead to higher quality feature extraction. Using a matching algorithm for N400 and P600 features or using a wavelet sweep across the signal may not only preserve more of the critical data points of each feature, but it may also help to mitigate errors that occur if an N400 or P600 peak has a quicker or more delayed onset.

There are a few different possibilities that could explain away the differences between the data presented here and the results provided by *Calvo et al.* Firstly, *Calvo et al.* looked at a classification based on ERPs alone, looking at the N400 region and P300 region, while this thesis examined classification based on a mixture of ERP and neural rhythms including N400, P600, alpha, and theta rhythms. While the addition of additional features should not negatively affect the CSP algorithm, as CSP allows for the selection of only dimensions providing the highest variance, using different ERPs for analysis could have an effect on the final result. Numerically the methods shown here appear to increase classification performance, however a direct comparison with *Calvo et al.* is still difficult for a variety of reasons. Firstly, it is not clear how *Calvo et al.* isolated their N400 and P300 components. This research handled N400, P600, alpha, and theta components by averaging the data points in the ranges where these features typically occur in order to get single average values for each trial. Differences in how these features were isolated could lead to potentially different results between studies. Secondly, it is unclear how *Calvo et al.* implemented their CSP filtering from their paper. An important part of the CSP algorithm is determining how many features are being examined from the overall set of features, as shown here using 2 features and using 56 features can lead to dramatically different results based on the classification method being used. With a more detailed overview of how data analysis and classification were done, it may be possible to understand how the two experiments differed and pinpoint discrepancies in the data processing. This would also aid in better supporting CSP as a possible method for use outside of only neural oscillations.

Based on these findings, it is clear that more research needs to be done to understand the full extent to which CSP using ERPs can be used in the classification of semantic responses. *Calvo et al.* presented strong results suggesting that CSP is a potent tool in semantic violation detection with ERPs, while this study supports the idea that CSP may have beneficial implications in classifying semantic relationships in reading. It will be vital that future studies clearly outline what methods are used for isolating features for CSP, how many CSP features are used, how these features are extracted from the EEG signal, what types of classifications methods are used, how these methods are implemented, and what experimental conditions data was collected in. This uniformity will allow for a clear verdict on the application of CSP technology and will also improve the functionality of this technology in real-world settings. By standardizing the methods used in this research, it will become increasingly easier to determine whether results are realistic or skewed due to other unforeseen mitigating factors.

As with any experiment involving human subjects, many of the results were highly variable and dependent on the specific subject. Subjects may achieve different results based on the mood they were in during testing, their relaxation in their testing environment, their ability to remain focused throughout the experiment, and even simply due to the placement of the electrodes on their head. In addition, since every subject had a different number of yes and no responses, it is quite possible that there are confounding words circumstances where subjects answered that words were classified differently than they may generally classify them in their head, as some subjects heavily favored yes answers while others heavily favored no answers. Additionally, as this experiment took place in Spanish with translations of the original words, there may be specific rhyming schemes or other associations that are not translated well and could cause erroneous answers. There may also be different modes of semantic processing that come into play when a word is found in a sentence as opposed to when it is only compared to another word. Future experiments that observe the differences when these violations occur in conversational sentences may achieve better results and will likely be more applicable to BCI users.

6. Limitations

The clearest and most obvious limitations of this research were those imposed by the COVID-19 pandemic. With school closings and students being sent home, it was then impossible to continue the research as originally planned and it became difficult to assess many of the research goals. One of the major changes brought on by this was the shift from collecting data based on semantic violations in sentences to using the data provided by an experiment that observed semantic responses between different words that were displayed consecutively to subjects. It is unclear how much of a difference this change could have had on the overall project, and the use of different research groups' datasets introduced problems with learning how the dataset was organized and collected. Ideally, in the future, this research could be completed in its full scope using the sentence pairs attached in the appendices, then EEG data for these sentence sets could be compared to see if ongoing semantic violations in sentences lead to greater and more separable N400 and P600 response in subjects. In this future experiment it may also be possible to remove the button pressing marker in order to remove a confounding variable and present more solid evidence that classification can be done based on semantic violations alone.

Additionally, another limitation of this research is the ability to compare and contrast methods with other similar research projects. Often these projects are published but do not contain sufficient details on how the analysis was conducted in order to allow for a direct comparison. Increased transparency of the methods used in BCI research would significantly improve the ability of different researchers to recreate and validate the experiments of others. Even with this research looking at CSP with Naïve Bayes classification, it is still unknown whether *Calvo et al.* used different methods for isolating features, a different algorithm and parameters for the Bayesian model training, or even how many features of the CSP matric were utilized in generating their datasets. Access to this information would significantly improve the direct comparisons and allow for more concrete answers on which methods are actually viable for continued research.

Finally, one last significant limitation of this research is the sample size. Across 18 subjects 150 word pairs are available for each subject, which only allows for a very minimal amount of training for an algorithm. Having access to much larger training sets will allow for much better models and possibly greater accuracy. It will also be more akin to a real-world situation where a BCI device is being trained continuously by the user instead of being trained on a small training set then tested.

7. Conclusion

This study has helped broaden the understanding of Common Spatial Patterns in ERP-based error detection and has shown that this model may have some promise for specific users of BCI technology. Furthermore, by analysis of specific semantic related ERP components, the study shows that CSP used in conjunction with classifier models such as LDA, QDA, NB, or KNN can provide classification results greater than random chance for data sets with inseparable means. This assertion challenges previous understandings in BCI research that state CSP is not useful when dealing with ERP-based data. Additional studies will help to solidify the degree to which CSP can be used and implemented in BCI devices, as well as identifying and understanding indicators that suggest which subjects will benefit the most from CSP-based ERP classification devices.

8. Future Work

The most important future work to determine the usefulness of CSP-based ERP detection algorithms will be repeated experiments that indicate the full applicability of CSP in these scenarios. It will be important to include full sentence trials, as well as testing of other types of classifier algorithms such as Decision Trees and Support Vector Machines. In a future experiment, a classification by committee algorithm that allows each classifier to vote may help to further improve the performance by combining many of the best-case scenarios. Additionally, it will be equally important that these studies carefully outline and explain their methods and procedures so that these tests can be repeated and verified with new data, or existing data can be reanalyzed using new features as our understanding of neurophysiology continues to grow. Additional studies can also examine other uses of CSP outside of semantic classification, as it may be a relatively useful tool in any situation where data means between two classes are the same. Finally, if CSP is established as a useful technology for ERP-based detection, the next big step will be translating this technology to real-world situations outside of a lab. This means designing robust EEG equipment and rejection criteria that can handle a subject interacting with the entire world, not just a screen for an experiment. These technologies must be able to function with a high degree of precision regardless of the circumstances for them to have an appreciable use for end-users.

List of References

[1] J. R. Wolpaw and E. W. Wolpaw, *Brain-Computer Interfaces: Principles and Practice*. 2012.

[2] W. Kester, “What the Nyquist Criterion Means to Your Sampled Data System Design,” , 2016.

[3] P. Ledwidge, “The Impact of Sports-Related Concussions on the Language System: A Case for Event-Related Brain Potentials,” *Ann. Behav. Neurosci.*, vol. 1, no. 1, pp. 36–46, 2018.

[4] A. M. Jette, C. M. Spicer, and J. L. Flaubert, *Eliminating or Reducing the Effects of Impairments*. 2017.

[5] T. Zeyl, E. Yin, M. Keightley, and T. Chau, “Adding real-time Bayesian ranks to error-related potential scores improves error detection and auto-correction in a P300 speller,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 1, pp. 46–56, 2016.

[6] E. Bullmore and O. Sporns, “Complex brain networks: Graph theoretical analysis of structural and functional systems,” *Nat. Rev. Neurosci.*, vol. 10, no. 3, pp. 186–198, 2009.

[7] A.R.V., “Histonomy of the Cerebral Cortex,” *Neurology*, vol. 11, no. 7. pp. 649–649, 1961.

[8] C. Van Petten and B. J. Luka, “Prediction during language comprehension: Benefits, costs, and ERP components,” *Int. J. Psychophysiol.*, vol. 83, no. 2, pp. 176–190, 2012.

[9] H. Brouwer and M. W. Crocker, “On the proper treatment of the N400 and P600 in language comprehension,” *Front. Psychol.*, vol. 8, no. AUG, pp. 1–5, 2017.

[10] B. Penolazzi, A. Angrilli, and R. Job, “Gamma EEG activity induced by semantic violation during sentence reading,” *Neurosci. Lett.*, vol. 465, no. 1, pp. 74–78, 2009.

[11] L. A. Hald, M. C. M. Bastiaansen, and P. Hagoort, “EEG theta and gamma responses to semantic violations in online sentence processing,” *Brain Lang.*, vol. 96, no. 1, pp. 90–105, 2006.

[12] H. Calvo, J. L. Paredes, and J. Figueroa-Nazuno, “Measuring concept semantic relatedness through common spatial pattern feature extraction on EEG signals,” *Cogn. Syst. Res.*, vol. 50, pp. 36–51, 2018.

[13] S. Mathôt, J. Melmi, L. Van Der Linden, and S. Van Der, “The Mind-Writing Pupil : A Human-Computer Interface Based on Decoding of Covert Attention through Pupillometry,” pp. 1–15, 2016.

[14] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, “Brain-Computer Interfaces in Medicine,” *JMCP*, vol. 87, no. 3, pp. 268–279, 2012.

[15] H. Higashi and T. Tanaka, “Common spatio-time-frequency patterns for motor imagery-based brain machine interfaces,” *Comput. Intell. Neurosci.*, vol. 2013, 2013.

[16] Brumberg JS; Nieto-Castanon A; Kennedy PR; Guenther FH., "Brain Computer Interfaces for Speech Communication," vol. 52, no. 4, pp. 367–379, 2010.

[17] J B.F. van Erp, F. Lotte and M. Tangerman, "Brain-Computer Interfaces: Beyond Medical Applications," *IEEE Comput.*, vol. 45, no. 4, pp. 26–34, 2012.

[19] V. Peterson, C. Galván, H. Hernández, and R. Spies, "A feasibility study of a complete low-cost consumer-grade brain-computer interface system," *Heliyon*, vol. 6, no. February, p. e03425, 2020.

[20] V. Jurcak, D. Tsuzuki, and I. Dan, "10 / 20 , 10 / 10 , and 10 / 5 systems revisited : Their validity as relative head-surface-based positioning systems ☆," vol. 34, pp. 1600–1611, 2007.

[21] D. Silverman, "The Rationale and History of the 10-20 System of the International Federation," vol. 9238, pp. 16–22, 2015.

[22] P. E. Clayson, S. A. Baldwin, and M. J. Larson, "Methodological reporting behavior , sample sizes , and statistical power in studies of event - related potentials : Barriers to reproducibility and replicability," no. May, pp. 1–17, 2019.

[23] A. V. Oppenheim and R. W. Schafer, "Discrete Time Signal Processing 2nd Edition," *Book*. 1998.

[24] H. Ramoser, J. Müller-Gerking, and P. Gert, "Optimal Spatial Filtering of Single Trial EEG During Imagined Hand Movement," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 441–446, 2000.

[25] A. D. Patel, E. Gibson, J. Ratner, M. Besson, and P. J. Holcomb, "Processing syntactic relations in language and music: An event-related potential study," *J. Cogn. Neurosci.*, vol. 10, no. 6, pp. 717–733, 1998.

[26] Z. Seyednozadi, R. Pishghadam, and M. Pishghadam, "Functional Role of the N400 and the P600 in Language-related ERP Studies with Respect to Semantic Processing: An Overview," *Arch. Neuropsychiatry*, no. 5, pp. 1–4, 2021.

[27] Y. T. Chang, L. C. Ku, C. L. Wu, and H. C. Chen, "Event-related potential (ERP) evidence for the differential cognitive processing of semantic jokes and pun jokes," *J. Cogn. Psychol.*, vol. 31, no. 2, pp. 131–144, 2019.

[28] S. N. Emerson, C. M. Conway, and Ş. Özçalışkan, "Semantic P600—but not N400—effects index crosslinguistic variability in speakers' expectancies for expression of motion," *Neuropsychologia*, vol. 149, no. September, 2020.

[29] N. Calma-Roddin and J. E. Drury, "Music, Language, and The N400: ERP Interference Patterns Across Cognitive Domains," *Sci. Rep.*, vol. 10, no. 1, pp. 1–14, 2020.

[30] C. Dudschig, I. G. Mackenzie, C. Maienborn, B. Kaup, and H. Leuthold, "Negation and the N400: investigating temporal aspects of negation integration using semantic and world-knowledge violations," *Lang. Cogn. Neurosci.*, vol. 34, no. 3, pp. 309–319, 2019.

[31] A. Schacht, W. Sommer, O. Shmuilovich, P. C. Martíenz, and M. Martín-Loeches,

"Differential task effects on N400 and P600 elicited by semantic and syntactic violations," *PLoS One*, vol. 9, no. 3, pp. 1–7, 2014.

[32] M. Balconi and U. Pozzoli, "N400 and P600 or the Role of the ERP Correlates in Sentence Comprehension: Some Applications to the Italian Language," *J. Gen. Psychol.*, vol. 131, no. 3, pp. 268–303, 2004.

[33] M. Kutas and K. D. Federmeier, "Thirty years and counting: Finding meaning in the N400 component of the event-related brain potential (ERP)," *Annu. Rev. Psychol.*, vol. 62, pp. 621–647, 2011.

[34] A. S. Mehravari, D. Tanner, E. K. Wampler, G. D. Valentine, and L. Osterhout, "Effects of grammaticality and morphological complexity on the P600 event-related potential component," *PLoS One*, vol. 10, no. 10, pp. 1–16, 2015.

[35] A. M. Collins and M. R. Quillian, "Retrieval Time from Semantic Memory 1," *J. Verbal Learning Verbal Behav.*, vol. 247, no. 1969, pp. 240–247, 1982.

[36] A. K. Engel and W. Singer, "Temporal binding and the neural correlates of sensory awareness," *Trends Cogn. Sci.*, vol. 5, no. 1, pp. 16–25, 2001.

[37] J. J. Foster, D. W. Sutterer, J. T. Serences, E. K. Vogel, and E. Awh, "Alpha-Band Oscillations Enable Spatially and Temporally Resolved Tracking of Covert Spatial Attention," *Psychol. Sci.*, vol. 28, no. 7, pp. 929–941, 2017.

[38] G. Buzsáki, "Theta rhythm of navigation: Link between path integration and landmark navigation, episodic and semantic memory," *Hippocampus*, vol. 15, no. 7, pp. 827–840, 2005.

[39] M. Congedo, L. Korczowski, A. Delorme, and F. Lopes da silva, "Spatio-temporal common pattern: A companion method for ERP analysis in the time domain," *J. Neurosci. Methods*, vol. 267, pp. 74–88, 2016.

[40] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," *Clin. Neurophysiol.*, vol. 110, no. 5, pp. 787–798, 1999.

[41] E. R. Schotter, R. Tran, and K. Rayner, "Don't believe what you read (Only Once): Comprehension is supported by regressions during reading," *Psychol. Sci.*, vol. 25, no. 6, pp. 1218–1226, 2014.

[42] S. Benedetto, A. Carbone, M. Pedrotti, K. Le Fevre, L. A. Y. Bey, and T. Baccino, "Rapid serial visual presentation in reading: The case of Spritz," *Comput. Human Behav.*, vol. 45, pp. 352–358, 2015.

[43] "Augmentative and Alternative Communication." *American Speech-Language-Hearing Association*, American Speech-Language-Hearing Association, www.asha.org/practice-portal/professional-issues/augmentative-and-alternative-communication/.

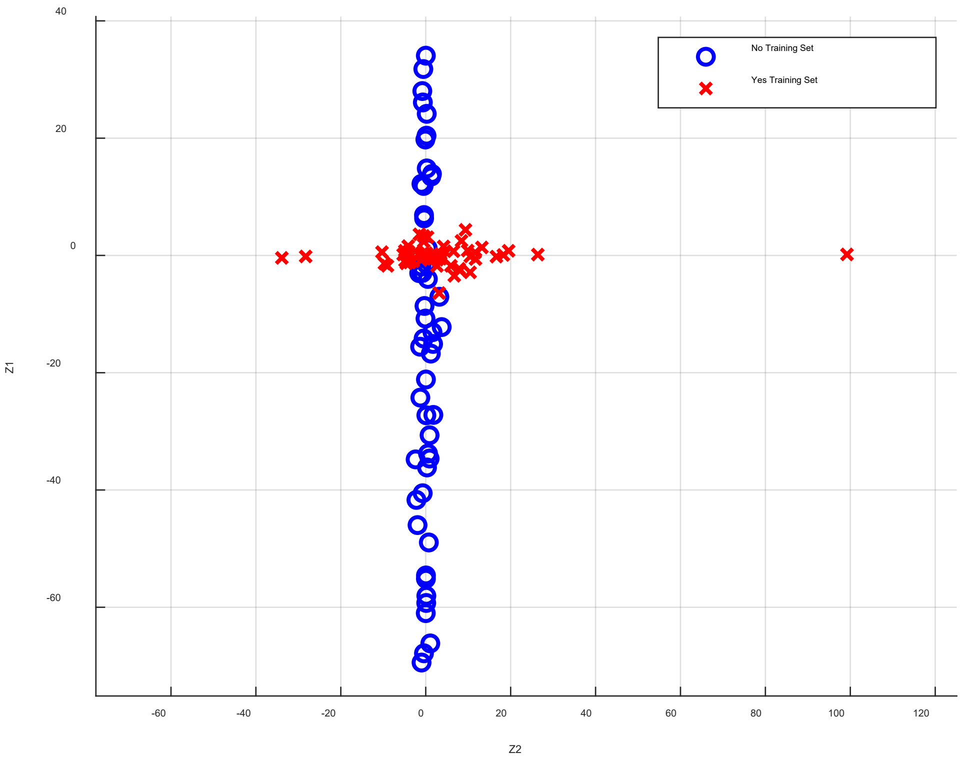
[44] “Augmentative and Alternative Communication (AAC).” *American Speech-Language-Hearing Association*, American Speech-Language-Hearing Association, www.asha.org/njc/aac/.

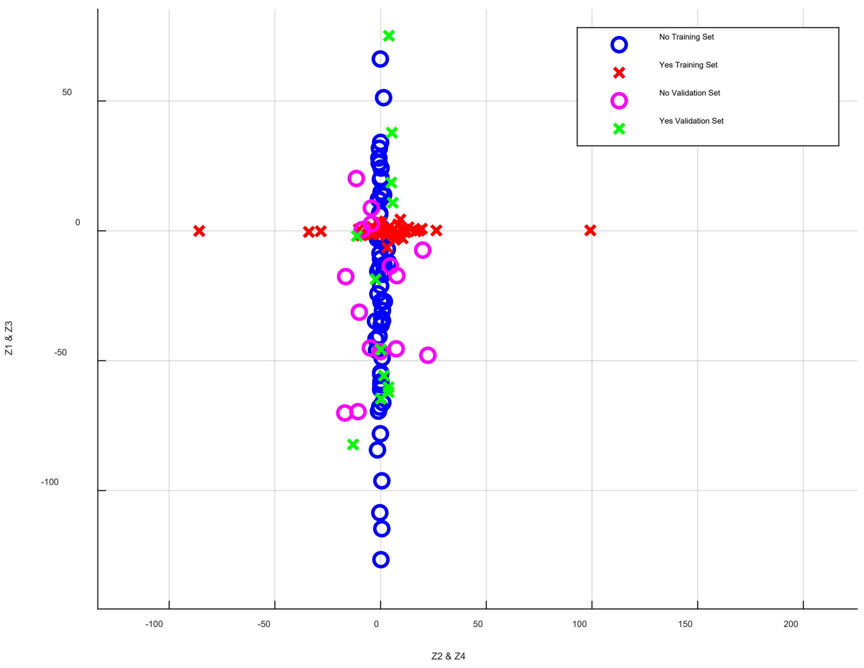
[45] “Convolutional Neural Networks for Visual Recognition.” *CS231n Convolutional Neural Networks for Visual Recognition*, Stanford, cs231n.github.io/classification/.

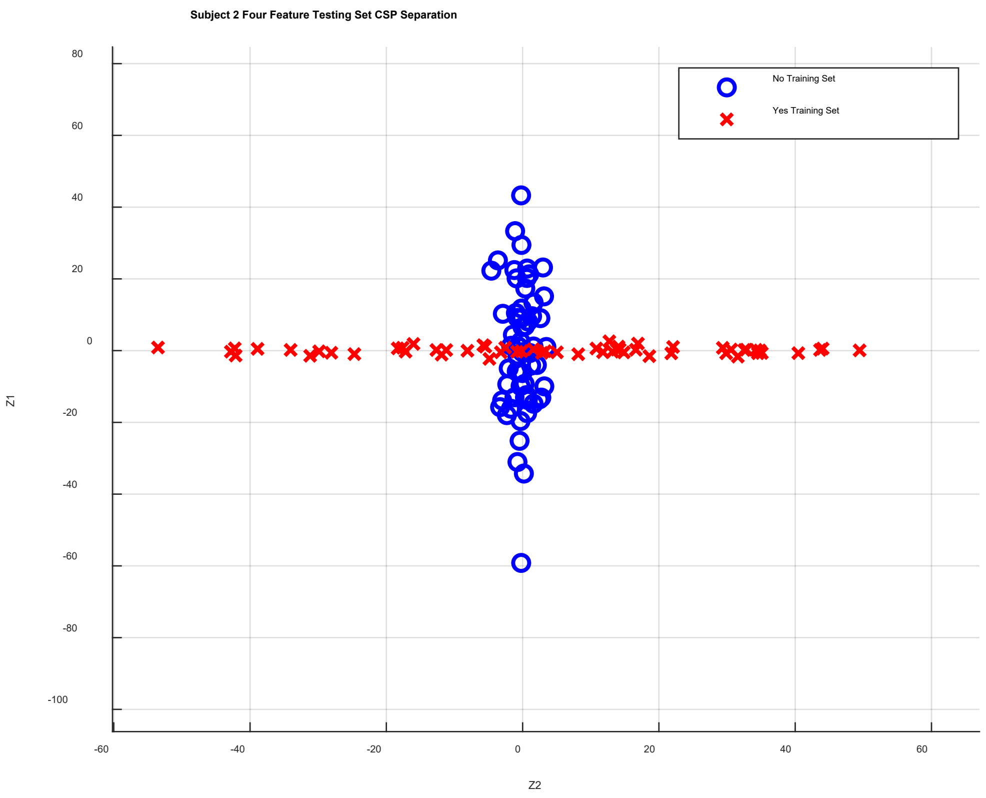
[46] “1.2. Linear and Quadratic Discriminant Analysis.” *Scikit*, scikit-learn.org/stable/modules/lda_qda.html.

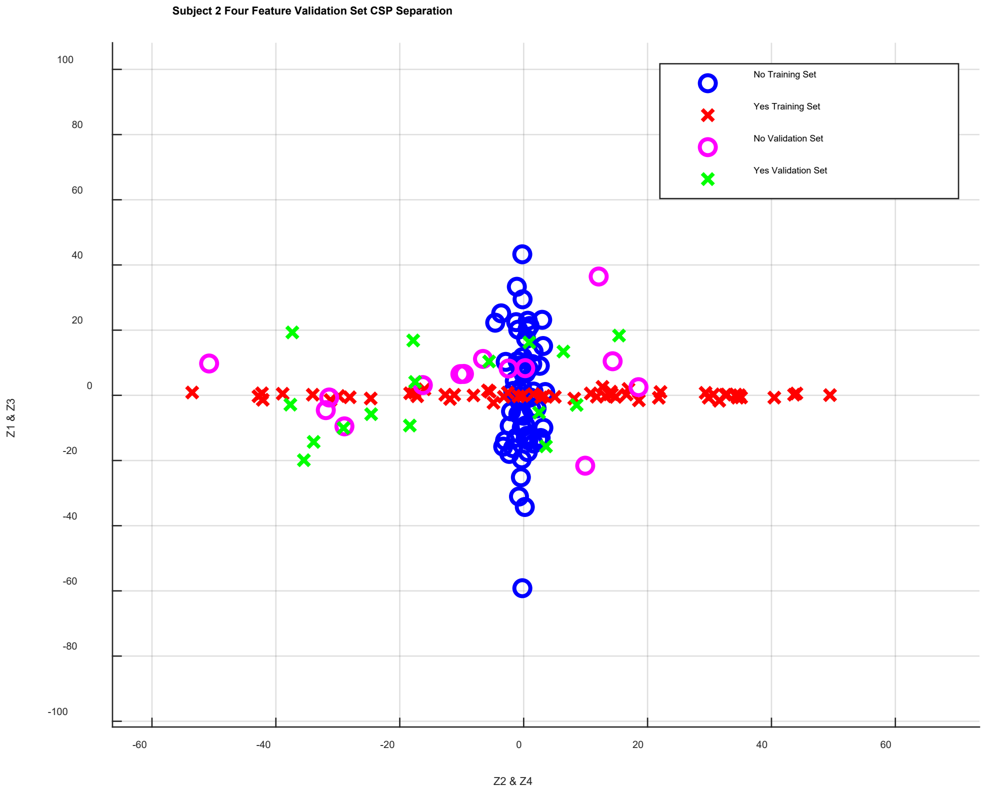
Appendices

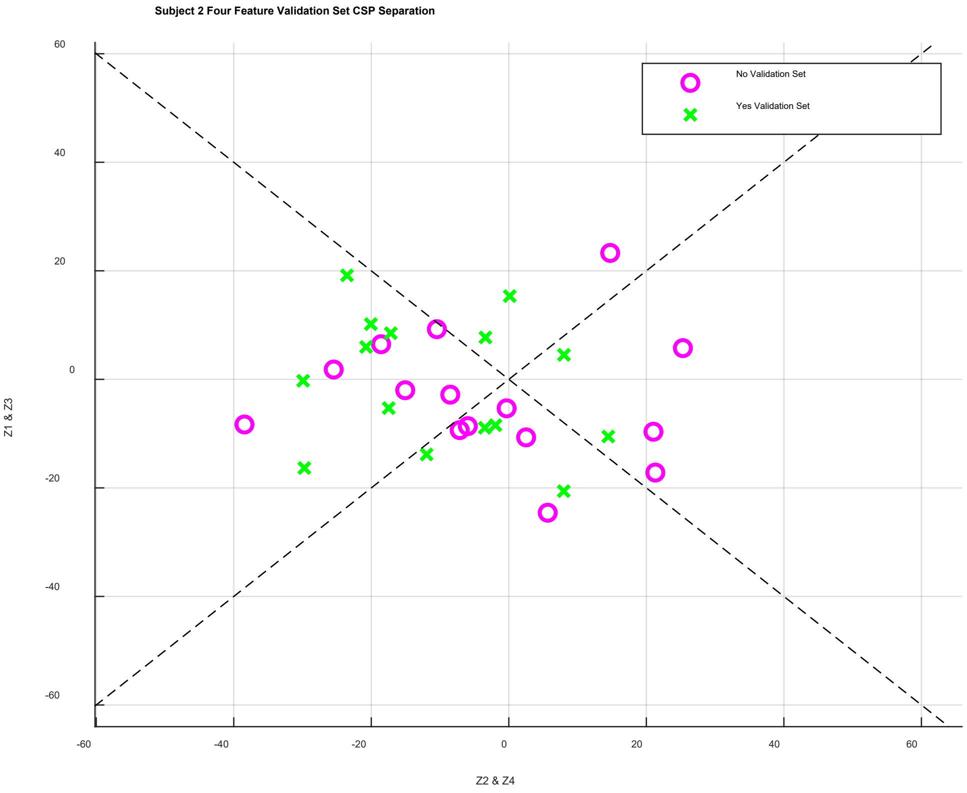
Appendix A

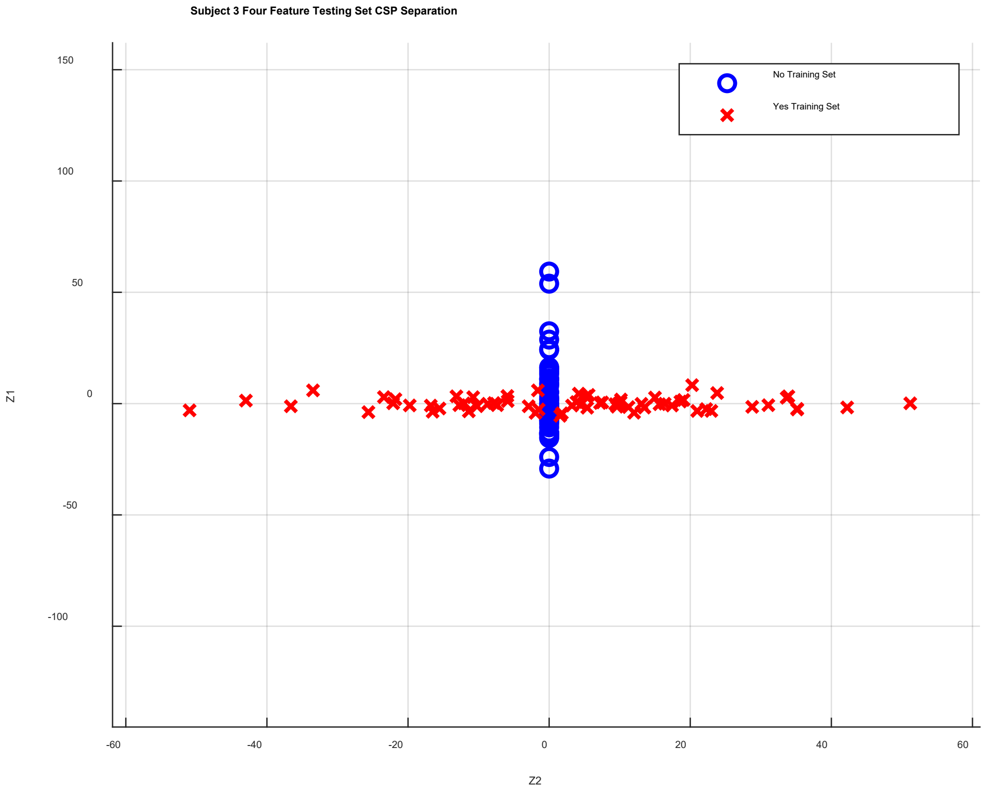

Subject 1 Four Feature Testing Set CSP Separation


Subject One Four Feature Validation Set CSP Separation

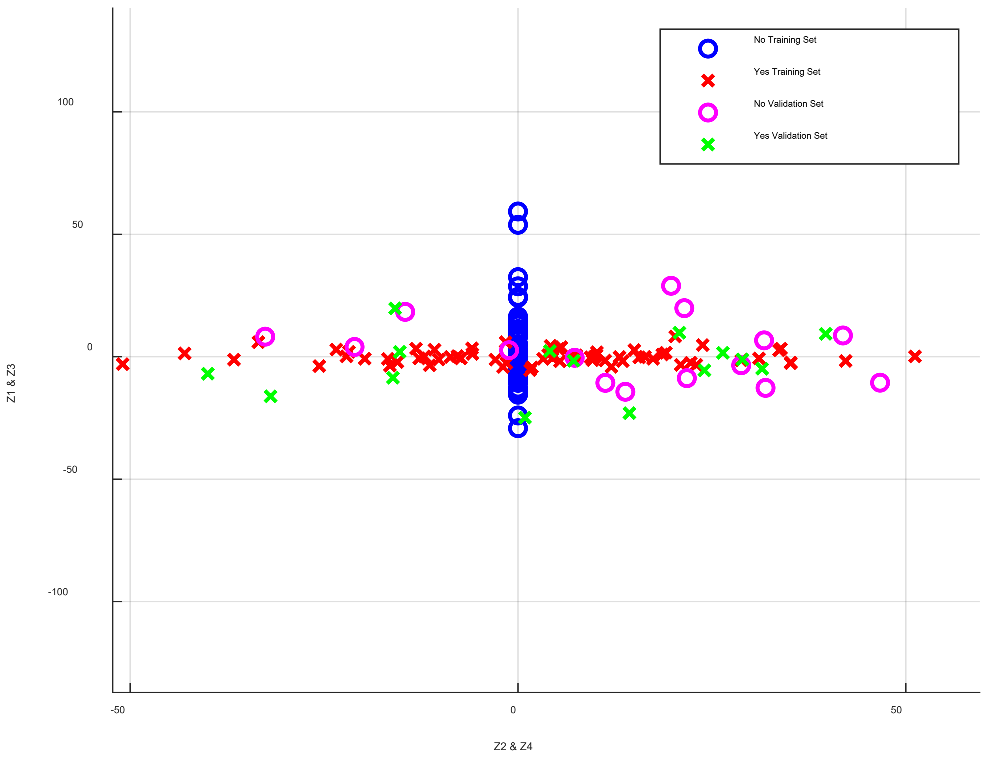

Subject 2 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
80
60
40
20
0
-20
-40
-60
-80
-100
Z1
-60
-40
-20
0
20
40
60
Z2


Subject 2 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
100
80
60
40
20
0
-20
-40
-60
-80
-100
Z1 & Z3
-60
-40
-20
0
20
40
60
Z2 & Z4

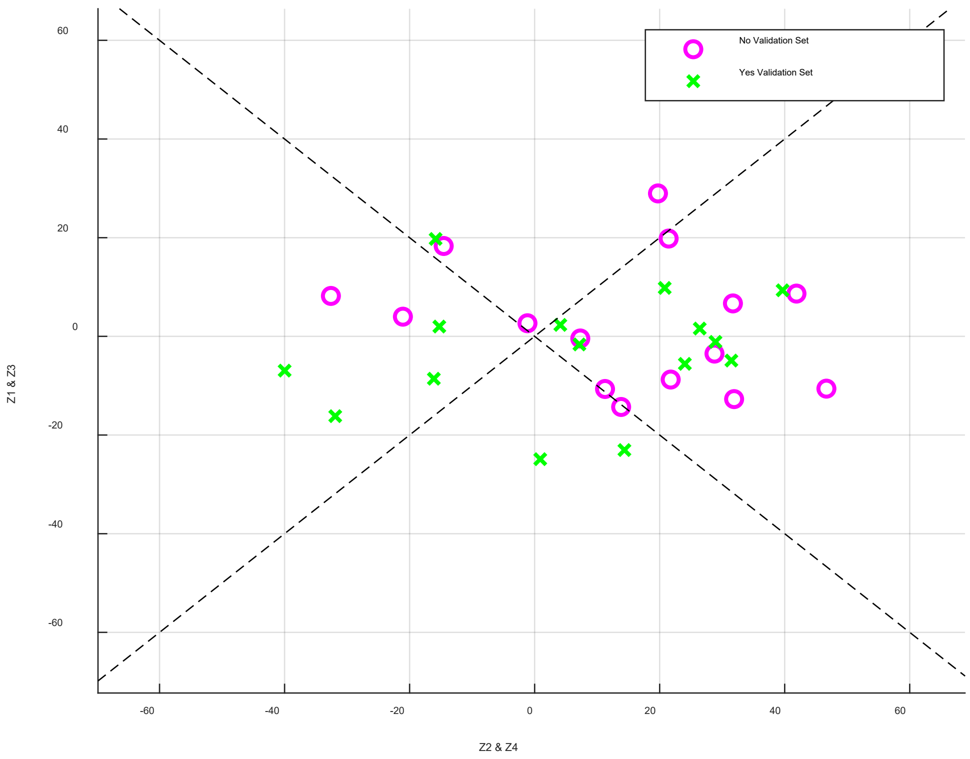

Subject 2 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
60
40
20
0
-20
-40
-60
Z1 & Z3
-60
-40
-20
0
20
40
60
Z2 & Z4

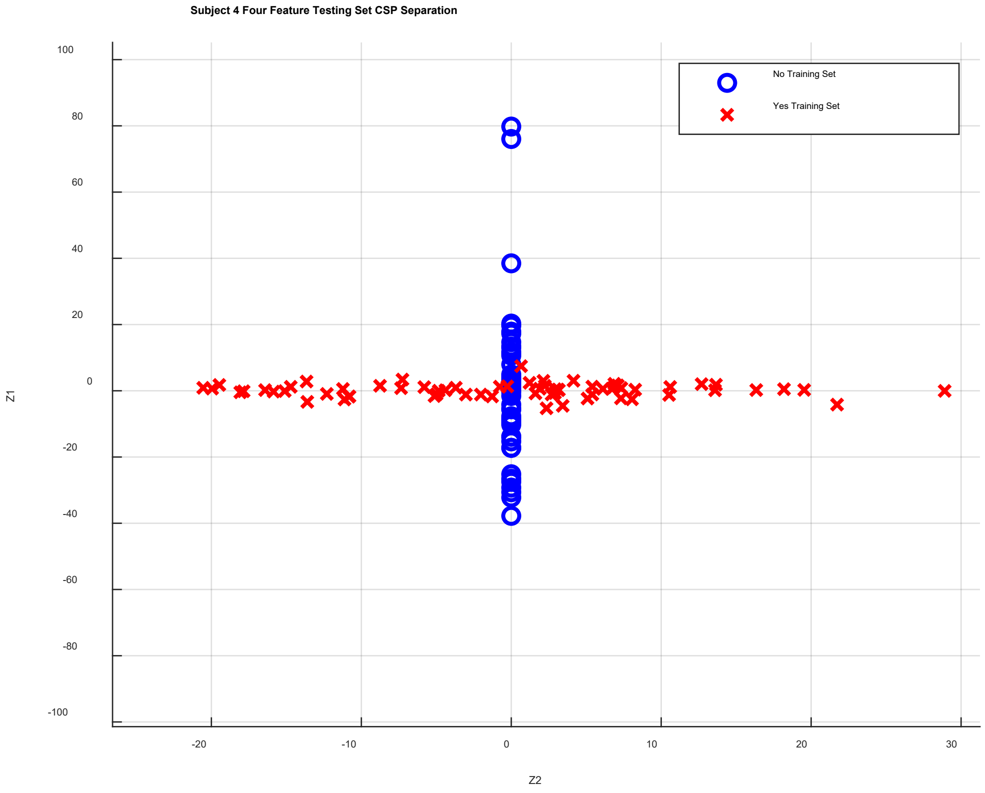

Subject 3 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
150
100
50
0
-50
-100
Z1
-60
-40
-20
0
20
40
60
Z2

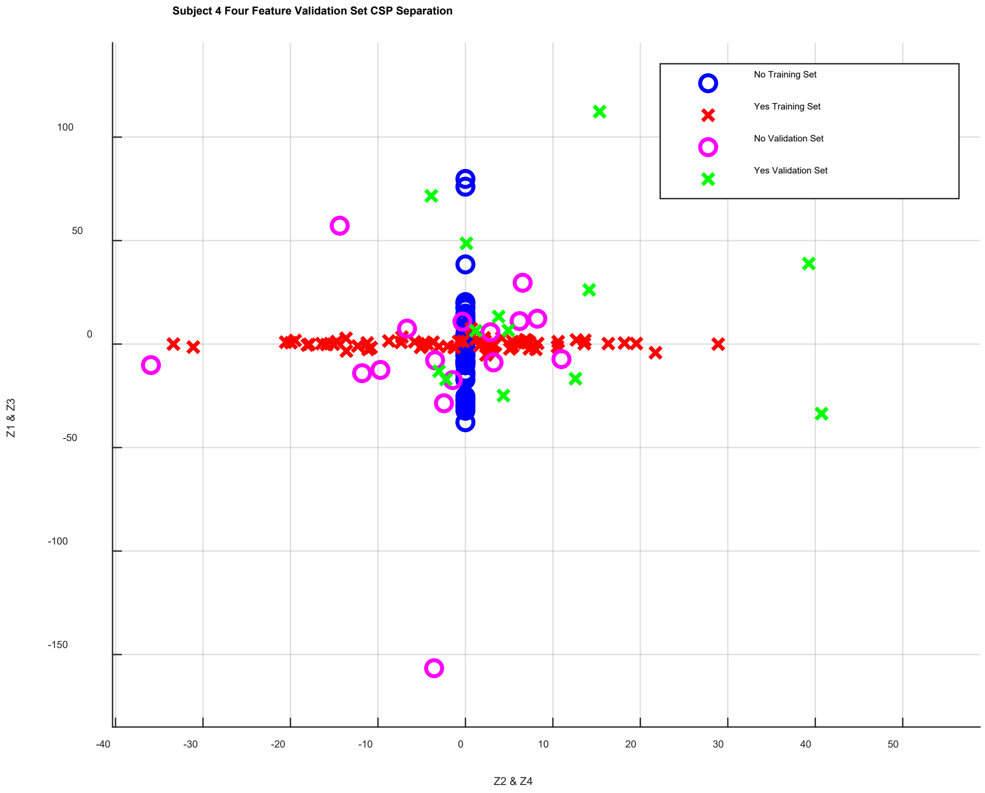
Subject 3 Four Feature Validation Set CSP Separation

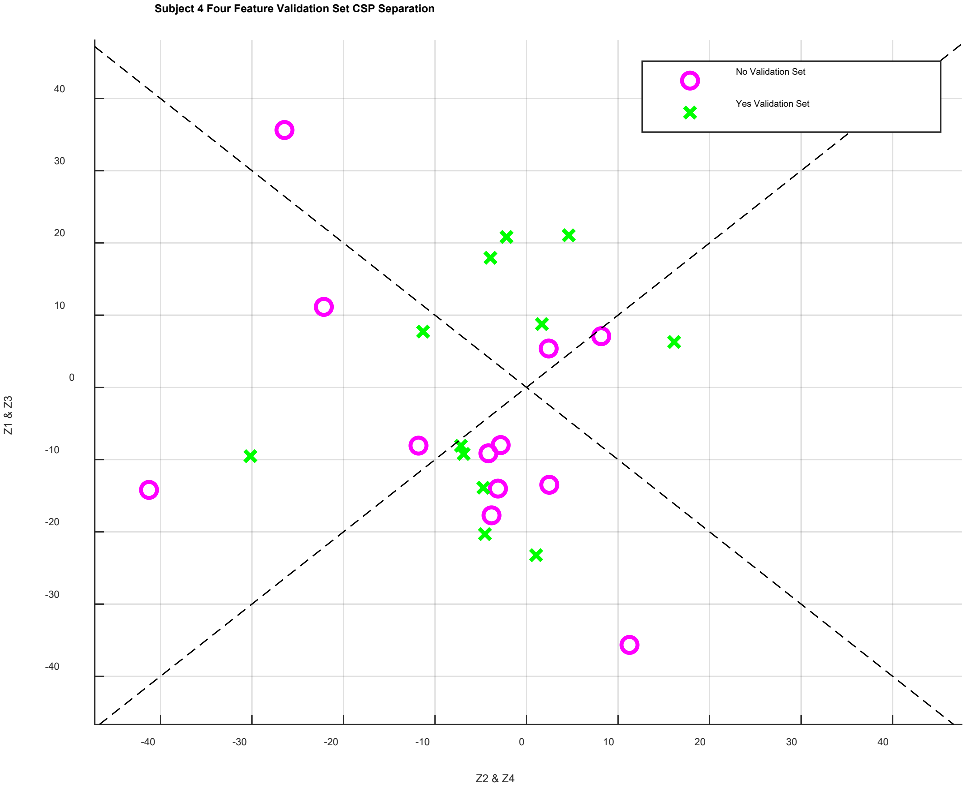


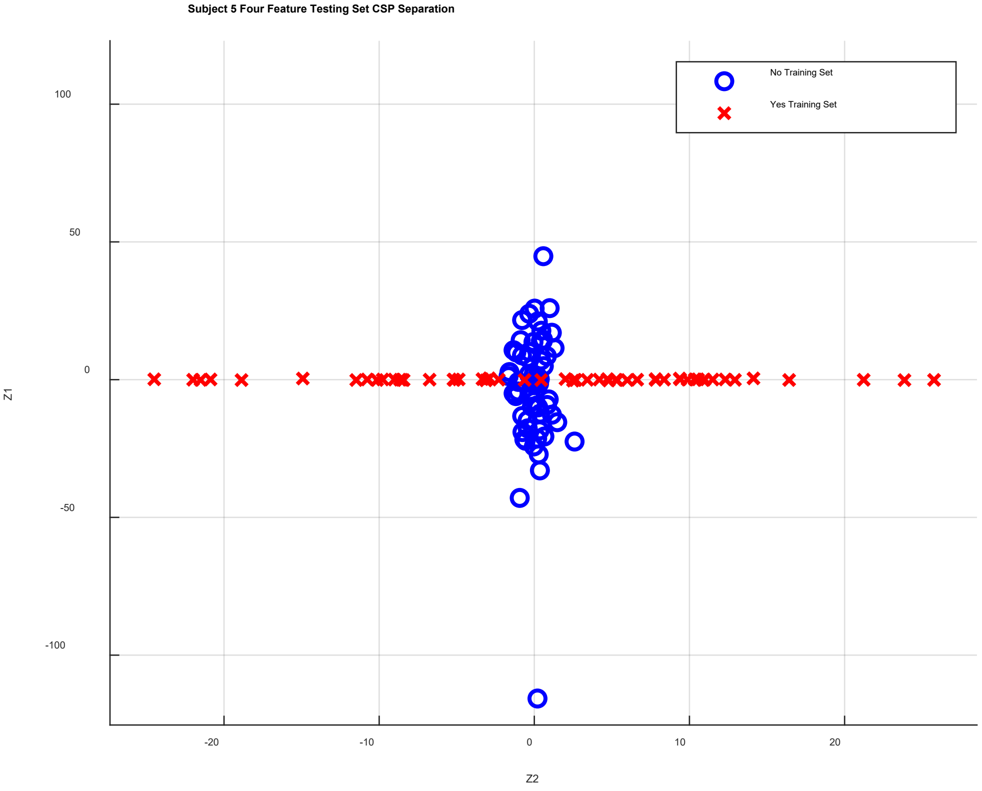
Subject 3 Four Feature Validation Set CSP Separation

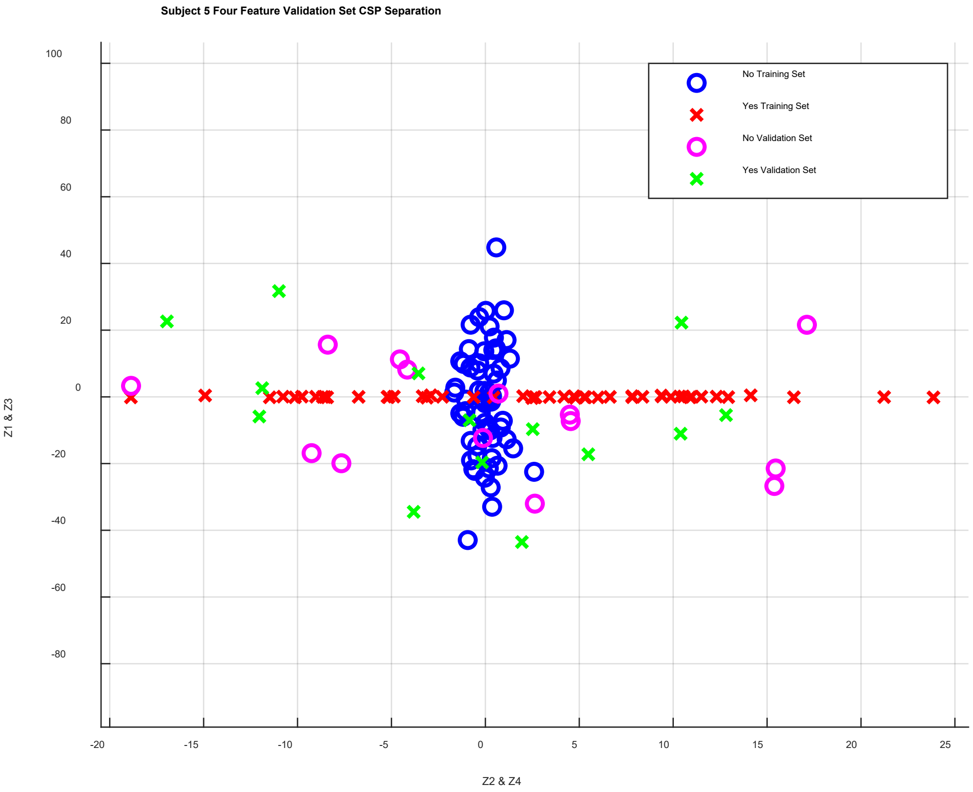


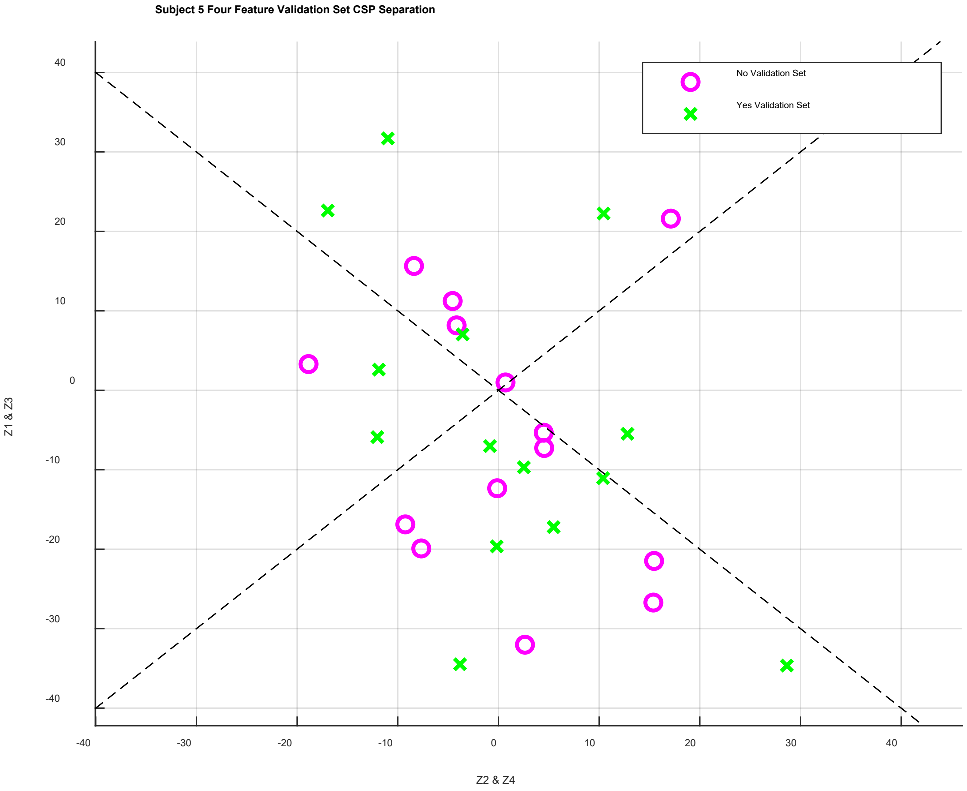

Subject 4 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
100
80
60
40
20
0
-20
-40
-60
-80
-100
Z1
-20
-10
0
10
20
30
Z2

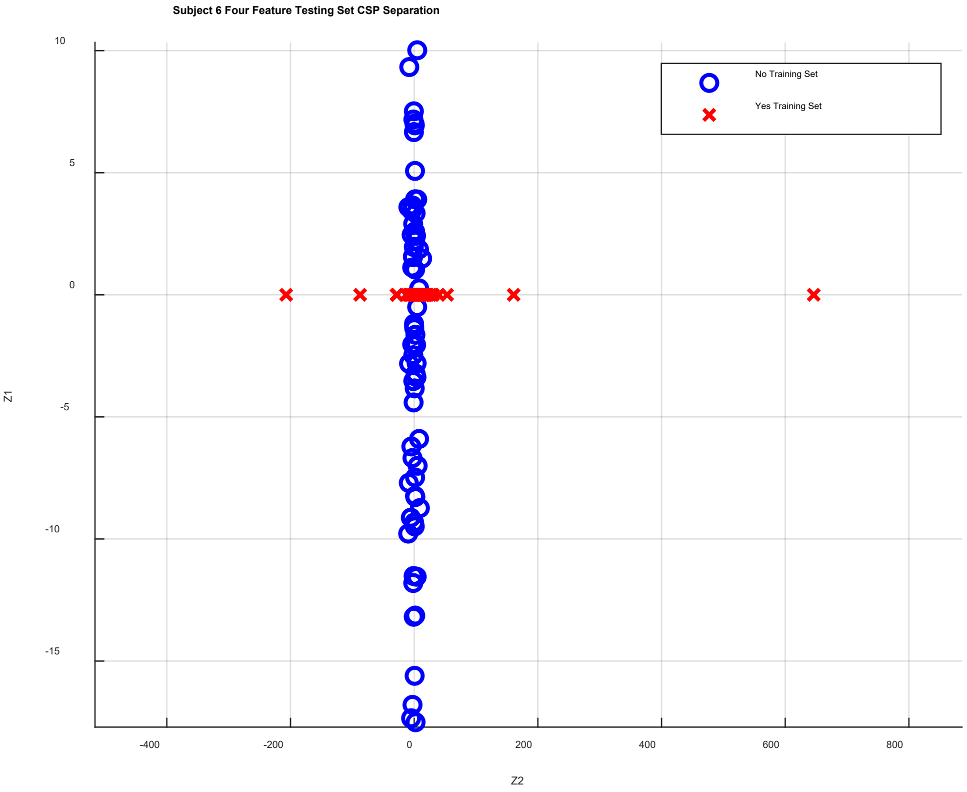

Subject 4 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
100
50
0
-50
-100
-150
Z1 & Z3
-40
-30
-20
-10
0
10
20
30
40
50
Z2 & Z4

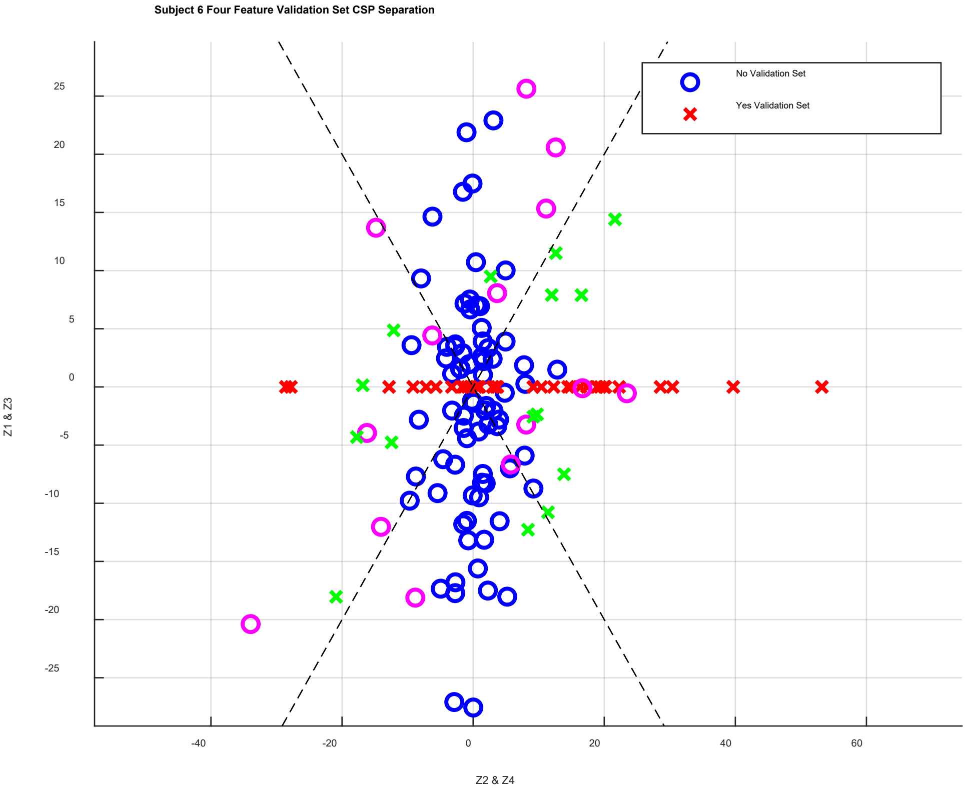

Subject 4 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
40
30
20
10
0
-10
-20
-30
-40
Z1 & Z3
-40
-30
-20
-10
0
10
20
30
40
Z2 & Z4

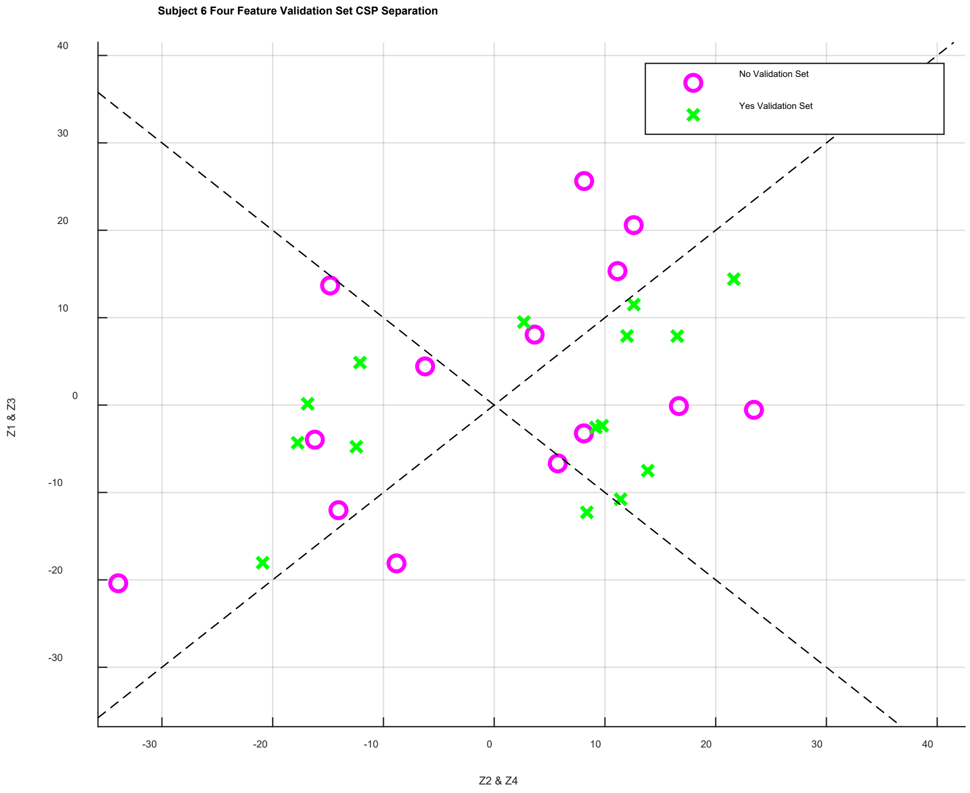

Subject 5 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
100
50
0
-50
-100
Z1
-20
-10
0
10
20
Z2

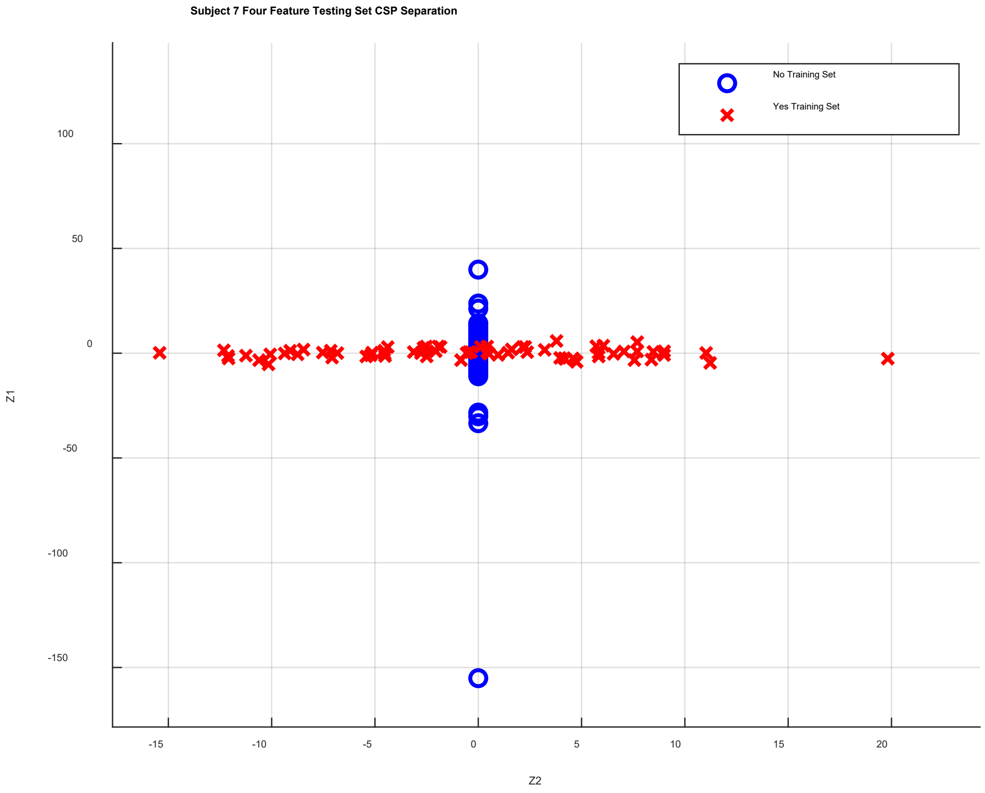

Subject 5 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
100
80
60
40
20
0
-20
-40
-60
-80
Z1 & Z3
-20
-15
-10
-5
0
5
10
15
20
25
Z2 & Z4


Subject 5 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
40
30
20
10
0
-10
-20
-30
-40
Z1 & Z3
-40
-30
-20
-10
0
10
20
30
40
Z2 & Z4

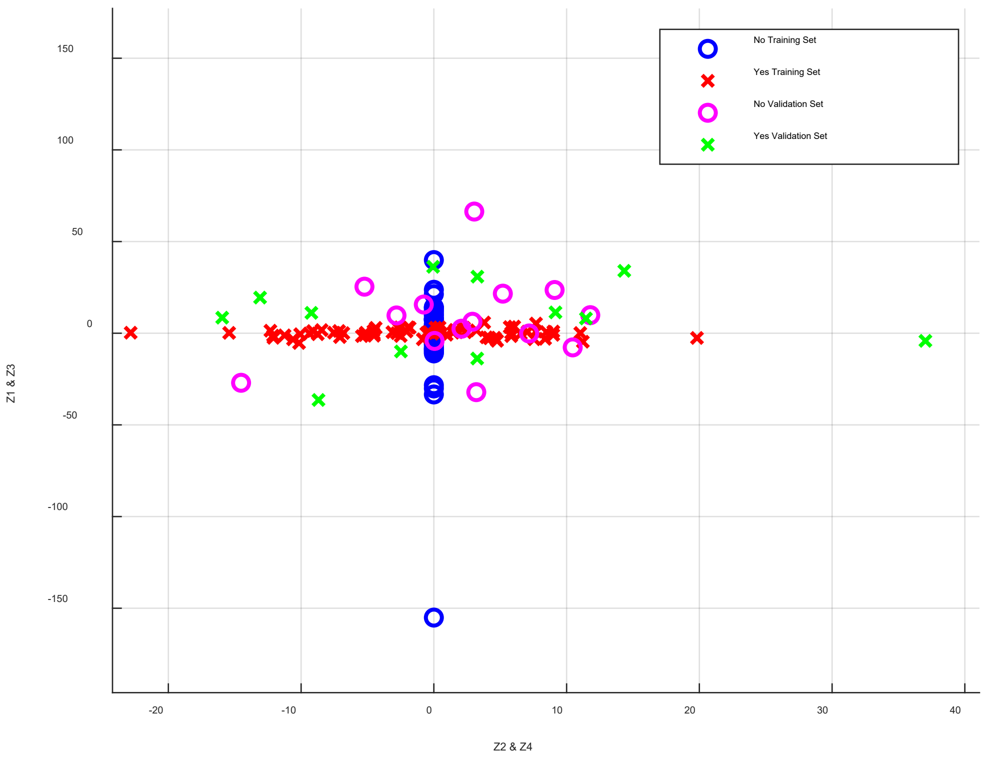

Subject 6 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
10
5
0
-5
-10
-15
Z1
-400
-200
0
200
400
600
800
Z2


Subject 6 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
25
20
15
10
5
0
-5
-10
-15
-20
-25
Z1 & Z3
-40
-20
0
20
40
60
Z2 & Z4

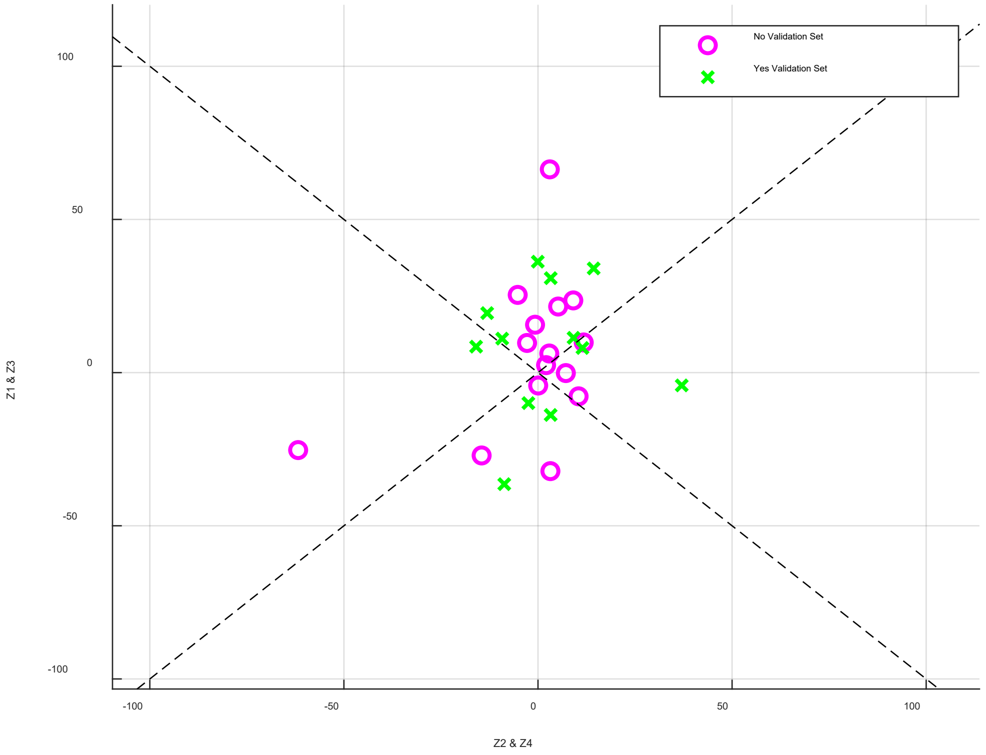

Subject 6 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
40
30
20
10
0
-10
-20
-30
Z1 & Z3
-30
-20
-10
0
10
20
30
40
Z2 & Z4

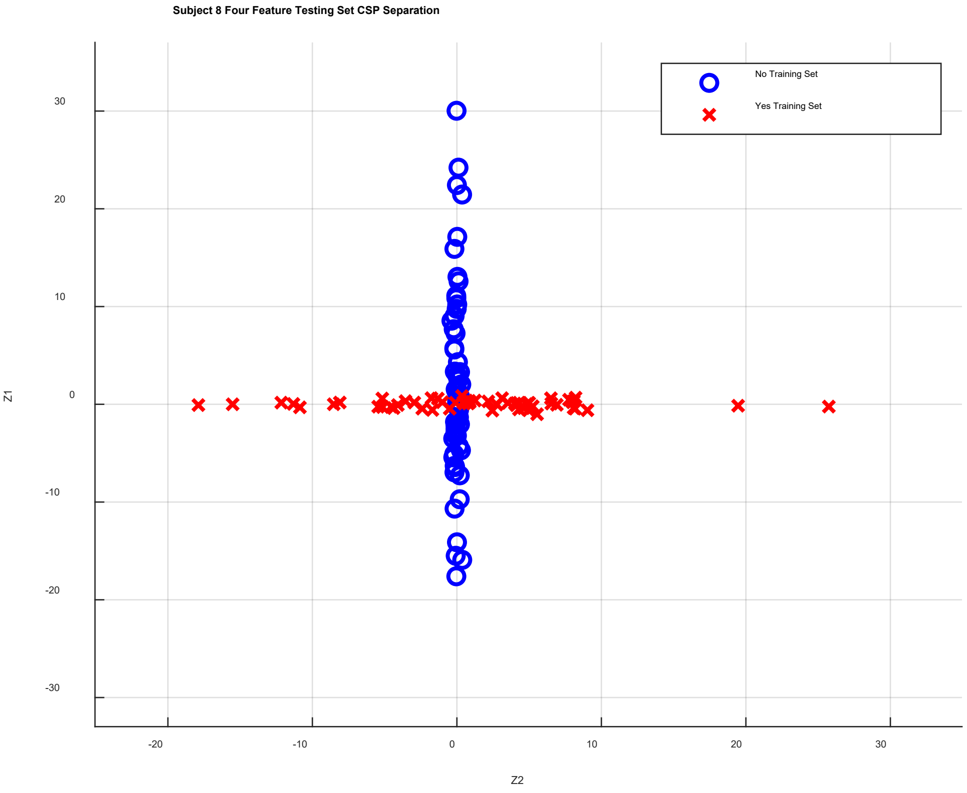

Subject 7 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
100
50
0
-50
-100
-150
Z1
-15
-10
-5
0
5
10
15
20
Z2

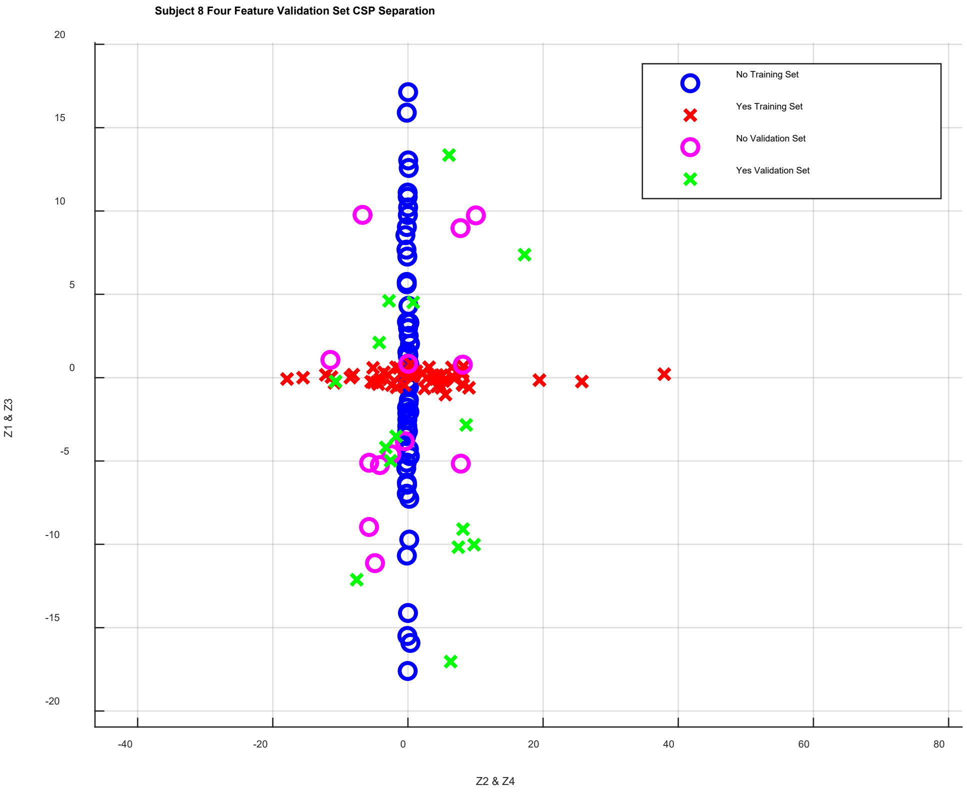
Subject 7 Four Feature Validation Set CSP Separation

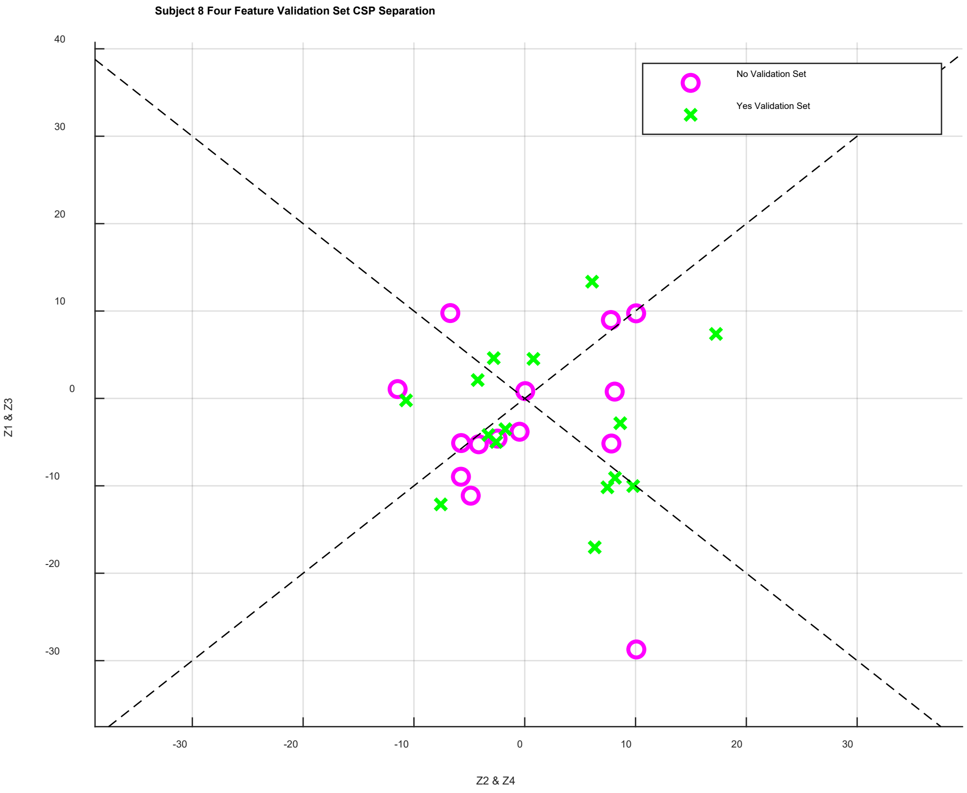


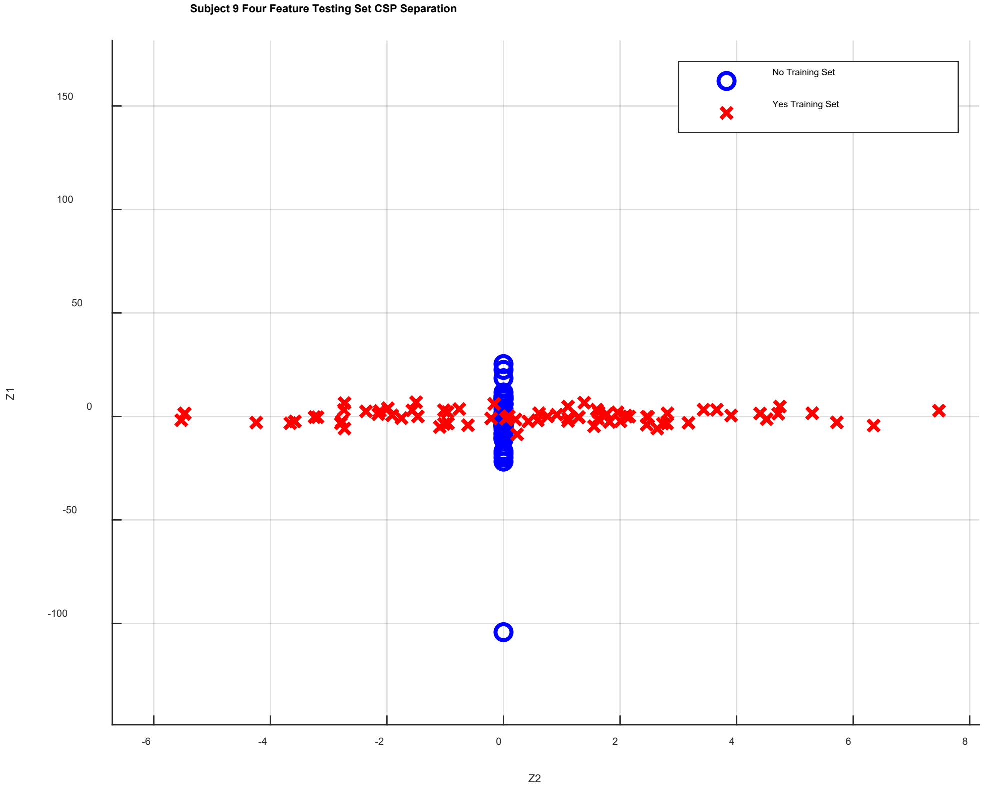
Subject 7 Four Feature Validation Set CSP Separation

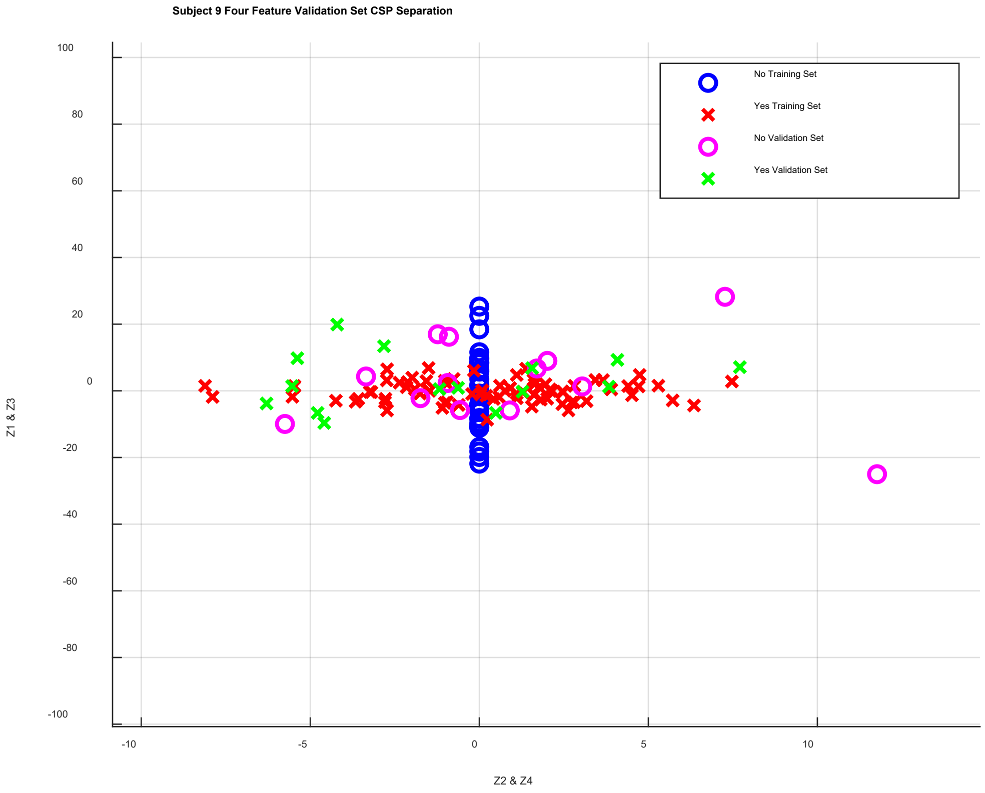


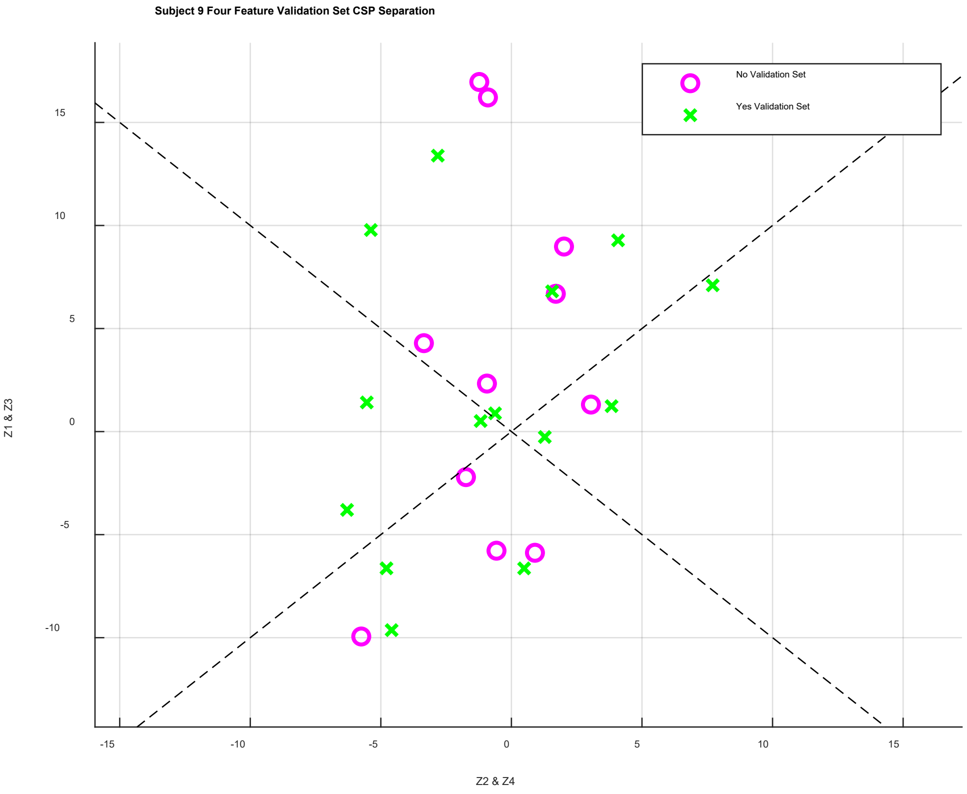

Subject 8 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
30
20
10
0
-10
-20
-30
Z1
-20
-10
0
10
20
30
Z2

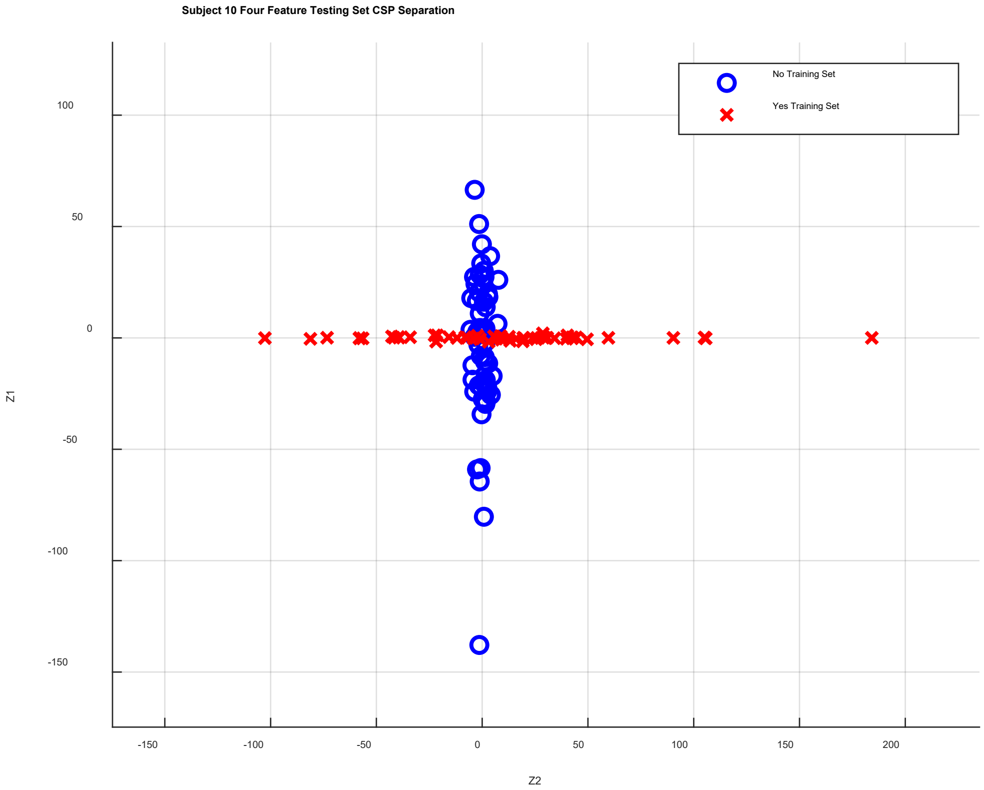

Subject 8 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
20
15
10
5
0
-5
-10
-15
-20
Z1 & Z3
-40
-20
0
20
40
60
80
Z2 & Z4


Subject 8 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
40
30
20
10
0
-10
-20
-30
Z1 & Z3
-30
-20
-10
0
10
20
30
Z2 & Z4

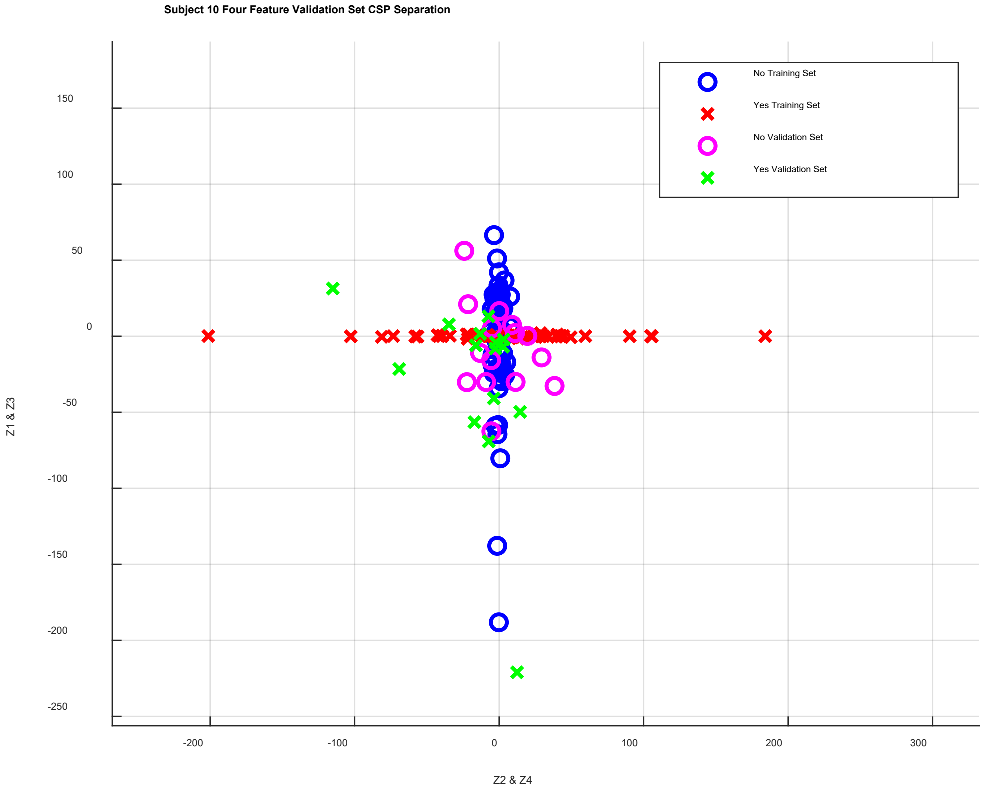

Subject 9 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
150
100
50
0
-50
-100
Z1
-6
-4
-2
0
2
4
6
8
Z2


Subject 9 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
100
80
60
40
20
0
-20
-40
-60
-80
-100
-10
-5
0
5
10
Z1 & Z3
Z2 & Z4

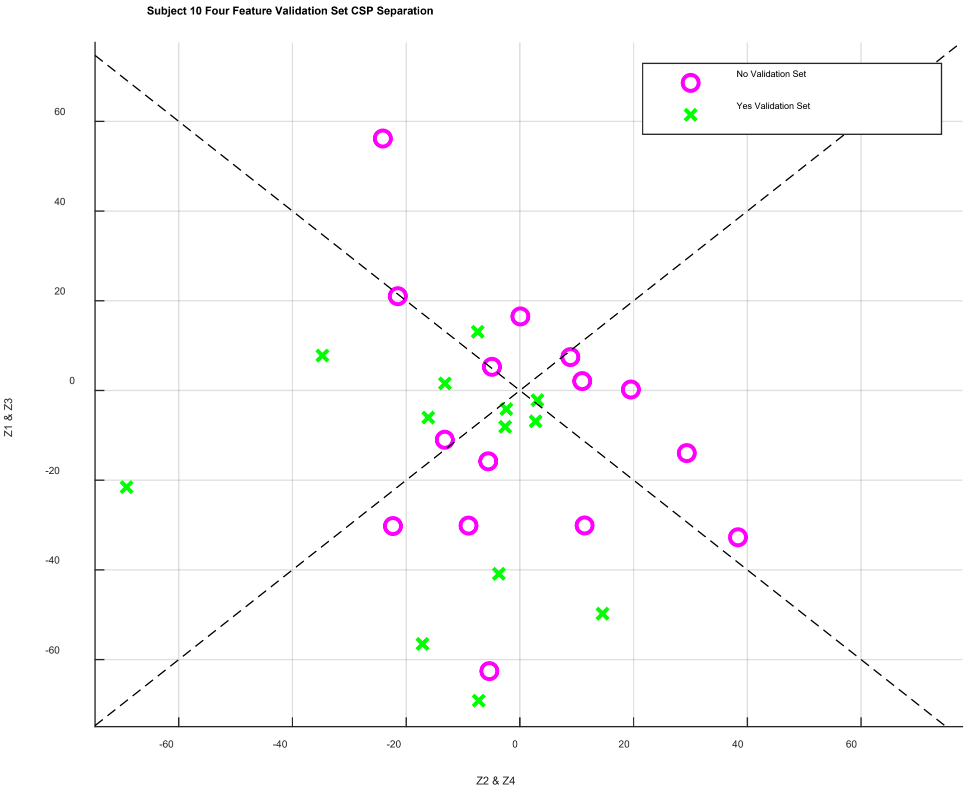

Subject 9 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
15
10
5
0
-5
-10
-15
-10
-5
0
5
10
15
Z1 & Z3
Z2 & Z4

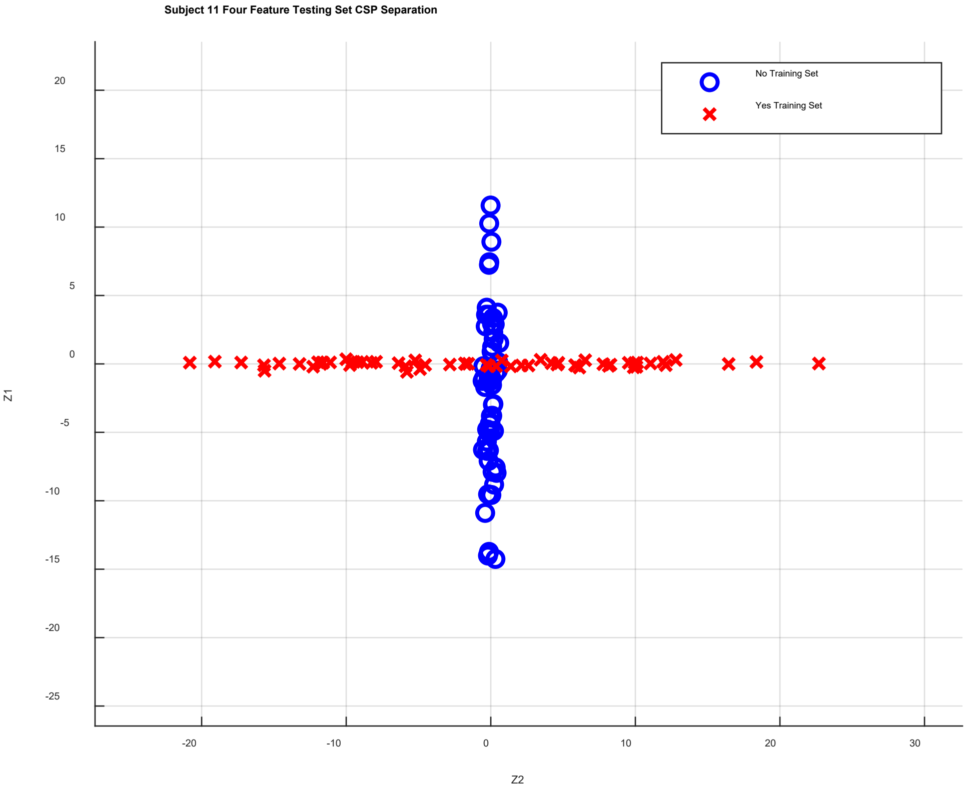


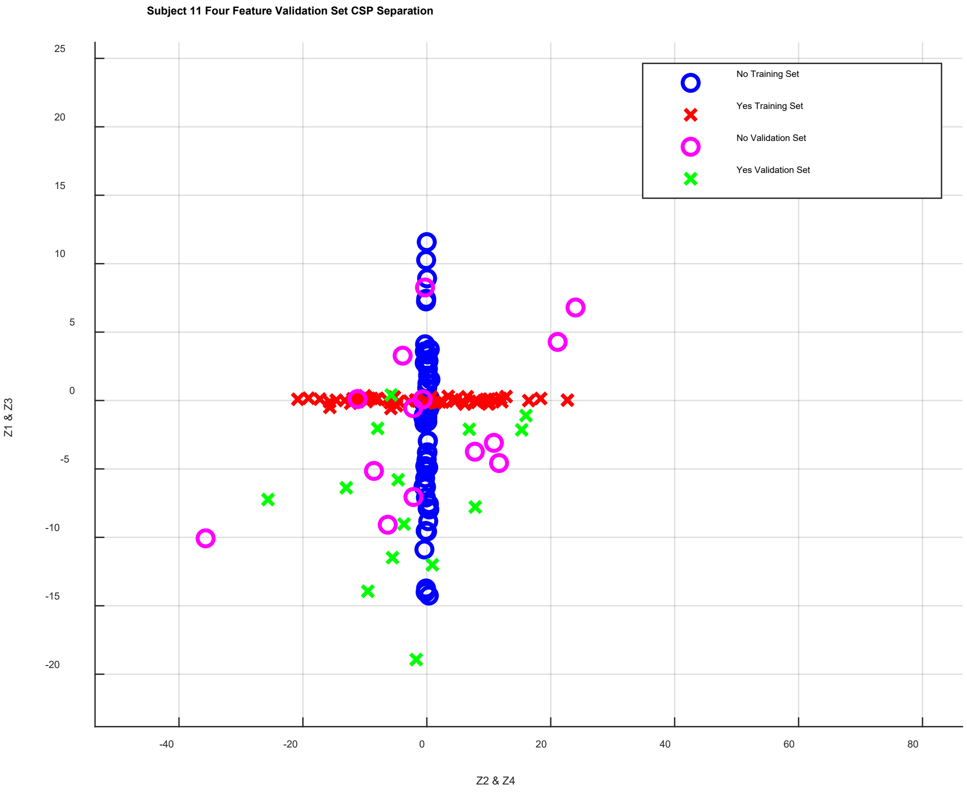
Subject 10 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
100
50
0
-50
-100
-150
Z1
-150
-100
-50
0
50
100
150
200
Z2

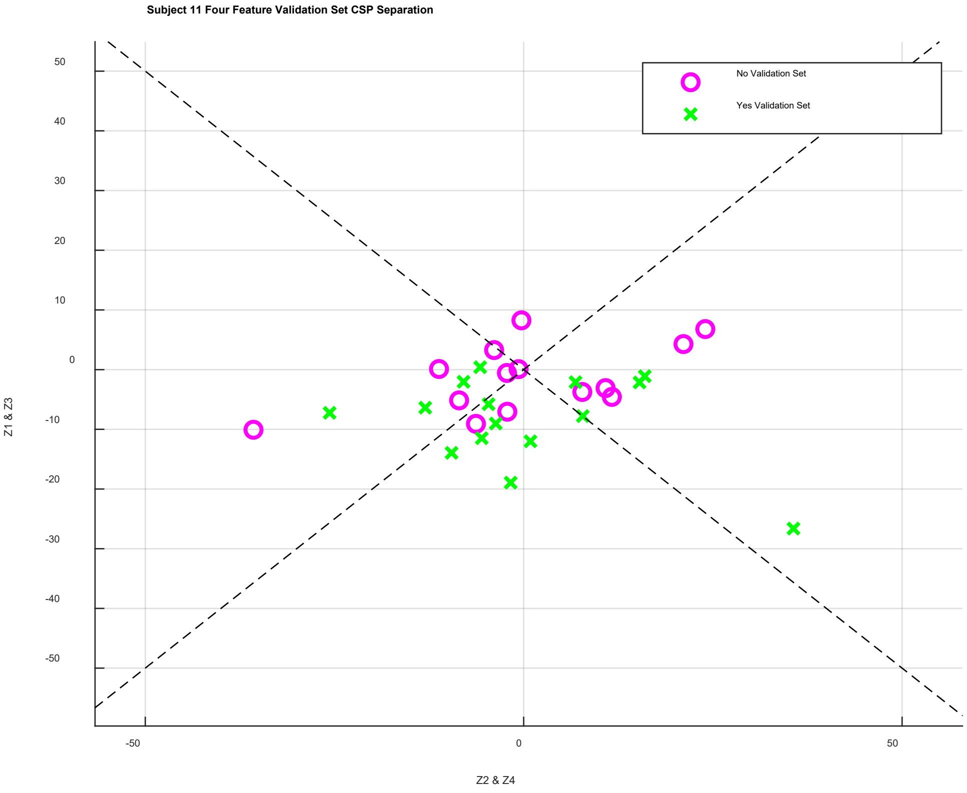


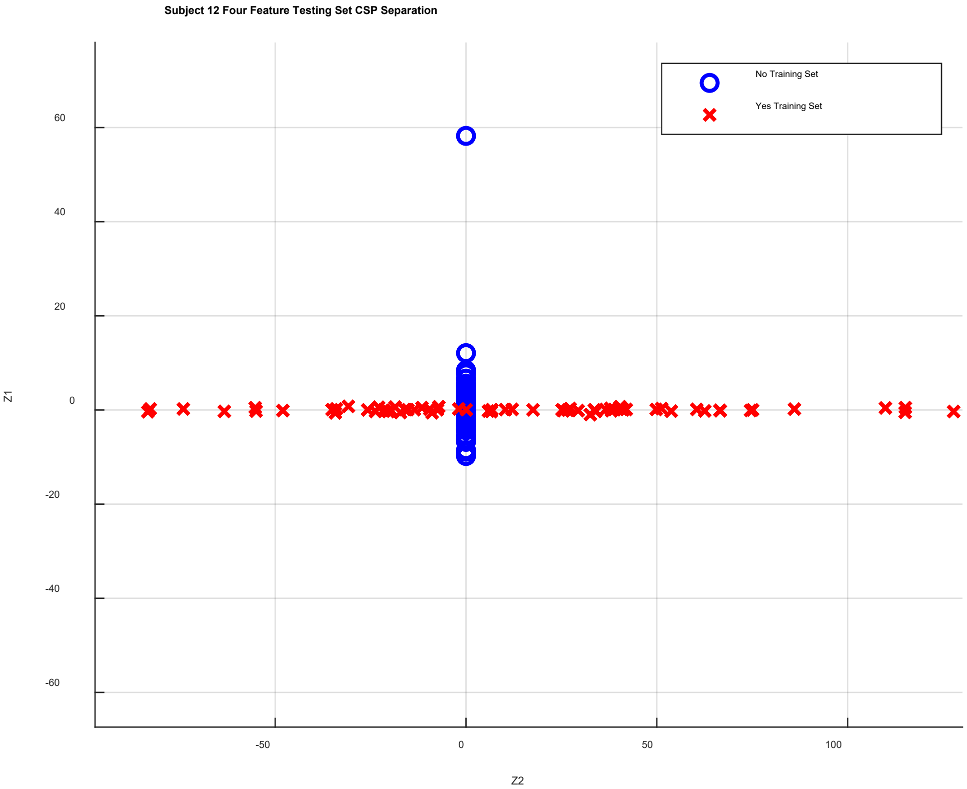
Subject 10 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
150
100
50
0
-50
-100
-150
-200
-250
Z1 & Z3
-200
-100
0
100
200
300
Z2 & Z4

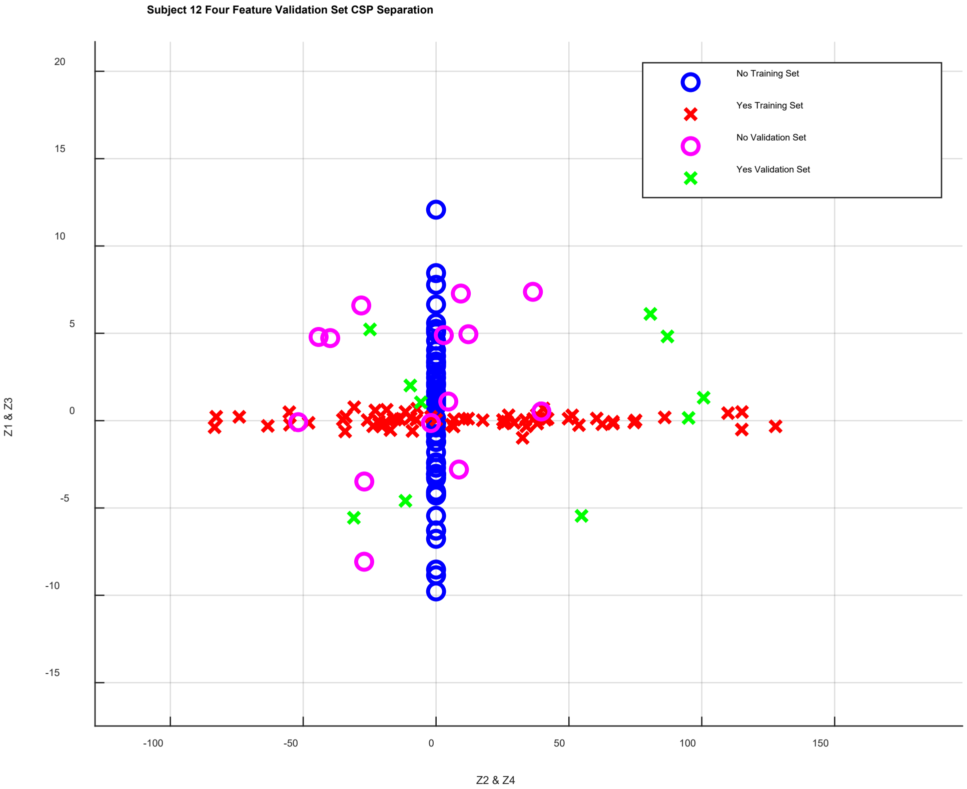

Subject 10 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
60
40
20
0
-20
-40
-60
Z1 & Z3
-60
-40
-20
0
20
40
60
Z2 & Z4

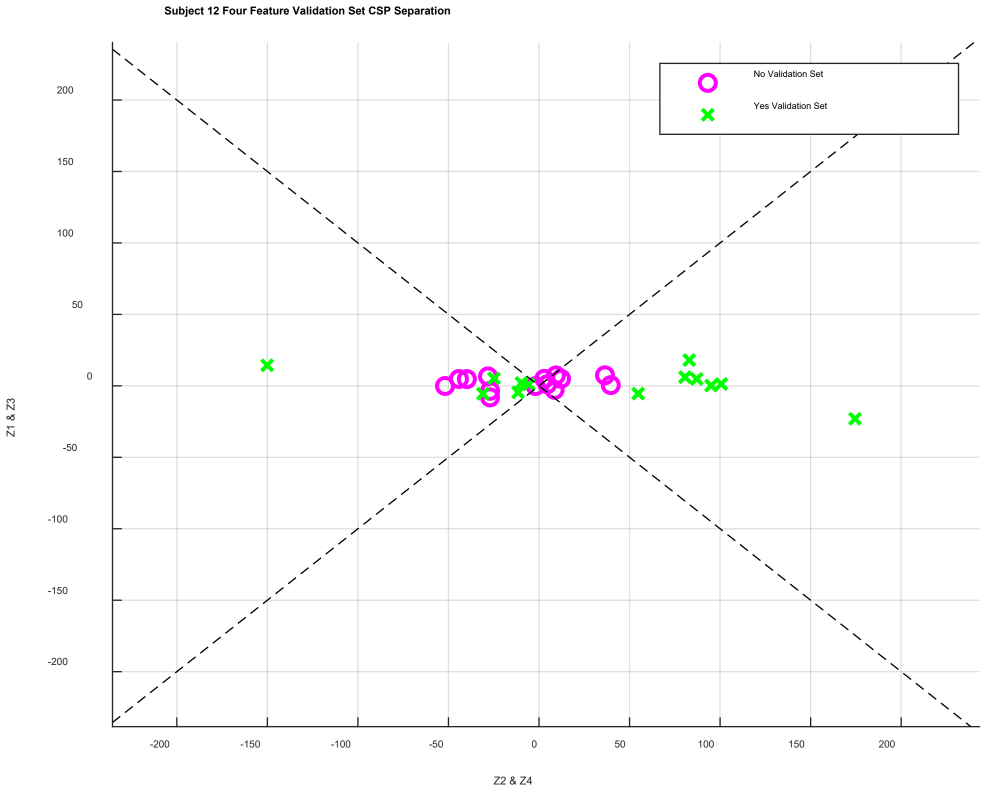

Subject 11 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
20
15
10
5
0
-5
-10
-15
-20
-25
Z1
-20
-10
0
10
20
30
Z2

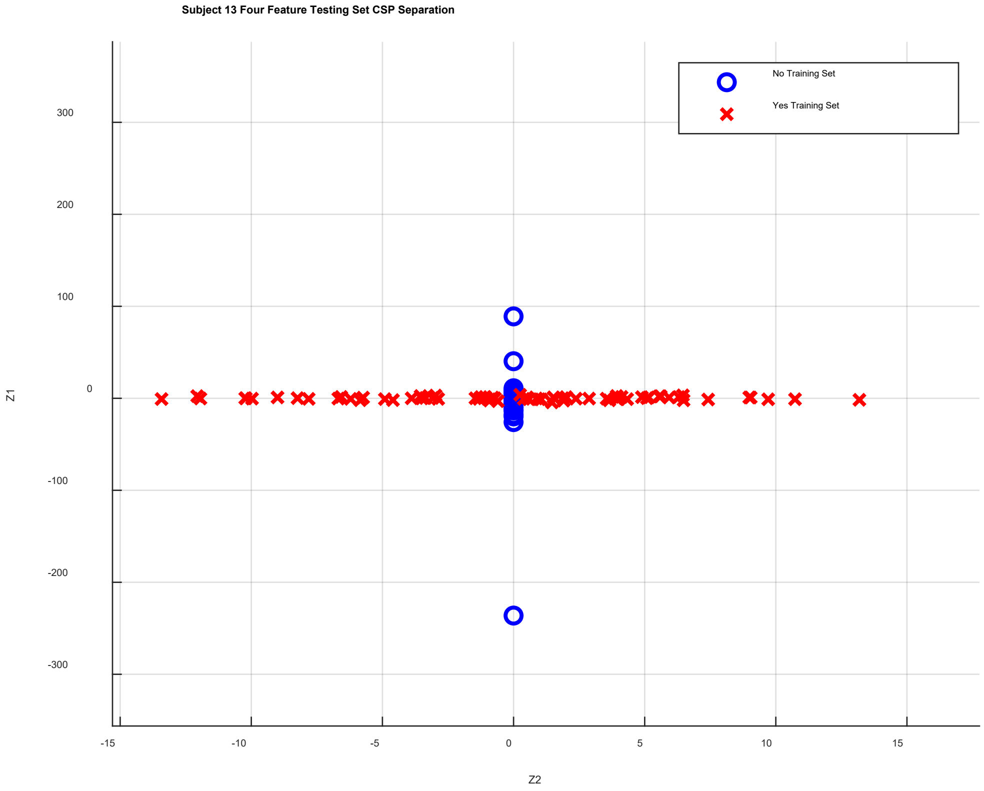

Subject 11 Four Feature Validation Set CSP Separation
25
20
15
10
5
0
-5
-10
-15
-20
Z1 & Z3
-40
-20
0
20
40
60
80
Z2 & Z4
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set

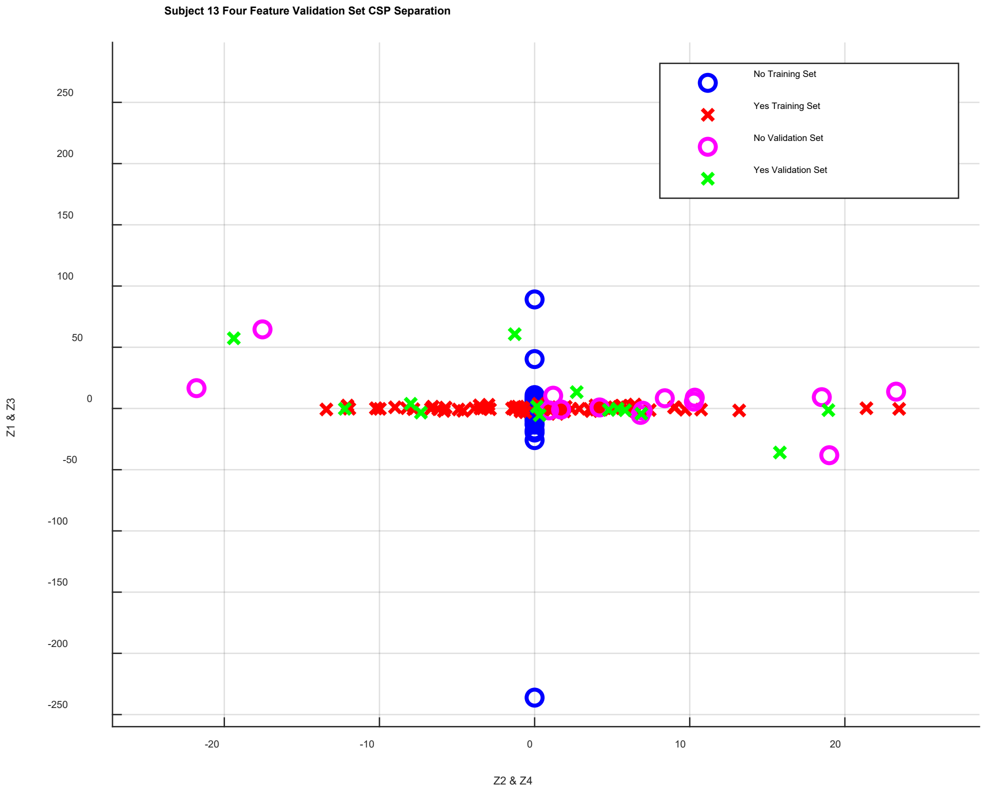

Subject 11 Four Feature Validation Set CSP Separation
50
40
30
20
10
0
-10
-20
-30
-40
-50
Z1 & Z3
-50
0
50
Z2 & Z4
No Validation Set
Yes Validation Set

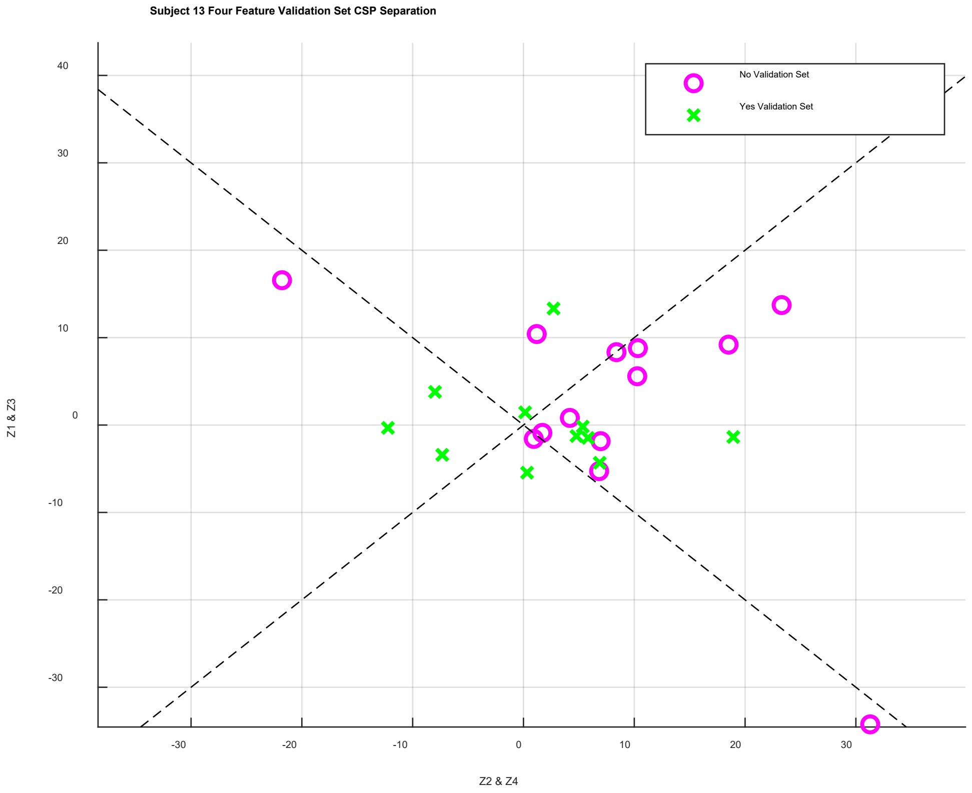

Subject 12 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
60
40
20
0
-20
-40
-60
Z1
-50
0
50
100
Z2


Subject 12 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
20
15
10
5
0
-5
-10
-15
Z1 & Z3
-100
-50
0
50
100
150
Z2 & Z4

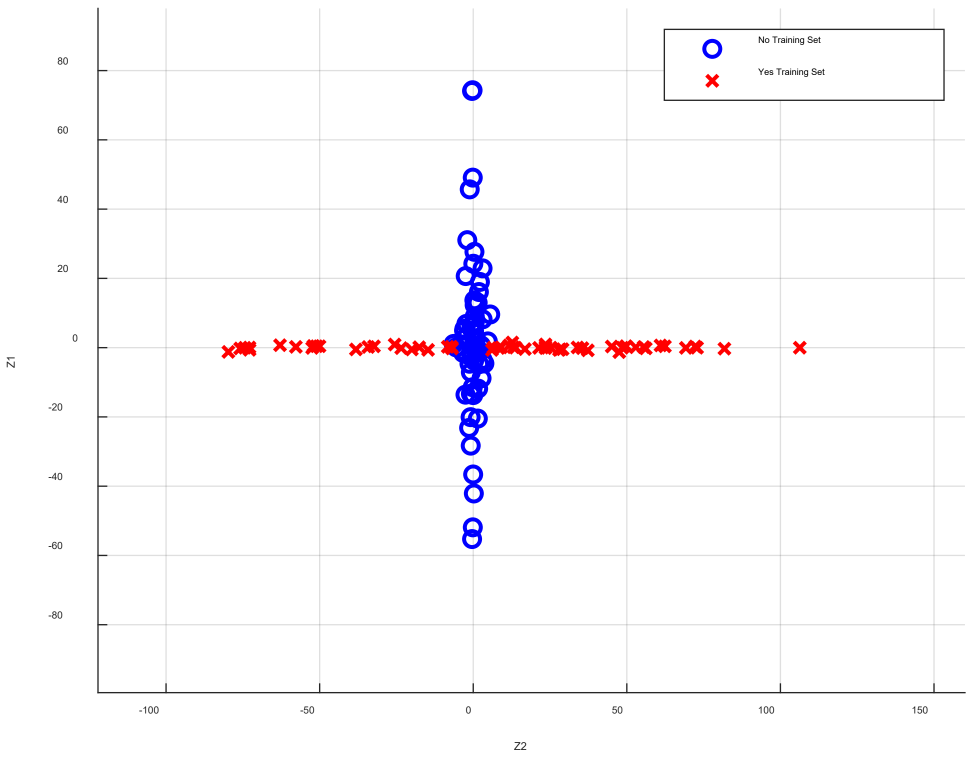

Subject 12 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
200
150
100
50
0
-50
-100
-150
-200
Z1 & Z3
-200
-150
-100
-50
0
50
100
150
200
Z2 & Z4


Subject 13 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
300
200
100
0
-100
-200
-300
Z1
-15
-10
-5
0
5
10
15
Z2

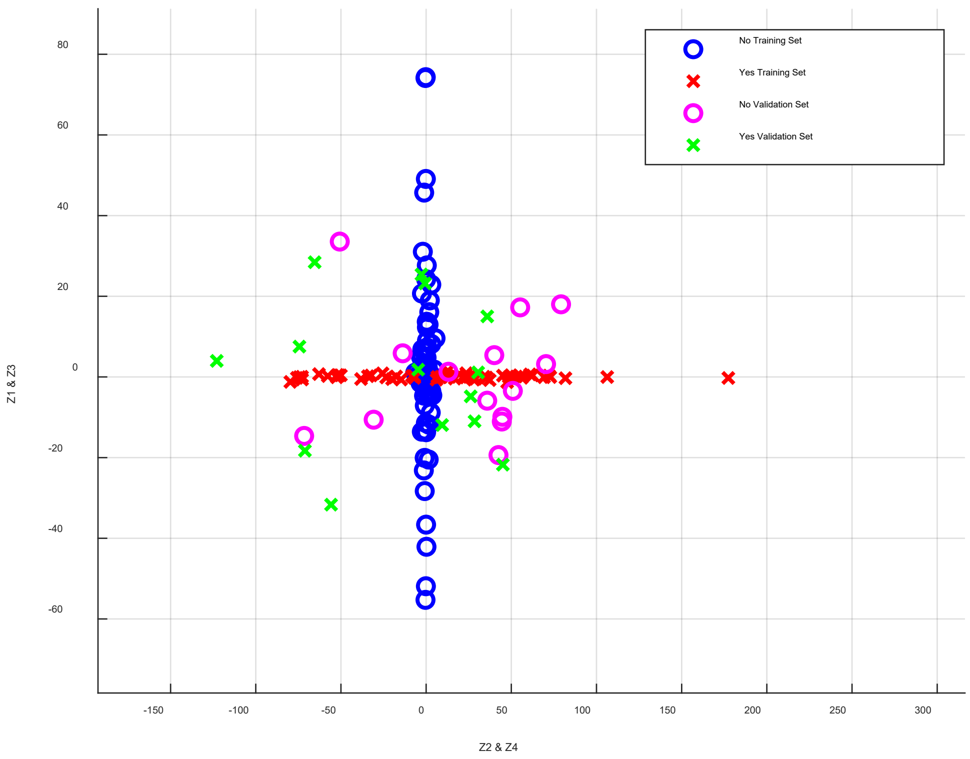

Subject 13 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
250
200
150
100
50
0
-50
-100
-150
-200
-250
Z1 & Z3
-20
-10
0
10
20
Z2 & Z4

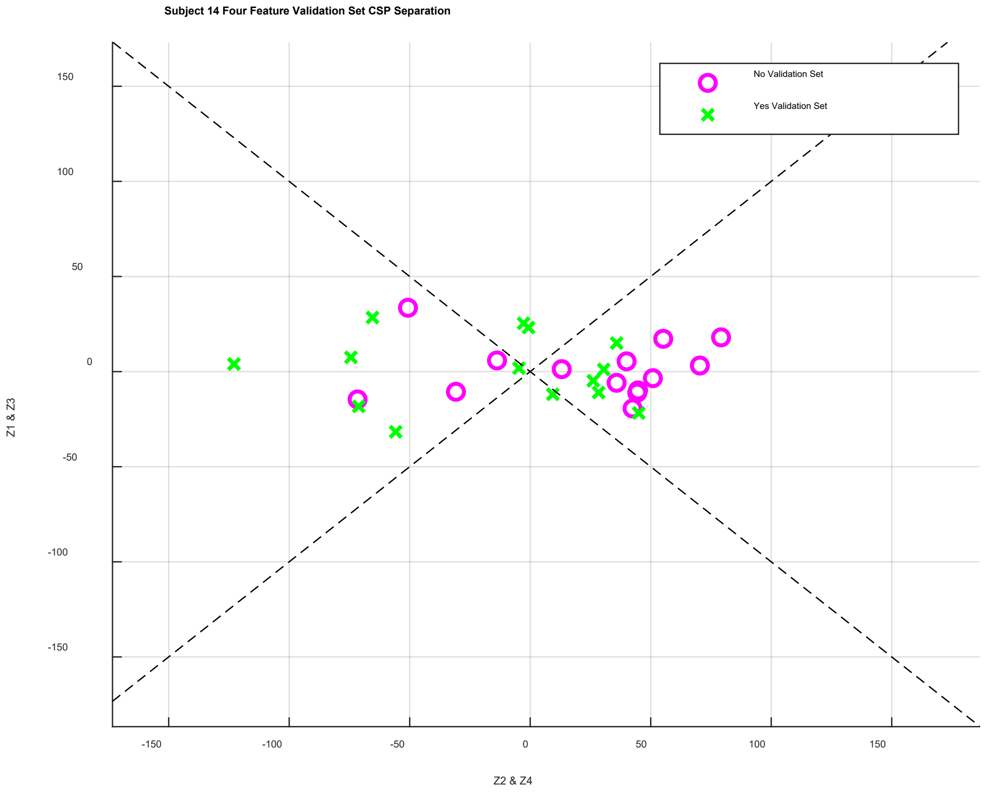

Subject 13 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
40
30
20
10
0
-10
-20
-30
Z1 & Z3
-30
-20
-10
0
10
20
30
Z2 & Z4

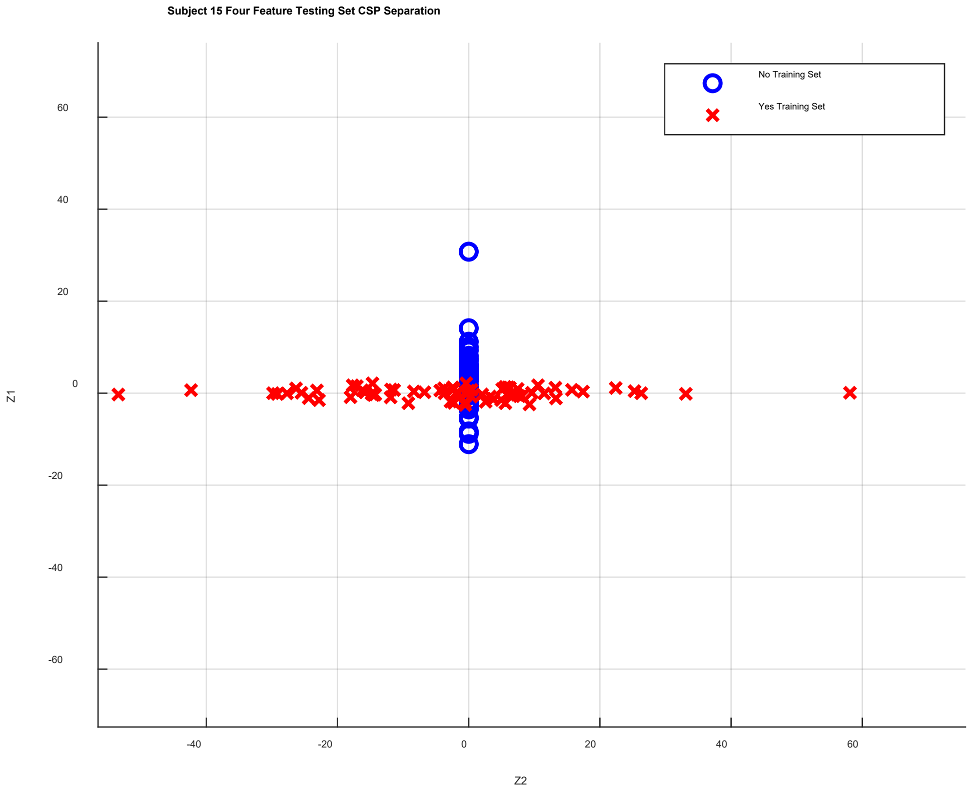
Subject 14 Four Feature Testing Set CSP Separation

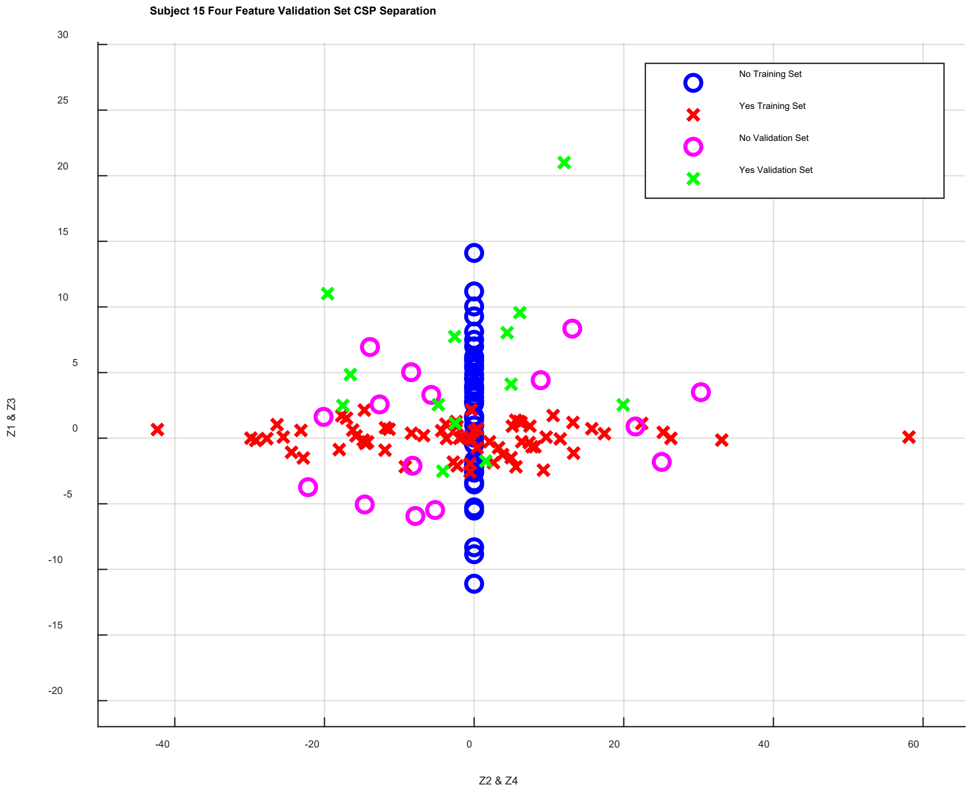


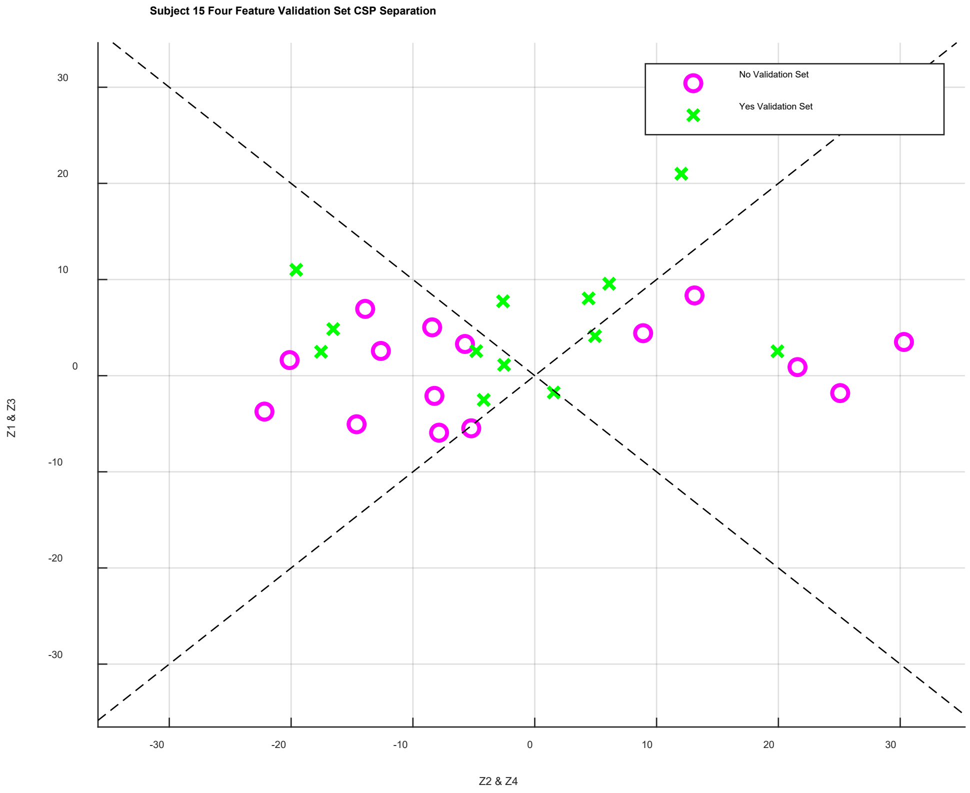
Subject 14 Four Feature Validation Set CSP Separation

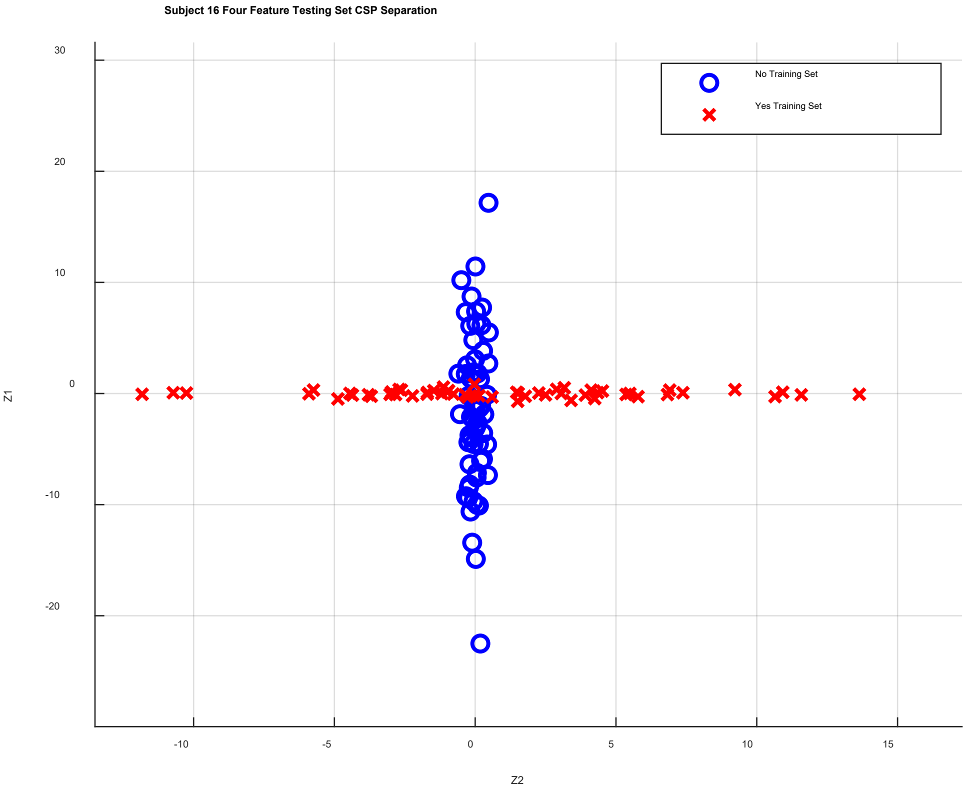


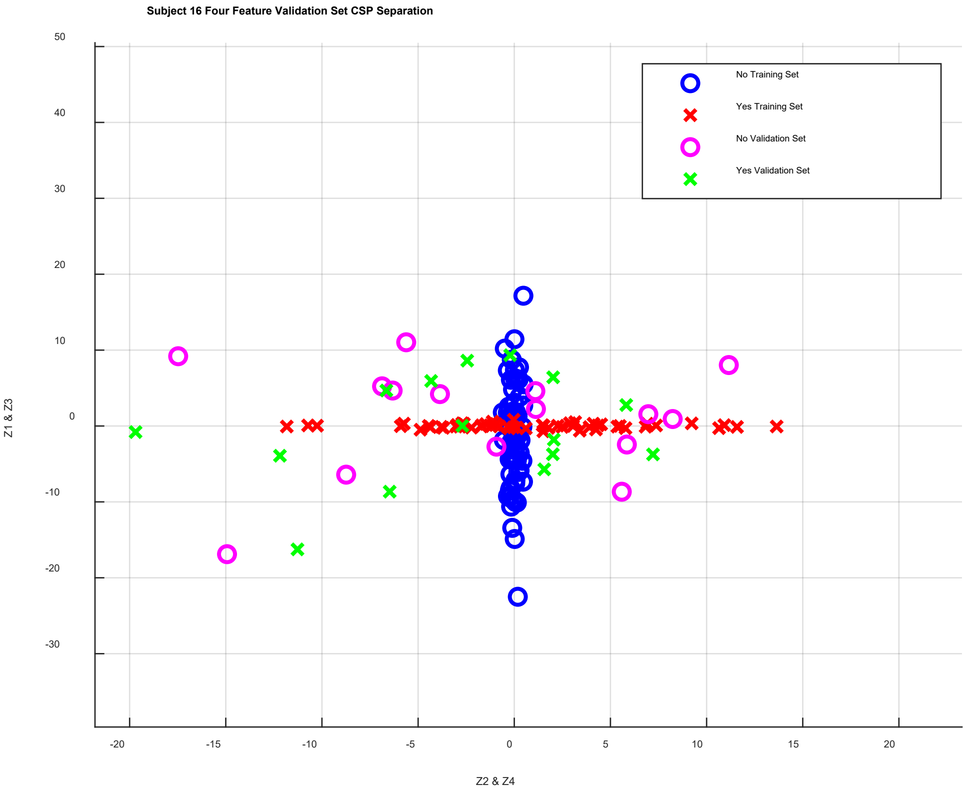

Subject 14 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
150
100
50
0
-50
-100
-150
Z1 & Z3
-150
-100
-50
0
50
100
150
Z2 & Z4


Subject 15 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
60
40
20
0
-20
-40
-60
Z1
-40
-20
0
20
40
60
Z2

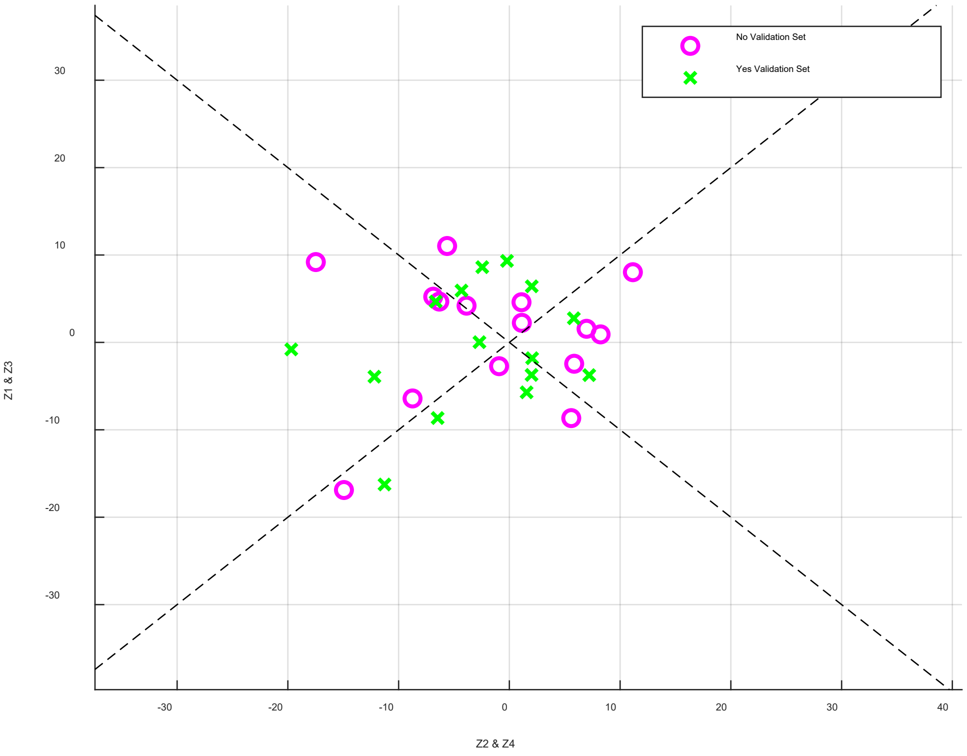

Subject 15 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
30
25
20
15
10
5
0
-5
-10
-15
-20
Z1 & Z3
-40
-20
0
20
40
60
Z2 & Z4


Subject 15 Four Feature Validation Set CSP Separation
No Validation Set
Yes Validation Set
30
20
10
0
-10
-20
-30
Z1 & Z3
-30
-20
-10
0
10
20
30
Z2 & Z4

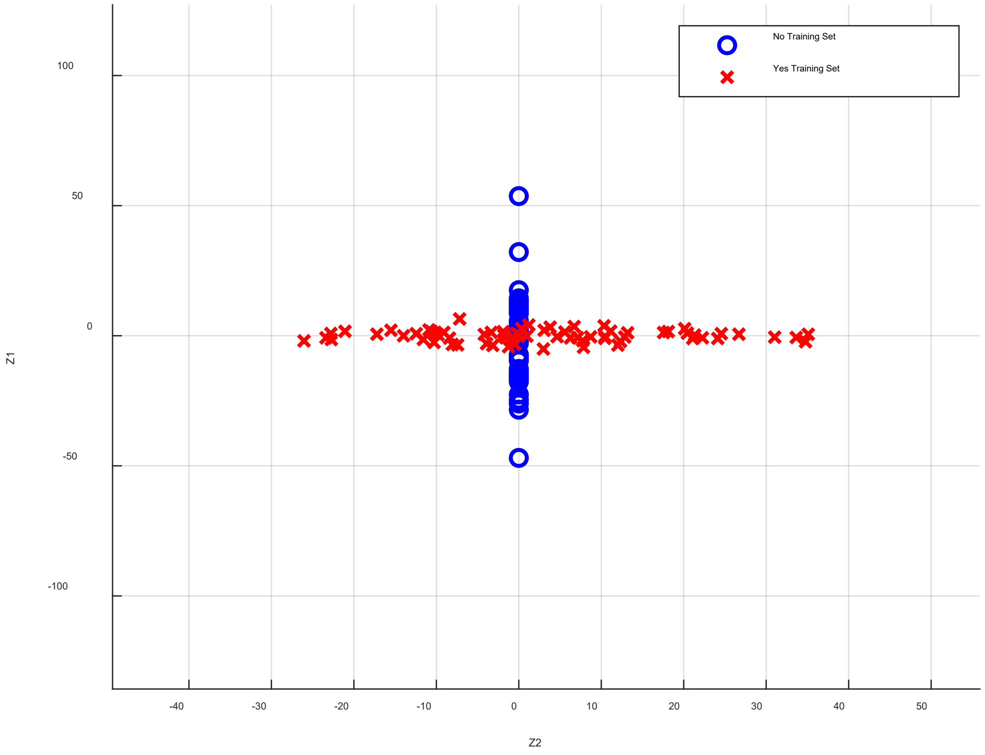

Subject 16 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
30
20
10
0
-10
-20
Z1
-10
-5
0
5
10
15
Z2

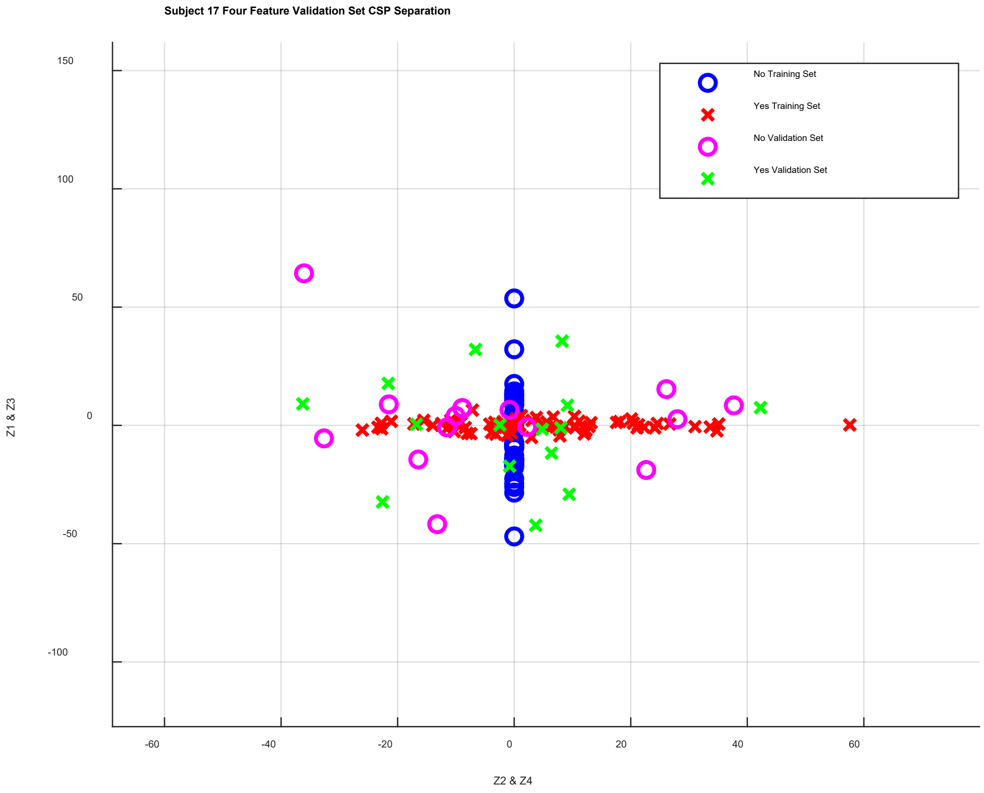

Subject 16 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
50
40
30
20
10
0
-10
-20
-30
Z1 & Z3
-20
-15
-10
-5
0
5
10
15
20
Z2 & Z4

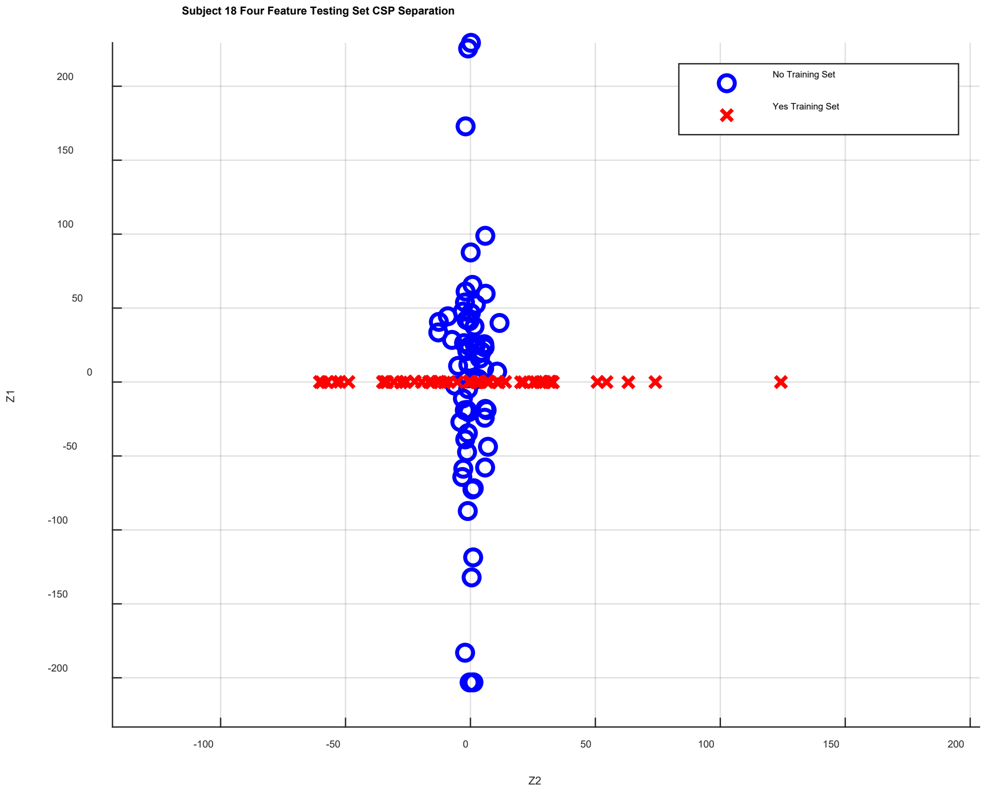
Subject 16 Four Feature Validation Set CSP Separation

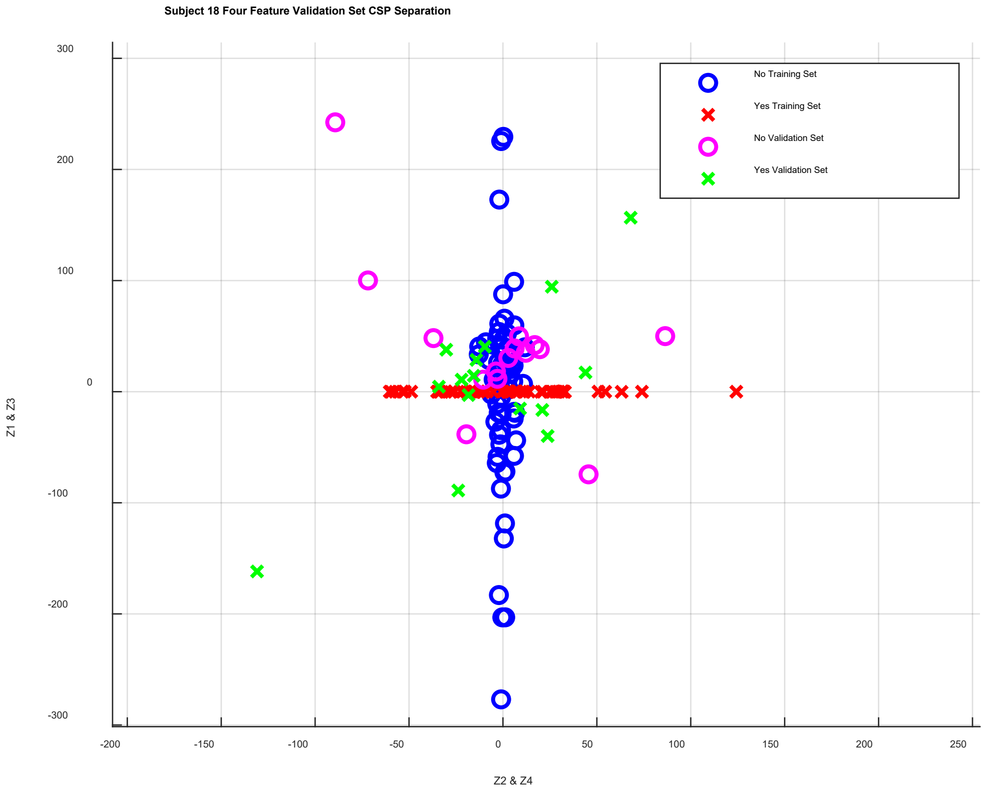


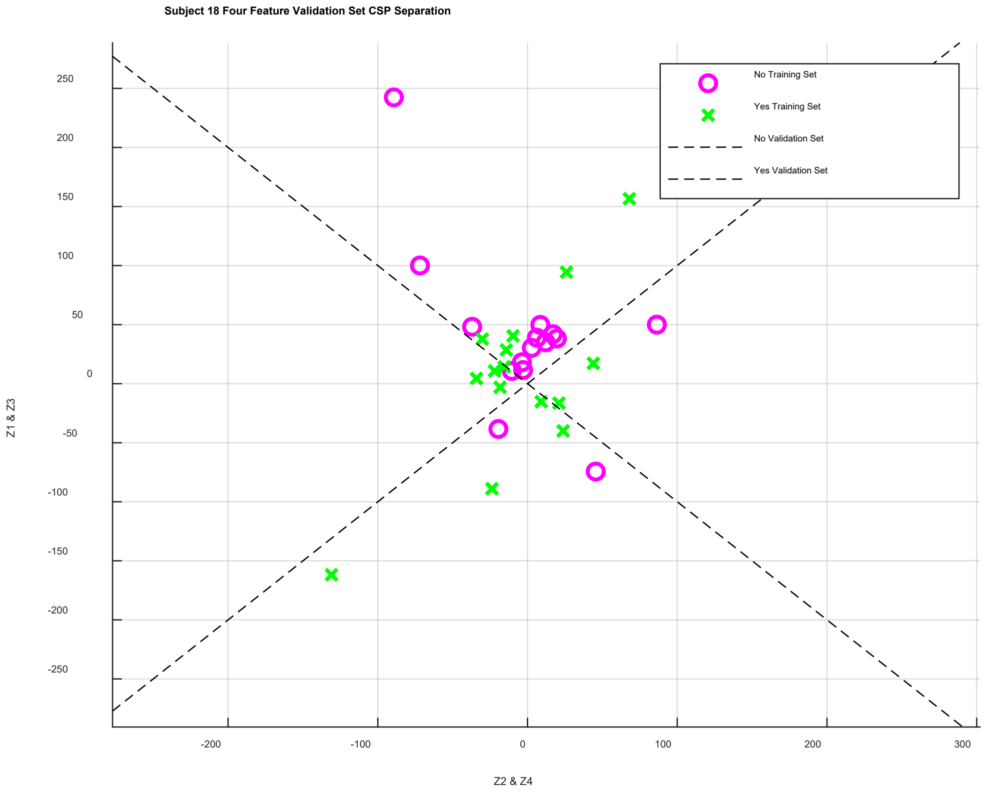
Subject 17 Four Feature Testing Set CSP Separation




Subject 17 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
150
100
50
0
-50
-100
Z1 & Z3
-60
-40
-20
0
20
40
60
Z2 & Z4


Subject 18 Four Feature Testing Set CSP Separation
No Training Set
Yes Training Set
200
150
100
50
0
-50
-100
-150
-200
Z1
-100
-50
0
50
100
150
200
Z2


Subject 18 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
300
200
100
0
-100
-200
-300
Z1 & Z3
-200
-150
-100
-50
0
50
100
150
200
250
Z2 & Z4


Subject 18 Four Feature Validation Set CSP Separation
No Training Set
Yes Training Set
No Validation Set
Yes Validation Set
250
200
150
100
50
0
-50
-100
-150
-200
-250
Z1 & Z3
-200
-100
0
100
200
300
Z2 & Z4

Appendix B

Semantic Violation Pairs
Middle-school reading level
Violation at the end of the sentence (violations are all nouns)
First sentence is the violation, second sentence is the correct version

1.
She flew the flag at the top of the hamburger.
She flew the flag at the top of the flagpole.

2.
He went to school on a February.
He went to school on a Thursday.

3.
The class gathered around the teacher’s purple.
The class gathered around the teacher’s desk.

4.
The blanket was warm and afternoon.
The blanket was warm and soft.

5.
She left her book on the top bicycle.
She left her book on the top shelf.

6.
They came back down the hiking milk.
They came back down the hiking trail.

7.
He went to the barber shop for a salad.
He went to the barber shop for a haircut.

8.
They bought a computer from the pizza.
The bought a computer from the store.

9.
She drinks tea with her toast in the sweater.
She drinks tea with her toast in the morning.

10.
He turned the light on with the light turnip.
He turned the light on with the light switch.

11.
They visited the town fair last cucumber.
They visited the town fair last week.

12.
The navy cruiser sailed across the basketball.
The navy cruiser sailed across the ocean.

13.
The astronaut jumped from the shuttle and landed on the squash.
The astronaut jumped from the shuttle and landed on the moon.

14.
She bought tickets for her favorite rock dishwasher.
She bought tickets for her favorite rock band.

15.
The sun rose in the umbrella.
The sun rose in the east.

16.
He went bird-watching to catch a glimpse of a shark.
He went bird-watching to catch a glimpse of an owl.

17.
They went to the mall to buy a new pair of pancakes.
They went to the mall to buy a new pair of jeans.

18.
He and his father attended a baseball eggplant.
He and his father attended a baseball game.

19.
It was raining, so she grabbed her lamp.
It was raining, so she grabbed her umbrella.

20.
He loves putting chocolate sauce on his television.
He loves putting chocolate sauce on his sundaes.

21.
He likes avocado and red onions on his socks.
He likes avocado and red onions on his toast.

22.
I think I can finish this homework yesterday.
I think I can finish this homework today.

23.
She always likes working out, so she often goes to the pub.
She always likes working out, so she often goes to the gym.

24.
He studies Russian because he wants to go to Australia.
He studies Russian because he wants to go to Russia.

25.
I'm on a diet, but I cannot stop eating food at the parking lot.
I'm on a diet, but I cannot stop eating food at night.

26.
My friend often goes to the theater to watch a cheesecake.
My friend often goes to the theater to watch a movie.

27.
She likes to drink a cup of coffee with some socks.
She likes to drink a cup of coffee with some snacks.

28.
This TV show is the most popular in the textbook.
This TV show is the most popular in the world.

29.
My laptop was broken because I dropped the test.
My laptop was broken because I dropped it.

30.
No student sleeps in water.
No student sleeps in class.

31.
The container has 6 arms.
The container has 6 drawers.

32.
This teacher gives me a lot of HW every single detergent.
This teacher gives me a lot of HW every single class.

33.
Speaking of coffee, Starbucks has many different kinds of shoes.
Speaking of coffee, Starbucks has many different kinds of coffee.

34.
A belt is normally made of eggs.
A belt is normally made of leather.

35.
In the U.S, if you are under 21years old, you cannot drink socks.
In the U.S, if you are under 21years old, you cannot drink alcohol.

36.
Some people prefer to stay home instead of going out because they get tired of socks.
Some people prefer to stay home instead of going out because they get tired of social interaction.

37.
My friend likes to drink wine with a slice of cork.
My friend likes to drink wine with a slice of cheese.

38.
My laptop often shuts down suddenly because of its bad taxi.
My laptop often shuts down suddenly because of its bad battery.

39.
My friend and my brother were born in the same laundry.
My friend and my brother were born in the same country.

40.
My friend broke my glasses finger.
My friend broke my glasses frame.

41.
He turned the corner and went up the pineapple.
He turned the corner and went up the stairs.

42.
Her new bicycle was purple and eggs.
Her new bicycle was purple and black.

43.
The pirate ship sank in the middle of the shoe.
The pirate ship sank in the middle of the sea.

44.
She opened the blinds and saw the octopus.
She opened the blinds and saw the sunrise.

45.
He picked her up for their Saturday movie squirrel.
He picked her up for their Saturday movie date.

46.
The fireplace flickered with orange and red lamppost.
The fireplace flickered with orange and red flames.

47.
He stole a pair of shoes from the local kangaroo.
He stole a pair of shoes from the local mall.

48.
She guarded her treasure chest with her vegetable.
She guarded her treasure chest with her life.

49.
The ski slopes were packed with skiers and turnips.
The ski slopes were packed with skiers and snowboarders.

50.
He went to the airport to catch a late toothbrush.
He went to the airport to catch a late flight.

51.
I looked out the window to see three feet of unicycles.
I looked out the window to see three feet of snow.

52.
With a mighty roar, the lion scared off the eggplant.
With a mighty roar, the lion scared off the zebras.

53.
Her chocolate chip cookies were sweet and orangutan.
Her chocolate chip cookies were sweet and soft.

54.
On Sundays, he did his kettle.
On Sundays, he did his laundry.

55.
They walked across the plaza to the river spaghetti.
They walked across the plaza to the river park.

56.
As a carpenter, he constructed tables and January.
As a carpenter, he constructed tables and chairs.

57.
She was a very successful computer tomato.
She was a very successful computer programmer.

58.
They went swimming on the hottest day of the avocado.
They went swimming on the hottest day of the summer.

59.
He was only late by a few lawnmower.
He was only late by a few minutes.

60.
Her dress featured green and yellow mailboxes.
Her dress featured green and yellow flowers.

Appendix C

```
% Biomedical Engineering Thesis Processing
% Written by: Michael Doyel
% November 20th, 2020
% Thesis Advisor: Dr. Chiu
%
% The data in this module was taken from the paper Measuring Concept
% Semantic Relatedness through Common Spatial Pattern Feature Extraction
% on EEG Signals. This data was used in place of data obtained in the
% labs at Rose-Hulman due to the ongoing COVID-19 crisis. In this module
% the data is parsed and examined for the N400 and P600 waveforms that help
% to identify semantic violation. Frequency domain features such as alpha
% and theta waves are also isolated for CSP analysis. From there we can
% work to use the data to assist in the machine learning of a BCI system.
%
%
% Sampling Frequency is 128Hz and is band limited from 0 to 45 Hz
% Sample 100ms before stimulation and average this value, then adjust
% post stimulus N400 and P600 based on this value. Only 128 samples after
% stimulus need to be viewed 400ms is at 51.2 samples and 600ms is at 76.8
% ms. It may be wise to do 350-550 for N400 meaning average 45 to 71
% samples. P600 will be the average of 77 to 115 samples. Note some sources
% disagree on N400, here we use 350 to 550 samples, but some sources state
% that 250 to 500 is a better standard which would use 32 to 64.
% Total samples analyzed for each word will be 141, 100ms being 13 samples
% before stimulus
%
% For 141 samples N400 is (58 to 84) P600 is(90 to 128)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all

% Load the proper subject data file
load('S1.mat');

% Indicate the subject number for labeling of datasets
S = 1;

% Initialize two vectors for sorting of data
start = zeros(150,1);
type = zeros(150,1);

% This for loops strips off the start times for stimulus & Y/N data
% start(k) contains the start time of every stimulus
% type(k) contains the subject response for each stimulus
for k = 1:150
    eval(['start(k) = str2num(Eventos_S' num2str(S) '{2*k,3});'])
    eval(['type(k) = Eventos_S' num2str(S) '{2*k,5};'])
end

CT1 = 1;
CT2 = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The next five for loops act as a sorting function. They strip away the 13
% samples before a stimulus occurs as well as the 128 samples after the
% stimulus occurs in order to get a full second of data for every stimulus.
% The yes and no responses are placed in separate matrices based on the
% string value found in the type(k) matrix. The five separate data blocks
% are concatenated into two yes and no blocks which contain all 150 word
% responses
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Block One
for k = 1:30
    if type(k) == 83
        eval(['CH1Y(CT1,:) = MuestreoB1_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2Y(CT1,:) = MuestreoB1_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3Y(CT1,:) = MuestreoB1_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4Y(CT1,:) = MuestreoB1_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5Y(CT1,:) = MuestreoB1_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6Y(CT1,:) = MuestreoB1_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7Y(CT1,:) = MuestreoB1_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8Y(CT1,:) = MuestreoB1_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9Y(CT1,:) = MuestreoB1_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10Y(CT1,:) = MuestreoB1_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11Y(CT1,:) = MuestreoB1_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12Y(CT1,:) = MuestreoB1_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13Y(CT1,:) = MuestreoB1_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14Y(CT1,:) = MuestreoB1_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT1 = CT1 + 1;
    elseif type(k) == 78
        eval(['CH1N(CT2,:) = MuestreoB1_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2N(CT2,:) = MuestreoB1_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3N(CT2,:) = MuestreoB1_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4N(CT2,:) = MuestreoB1_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5N(CT2,:) = MuestreoB1_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6N(CT2,:) = MuestreoB1_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7N(CT2,:) = MuestreoB1_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8N(CT2,:) = MuestreoB1_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9N(CT2,:) = MuestreoB1_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10N(CT2,:) = MuestreoB1_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11N(CT2,:) = MuestreoB1_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12N(CT2,:) = MuestreoB1_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13N(CT2,:) = MuestreoB1_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14N(CT2,:) = MuestreoB1_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT2 = CT2 + 1;
    end
end

%Block Two
for k = 31:60
    if type(k) == 83   % This for loop generates the Y and N data strings for the second block
        eval(['CH1Y(CT1,:) = MuestreoB2_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2Y(CT1,:) = MuestreoB2_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3Y(CT1,:) = MuestreoB2_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4Y(CT1,:) = MuestreoB2_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5Y(CT1,:) = MuestreoB2_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6Y(CT1,:) = MuestreoB2_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7Y(CT1,:) = MuestreoB2_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8Y(CT1,:) = MuestreoB2_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9Y(CT1,:) = MuestreoB2_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10Y(CT1,:) = MuestreoB2_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11Y(CT1,:) = MuestreoB2_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12Y(CT1,:) = MuestreoB2_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13Y(CT1,:) = MuestreoB2_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14Y(CT1,:) = MuestreoB2_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT1 = CT1 + 1;
    elseif type(k) == 78
        eval(['CH1N(CT2,:) = MuestreoB2_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2N(CT2,:) = MuestreoB2_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3N(CT2,:) = MuestreoB2_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4N(CT2,:) = MuestreoB2_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5N(CT2,:) = MuestreoB2_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6N(CT2,:) = MuestreoB2_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7N(CT2,:) = MuestreoB2_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8N(CT2,:) = MuestreoB2_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9N(CT2,:) = MuestreoB2_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10N(CT2,:) = MuestreoB2_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11N(CT2,:) = MuestreoB2_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12N(CT2,:) = MuestreoB2_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13N(CT2,:) = MuestreoB2_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14N(CT2,:) = MuestreoB2_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT2 = CT2 + 1;
    end
end

% Block Three
```

```
for k = 61:90
    if type(k) == 83   % This for loop generates the Y and N data strings for the third block
        eval(['CH1Y(CT1,:) = MuestreoB3_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2Y(CT1,:) = MuestreoB3_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3Y(CT1,:) = MuestreoB3_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4Y(CT1,:) = MuestreoB3_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5Y(CT1,:) = MuestreoB3_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6Y(CT1,:) = MuestreoB3_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7Y(CT1,:) = MuestreoB3_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8Y(CT1,:) = MuestreoB3_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9Y(CT1,:) = MuestreoB3_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10Y(CT1,:) = MuestreoB3_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11Y(CT1,:) = MuestreoB3_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12Y(CT1,:) = MuestreoB3_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13Y(CT1,:) = MuestreoB3_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14Y(CT1,:) = MuestreoB3_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT1 = CT1 + 1;
    elseif type(k) == 78
        eval(['CH1N(CT2,:) = MuestreoB3_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2N(CT2,:) = MuestreoB3_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3N(CT2,:) = MuestreoB3_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4N(CT2,:) = MuestreoB3_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5N(CT2,:) = MuestreoB3_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6N(CT2,:) = MuestreoB3_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7N(CT2,:) = MuestreoB3_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8N(CT2,:) = MuestreoB3_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9N(CT2,:) = MuestreoB3_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10N(CT2,:) = MuestreoB3_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11N(CT2,:) = MuestreoB3_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12N(CT2,:) = MuestreoB3_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13N(CT2,:) = MuestreoB3_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14N(CT2,:) = MuestreoB3_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT2 = CT2 + 1;
    end
end

% Block Four
for k = 91:120
    if type(k) == 83   % This for loop generates the Y and N data strings for the fourth block
        eval(['CH1Y(CT1,:) = MuestreoB4_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2Y(CT1,:) = MuestreoB4_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3Y(CT1,:) = MuestreoB4_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4Y(CT1,:) = MuestreoB4_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5Y(CT1,:) = MuestreoB4_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6Y(CT1,:) = MuestreoB4_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7Y(CT1,:) = MuestreoB4_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8Y(CT1,:) = MuestreoB4_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9Y(CT1,:) = MuestreoB4_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10Y(CT1,:) = MuestreoB4_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11Y(CT1,:) = MuestreoB4_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12Y(CT1,:) = MuestreoB4_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13Y(CT1,:) = MuestreoB4_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14Y(CT1,:) = MuestreoB4_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT1 = CT1 + 1;
    elseif type(k) == 78
        eval(['CH1N(CT2,:) = MuestreoB4_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2N(CT2,:) = MuestreoB4_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3N(CT2,:) = MuestreoB4_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4N(CT2,:) = MuestreoB4_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5N(CT2,:) = MuestreoB4_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6N(CT2,:) = MuestreoB4_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7N(CT2,:) = MuestreoB4_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8N(CT2,:) = MuestreoB4_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9N(CT2,:) = MuestreoB4_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10N(CT2,:) = MuestreoB4_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11N(CT2,:) = MuestreoB4_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12N(CT2,:) = MuestreoB4_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13N(CT2,:) = MuestreoB4_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14N(CT2,:) = MuestreoB4_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT2 = CT2 + 1;
    end
```

```
end

% Block Five
for k = 121:150
    if type(k) == 83   % This for loop generates the Y and N data strings for the fifth block
        eval(['CH1Y(CT1,:) = MuestreoB5_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2Y(CT1,:) = MuestreoB5_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3Y(CT1,:) = MuestreoB5_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4Y(CT1,:) = MuestreoB5_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5Y(CT1,:) = MuestreoB5_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6Y(CT1,:) = MuestreoB5_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7Y(CT1,:) = MuestreoB5_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8Y(CT1,:) = MuestreoB5_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9Y(CT1,:) = MuestreoB5_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10Y(CT1,:) = MuestreoB5_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11Y(CT1,:) = MuestreoB5_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12Y(CT1,:) = MuestreoB5_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13Y(CT1,:) = MuestreoB5_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14Y(CT1,:) = MuestreoB5_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT1 = CT1 + 1;
    elseif type(k) == 78
        eval(['CH1N(CT2,:) = MuestreoB5_S' num2str(S) '(1, start(k)-13:start(k)+127);']);
        eval(['CH2N(CT2,:) = MuestreoB5_S' num2str(S) '(2, start(k)-13:start(k)+127);']);
        eval(['CH3N(CT2,:) = MuestreoB5_S' num2str(S) '(3, start(k)-13:start(k)+127);']);
        eval(['CH4N(CT2,:) = MuestreoB5_S' num2str(S) '(4, start(k)-13:start(k)+127);']);
        eval(['CH5N(CT2,:) = MuestreoB5_S' num2str(S) '(5, start(k)-13:start(k)+127);']);
        eval(['CH6N(CT2,:) = MuestreoB5_S' num2str(S) '(6, start(k)-13:start(k)+127);']);
        eval(['CH7N(CT2,:) = MuestreoB5_S' num2str(S) '(7, start(k)-13:start(k)+127);']);
        eval(['CH8N(CT2,:) = MuestreoB5_S' num2str(S) '(8, start(k)-13:start(k)+127);']);
        eval(['CH9N(CT2,:) = MuestreoB5_S' num2str(S) '(9, start(k)-13:start(k)+127);']);
        eval(['CH10N(CT2,:) = MuestreoB5_S' num2str(S) '(10, start(k)-13:start(k)+127);']);
        eval(['CH11N(CT2,:) = MuestreoB5_S' num2str(S) '(11, start(k)-13:start(k)+127);']);
        eval(['CH12N(CT2,:) = MuestreoB5_S' num2str(S) '(12, start(k)-13:start(k)+127);']);
        eval(['CH13N(CT2,:) = MuestreoB5_S' num2str(S) '(13, start(k)-13:start(k)+127);']);
        eval(['CH14N(CT2,:) = MuestreoB5_S' num2str(S) '(14, start(k)-13:start(k)+127);']);
        CT2 = CT2 + 1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% These for loops allow for the baseline averaging of the yes and no
% responses based on the 13 samples of data that occured before the onset
% of the stimulus.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Baseline of Yes reponses
for n = 1:(CT1-1)
    Baseline_CH1Y(n,1:141) = mean(CH1Y(n,1:13)); %Baseline average of first 13 samples of noise
    Baseline_CH2Y(n,1:141) = mean(CH2Y(n,1:13));
    Baseline_CH3Y(n,1:141) = mean(CH3Y(n,1:13));
    Baseline_CH4Y(n,1:141) = mean(CH4Y(n,1:13));
    Baseline_CH5Y(n,1:141) = mean(CH5Y(n,1:13));
    Baseline_CH6Y(n,1:141) = mean(CH6Y(n,1:13));
    Baseline_CH7Y(n,1:141) = mean(CH7Y(n,1:13));
    Baseline_CH8Y(n,1:141) = mean(CH8Y(n,1:13));
    Baseline_CH9Y(n,1:141) = mean(CH9Y(n,1:13));
    Baseline_CH10Y(n,1:141) = mean(CH10Y(n,1:13));
    Baseline_CH11Y(n,1:141) = mean(CH11Y(n,1:13));
    Baseline_CH12Y(n,1:141) = mean(CH12Y(n,1:13));
    Baseline_CH13Y(n,1:141) = mean(CH13Y(n,1:13));
    Baseline_CH14Y(n,1:141) = mean(CH14Y(n,1:13));
end

%Baseline of No responses
for m = 1:(CT2 - 1)
    Baseline_CH1N(m,1:141) = mean(CH1N(m,1:13));
    Baseline_CH2N(m,1:141) = mean(CH2N(m,1:13));
    Baseline_CH3N(m,1:141) = mean(CH3N(m,1:13));
    Baseline_CH4N(m,1:141) = mean(CH4N(m,1:13));
    Baseline_CH5N(m,1:141) = mean(CH5N(m,1:13));
    Baseline_CH6N(m,1:141) = mean(CH6N(m,1:13));
    Baseline_CH7N(m,1:141) = mean(CH7N(m,1:13));
    Baseline_CH8N(m,1:141) = mean(CH8N(m,1:13));
    Baseline_CH9N(m,1:141) = mean(CH9N(m,1:13));
    Baseline_CH10N(m,1:141) = mean(CH10N(m,1:13));
    Baseline_CH11N(m,1:141) = mean(CH11N(m,1:13));
    Baseline_CH12N(m,1:141) = mean(CH12N(m,1:13));
    Baseline_CH13N(m,1:141) = mean(CH13N(m,1:13));
    Baseline_CH14N(m,1:141) = mean(CH14N(m,1:13));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Here each channel is baseline averaged
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

AVGBaselineCH1Y = Baseline_CH1Y - CH1Y;
AVGBaselineCH1N = Baseline_CH1N - CH1N;
AVGBaselineCH2Y = Baseline_CH2Y - CH2Y;
AVGBaselineCH2N = Baseline_CH2N - CH2N;
AVGBaselineCH3Y = Baseline_CH3Y - CH3Y;
AVGBaselineCH3N = Baseline_CH3N - CH3N;
AVGBaselineCH4Y = Baseline_CH4Y - CH4Y;
AVGBaselineCH4N = Baseline_CH4N - CH4N;
AVGBaselineCH5Y = Baseline_CH5Y - CH5Y;
AVGBaselineCH5N = Baseline_CH5N - CH5N;
AVGBaselineCH6Y = Baseline_CH6Y - CH6Y;
AVGBaselineCH6N = Baseline_CH6N - CH6N;
AVGBaselineCH7Y = Baseline_CH7Y - CH7Y;
AVGBaselineCH7N = Baseline_CH7N - CH7N;
AVGBaselineCH8Y = Baseline_CH8Y - CH8Y;
AVGBaselineCH8N = Baseline_CH8N - CH8N;
AVGBaselineCH9Y = Baseline_CH9Y - CH9Y;
AVGBaselineCH9N = Baseline_CH9N - CH9N;
AVGBaselineCH10Y = Baseline_CH10Y - CH10Y;
AVGBaselineCH10N = Baseline_CH10N - CH10N;
AVGBaselineCH11Y = Baseline_CH11Y - CH11Y;
AVGBaselineCH11N = Baseline_CH11N - CH11N;
AVGBaselineCH12Y = Baseline_CH12Y - CH12Y;
AVGBaselineCH12N = Baseline_CH12N - CH12N;
AVGBaselineCH13Y = Baseline_CH13Y - CH13Y;
AVGBaselineCH13N = Baseline_CH13N - CH13N;
AVGBaselineCH14Y = Baseline_CH14Y - CH14Y;
AVGBaselineCH14N = Baseline_CH14N - CH14N;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% In this section two 3D matrices are generated that hold 14 channel
% response data for each word across the entire 1s duration of the
% partitioned time. the first dimension is the 14 channels, the second
% dimension is the stimulus sampling, the third dimension is the particular
% trial being sampled. This section also generates the average values for
% P600 and N400 for each trial and places them in their respective
% matrices.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

YesBlock = zeros(14,141, size(AVGBaselineCH1Y,1));
NoBlock = zeros (14,141, size(AVGBaselineCH1N,1));
N400No = zeros(14, size(AVGBaselineCH1N,1));
N400Yes = zeros(14, size(AVGBaselineCH1Y,1));
P600No = zeros(14, size(AVGBaselineCH1N,1));
P600Yes = zeros(14, size(AVGBaselineCH1Y,1));

for k = 1:size(AVGBaselineCH1Y,1)
        YesBlock(1,:,k) = AVGBaselineCH1Y(k,:);
        YesBlock(2,:,k) = AVGBaselineCH2Y(k,:);
        YesBlock(3,:,k) = AVGBaselineCH3Y(k,:);
        YesBlock(4,:,k) = AVGBaselineCH4Y(k,:);
        YesBlock(5,:,k) = AVGBaselineCH5Y(k,:);
        YesBlock(6,:,k) = AVGBaselineCH6Y(k,:);
        YesBlock(7,:,k) = AVGBaselineCH7Y(k,:);
        YesBlock(8,:,k) = AVGBaselineCH8Y(k,:);
        YesBlock(9,:,k) = AVGBaselineCH9Y(k,:);
        YesBlock(10,:,k) = AVGBaselineCH10Y(k,:);
        YesBlock(11,:,k) = AVGBaselineCH11Y(k,:);
        YesBlock(12,:,k) = AVGBaselineCH12Y(k,:);
        YesBlock(13,:,k) = AVGBaselineCH13Y(k,:);
        YesBlock(14,:,k) = AVGBaselineCH14Y(k,:);
        N400Yes(1,k) = mean(AVGBaselineCH1Y(k,58:84));
        N400Yes(2,k) = mean(AVGBaselineCH2Y(k,58:84));
        N400Yes(3,k) = mean(AVGBaselineCH3Y(k,58:84));
        N400Yes(4,k) = mean(AVGBaselineCH4Y(k,58:84));
        N400Yes(5,k) = mean(AVGBaselineCH5Y(k,58:84));
        N400Yes(6,k) = mean(AVGBaselineCH6Y(k,58:84));
        N400Yes(7,k) = mean(AVGBaselineCH7Y(k,58:84));
        N400Yes(8,k) = mean(AVGBaselineCH8Y(k,58:84));
        N400Yes(9,k) = mean(AVGBaselineCH9Y(k,58:84));
        N400Yes(10,k) = mean(AVGBaselineCH10Y(k,58:84));
        N400Yes(11,k) = mean(AVGBaselineCH11Y(k,58:84));
        N400Yes(12,k) = mean(AVGBaselineCH12Y(k,58:84));
        N400Yes(13,k) = mean(AVGBaselineCH13Y(k,58:84));
        N400Yes(14,k) = mean(AVGBaselineCH14Y(k,58:84));
        P600Yes(1,k) = mean(AVGBaselineCH1Y(k,90:128));
        P600Yes(2,k) = mean(AVGBaselineCH2Y(k,90:128));
        P600Yes(3,k) = mean(AVGBaselineCH3Y(k,90:128));
        P600Yes(4,k) = mean(AVGBaselineCH4Y(k,90:128));
        P600Yes(5,k) = mean(AVGBaselineCH5Y(k,90:128));
        P600Yes(6,k) = mean(AVGBaselineCH6Y(k,90:128));
        P600Yes(7,k) = mean(AVGBaselineCH7Y(k,90:128));
        P600Yes(8,k) = mean(AVGBaselineCH8Y(k,90:128));
        P600Yes(9,k) = mean(AVGBaselineCH9Y(k,90:128));
        P600Yes(10,k) = mean(AVGBaselineCH10Y(k,90:128));
        P600Yes(11,k) = mean(AVGBaselineCH11Y(k,90:128));
        P600Yes(12,k) = mean(AVGBaselineCH12Y(k,90:128));
        P600Yes(13,k) = mean(AVGBaselineCH13Y(k,90:128));
        P600Yes(14,k) = mean(AVGBaselineCH14Y(k,90:128));
end

for k = 1:size(AVGBaselineCH1N,1)
        NoBlock(1,:,k) = AVGBaselineCH1N(k,:);
        NoBlock(2,:,k) = AVGBaselineCH2N(k,:);
        NoBlock(3,:,k) = AVGBaselineCH3N(k,:);
        NoBlock(4,:,k) = AVGBaselineCH4N(k,:);
        NoBlock(5,:,k) = AVGBaselineCH5N(k,:);
        NoBlock(6,:,k) = AVGBaselineCH6N(k,:);
        NoBlock(7,:,k) = AVGBaselineCH7N(k,:);
        NoBlock(8,:,k) = AVGBaselineCH8N(k,:);
        NoBlock(9,:,k) = AVGBaselineCH9N(k,:);
        NoBlock(10,:,k) = AVGBaselineCH10N(k,:);
        NoBlock(11,:,k) = AVGBaselineCH11N(k,:);
        NoBlock(12,:,k) = AVGBaselineCH12N(k,:);
        NoBlock(13,:,k) = AVGBaselineCH13N(k,:);
        NoBlock(14,:,k) = AVGBaselineCH14N(k,:);
        N400No(1,k) = mean(AVGBaselineCH1N(k,58:84));
        N400No(2,k) = mean(AVGBaselineCH2N(k,58:84));
        N400No(3,k) = mean(AVGBaselineCH3N(k,58:84));
        N400No(4,k) = mean(AVGBaselineCH4N(k,58:84));
        N400No(5,k) = mean(AVGBaselineCH5N(k,58:84));
        N400No(6,k) = mean(AVGBaselineCH6N(k,58:84));
        N400No(7,k) = mean(AVGBaselineCH7N(k,58:84));
        N400No(8,k) = mean(AVGBaselineCH8N(k,58:84));
        N400No(9,k) = mean(AVGBaselineCH9N(k,58:84));
        N400No(10,k) = mean(AVGBaselineCH10N(k,58:84));
        N400No(11,k) = mean(AVGBaselineCH11N(k,58:84));
        N400No(12,k) = mean(AVGBaselineCH12N(k,58:84));
        N400No(13,k) = mean(AVGBaselineCH13N(k,58:84));
        N400No(14,k) = mean(AVGBaselineCH14N(k,58:84));
        P600No(1,k) = mean(AVGBaselineCH1N(k,90:128));
        P600No(2,k) = mean(AVGBaselineCH2N(k,90:128));
        P600No(3,k) = mean(AVGBaselineCH3N(k,90:128));
        P600No(4,k) = mean(AVGBaselineCH4N(k,90:128));
        P600No(5,k) = mean(AVGBaselineCH5N(k,90:128));
        P600No(6,k) = mean(AVGBaselineCH6N(k,90:128));
        P600No(7,k) = mean(AVGBaselineCH7N(k,90:128));
        P600No(8,k) = mean(AVGBaselineCH8N(k,90:128));
        P600No(9,k) = mean(AVGBaselineCH9N(k,90:128));
        P600No(10,k) = mean(AVGBaselineCH10N(k,90:128));
        P600No(11,k) = mean(AVGBaselineCH11N(k,90:128));
        P600No(12,k) = mean(AVGBaselineCH12N(k,90:128));
        P600No(13,k) = mean(AVGBaselineCH13N(k,90:128));
        P600No(14,k) = mean(AVGBaselineCH14N(k,90:128));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This section takes an FFT of the averaged dataset in order to determine
% the alpha and theta values present in the data for CSP processing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fs = 128;            % Sampling frequency
T = 1/Fs;             % Sampling period
L = 141;             % Length of signal
t = (0:L-1)*T;        % Time vector

freqdatayes = fft(YesBlock,[],2);
freqdatano = fft(NoBlock,[],2);

alphayes = zeros(14,size(AVGBaselineCH1Y,1));
alphano = zeros(14,size(AVGBaselineCH1N,1));
thetayes = zeros(14,size(AVGBaselineCH1Y,1));
thetano = zeros(14,size(AVGBaselineCH1N,1));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The power spectrum is computed and then averaged for storage in a matrix.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k = 1:size(AVGBaselineCH1Y,1);
    for z = 1:14;

        P2 = abs(freqdatayes(z,:,k)/L);
        alphayes(z,k) = mean(P2(8:12));
        thetayes(z,k) = mean(P2(4:7));
    end
end

for k = 1:size(AVGBaselineCH1N,1);
    for z = 1:14;

        P3 = abs(freqdatano(z,:,k)/L);
        alphano(z,k) = mean(P3(8:12));
        thetano(z,k) = mean(P3(4:7));
    end
end

% Here all of the variables are saved to be used in the CSP function.
save ('S1YesBlock', 'YesBlock','N400Yes','P600Yes','alphayes','thetayes');
save ('S1NoBlock', 'NoBlock','N400No','P600No','alphano','thetano');






%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Biomedical Engineering Thesis CSP Filtering
% Written by: Michael Doyel
% December 5th, 2020
% Thesis Advisor: Dr. Chiu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
close all

% Load data for specific subject analysis
load S1YesBlock.mat
load S1NoBlock.mat


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Theta Training Set (4/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M1 = thetano(:,1:62);
M2 = thetayes(:,1:58);

%Initialize:
R1 = M1* M1';
R2 = M2 * M2';
%Calculate:

R1 = R1/trace(R1);
R2 = R2/trace(R2);
Rsum=R1+R2;

[EVecsum,EValsum]=eig(Rsum);
[EValsum,ind] = sort(diag(EValsum), 'descend');
EVecsum=EVecsum(:,ind);

W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix

S1=W*R1*W';
S2=W*R2*W';

[B,D]=eig(S1,S2);
[D,ind]=sort(diag(D));
B=B(: ,[ind(1),ind(14)]); %To look at 1 and 2 change ind to ind1:2
MCSP=B'*W;

Z1 = MCSP*M1;
Z2 = MCSP*M2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Theta Evaluation Set (1/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M3 = thetano(:,63:size(thetano,2));
M4 = thetayes(:,59:size(thetayes,2));
Z3 = MCSP*M3;
Z4 = MCSP*M4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Theta Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% figure(1)
% hold on
% grid on
% scatter(y1(1,:),y1(2,:),'ob','Linewidth',1.5);
% scatter(y2(1,:),y2(2,:),'xr','Linewidth',1.5);
% title('Example Data Plotting Mx vs My');
% ylabel('Channel 1');
% xlabel('Channel 2');
% legend('Mx Example Set', 'My Example Set','Location','northeast');
% hold off
%
% figure(2)
% hold on
% grid on
% scatter(y1(1,:),y1(3,:),'ob','Linewidth',1.5);
% scatter(y2(1,:),y2(3,:),'xr','Linewidth',1.5);
% title('Example Data Plotting Mx vs My');
% ylabel('Channel 1');
% xlabel('Channel 3');
% legend('Mx Example Set', 'My Example Set','Location','northeast');
% hold off
%
% figure(3)
% hold on
% grid on
% scatter(y1(2,:),y1(3,:),'ob','Linewidth',1.5);
% scatter(y2(2,:),y2(3,:),'xr','Linewidth',1.5);
% title('Example Data Plotting Mx vs My');
% ylabel('Channel 2');
% xlabel('Channel 3');
% legend('Mx Example Set', 'My Example Set','Location','northeast');
% hold off
%
% figure(4)
% hold on
% grid on
% scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
% scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
% title('Example Data Plotting Z1 vs Z2');
% ylabel('Z1');
% xlabel('Z1');
% legend('Mx Example Set', 'My Example Set','Location','northeast');
% hold off


figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
title('Theta Wave Testing Set CSP Separation');
ylabel('Z1');
xlabel('Z2');
legend('Theta No Training Set', 'Theta Yes Training Set','Location','northeast');
hold off

figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
scatter(Z3(1,:),Z3(2,:),'om','Linewidth',1.5);
scatter(Z4(1,:),Z4(2,:),'xg','Linewidth',1.5);
title('Theta Wave Validation Set CSP Separation');
ylabel('Z1 & Z3')
xlabel('Z2 & Z4')
legend('Theta No Training Set', 'Theta Yes Training Set', 'Theta No Validation Set', 'Theta Yes
Validation Set', 'Location','northeast')


%figure
% hold on
%histogram(M1(10,:));
%histogram(M2(10,:));
%hold off
%% Alpha Training Set (4/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M1 = alphano(:,1:62);
M2 = alphayes(:,1:58);

%Initialize:
R1 = M1* M1';
R2 = M2 * M2';
%Calculate:

R1 = R1/trace(R1);
R2 = R2/trace(R2);
Rsum=R1+R2;

[EVecsum,EValsum]=eig(Rsum);
[EValsum,ind] = sort(diag(EValsum), 'descend');
EVecsum=EVecsum(:,ind);

W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix
```

```
S1=W*R1*W';
S2=W*R2*W';

[B,D]=eig(S1,S2);
[D,ind]=sort(diag(D));
B=B(: ,[ind(1),ind(14)]); %To look at 1 and 2 change ind to ind1:2
MCSP=B'*W;

Z1 = MCSP*M1;
Z2 = MCSP*M2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Alpha Evaluation Set (1/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M3 = alphano(:,63:size(alphano,2));
M4 = alphayes(:,59:size(alphayes,2));
Z3 = MCSP*M3;
Z4 = MCSP*M4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Alpha Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
title('Alpha Wave Testing Set CSP Separation');
ylabel('Z1');
xlabel('Z2');
legend('Alpha No Training Set', 'Alpha Yes Training Set','Location','northeast');
hold off

figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
scatter(Z3(1,:),Z3(2,:),'om','Linewidth',1.5);
scatter(Z4(1,:),Z4(2,:),'xg','Linewidth',1.5);
title('Alpha Wave Validation Set CSP Separation');
ylabel('Z1 & Z3')
xlabel('Z2 & Z4')
legend('Alpha No Training Set', 'Alpha Yes Training Set', 'Alpha No Validation Set', 'Alpha Yes
Validation Set', 'Location','northeast')
%% N400 ERP Training Set (4/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M1 = N400No(:,1:62);
M2 = N400Yes(:,1:58);

%Initialize:
R1 = M1* M1';
R2 = M2 * M2';
%Calculate:

R1 = R1/trace(R1);
R2 = R2/trace(R2);
Rsum=R1+R2;

[EVecsum,EValsum]=eig(Rsum);
[EValsum,ind] = sort(diag(EValsum), 'descend');
EVecsum=EVecsum(:,ind);

W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix

S1=W*R1*W';
S2=W*R2*W';

[B,D]=eig(S1,S2);
[D,ind]=sort(diag(D));
B=B(: ,[ind(1),ind(14)]); %To look at 1 and 2 change ind to ind1:2
```

```
MCSP=B'*W;

Z1 = MCSP*M1;
Z2 = MCSP*M2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% N400 ERP Evaluation Set (1/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M3 = N400No(:,63:size(N400No,2));
M4 = N400Yes(:,59:size(N400Yes,2));
Z3 = MCSP*M3;
Z4 = MCSP*M4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% N400 ERP Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
title('N400 ERP Testing Set CSP Separation');
ylabel('Z1');
xlabel('Z2');
legend('N400 No Training Set', 'N400 Yes Training Set','Location','northeast');
hold off

figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
scatter(Z3(1,:),Z3(2,:),'om','Linewidth',1.5);
scatter(Z4(1,:),Z4(2,:),'xg','Linewidth',1.5);
title('N400 Wave Validation Set CSP Separation');
ylabel('Z1 & Z3')
xlabel('Z2 & Z4')
legend('N400 No Training Set', 'N400 Yes Training Set', 'N400 No Validation Set', 'N400 Yes
Validation Set', 'Location','northeast')
%% P600 ERP Training Set (4/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

M1 = P600No(:,1:62);
M2 = P600Yes(:,1:58);

%Initialize:
R1 = M1* M1';
R2 = M2 * M2';
%Calculate:

R1 = R1/trace(R1);
R2 = R2/trace(R2);
Rsum=R1+R2;

[EVecsum,EValsum]=eig(Rsum);
[EValsum,ind] = sort(diag(EValsum), 'descend');
EVecsum=EVecsum(:,ind);

W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix

S1=W*R1*W';
S2=W*R2*W';

[B,D]=eig(S1,S2);
[D,ind]=sort(diag(D));
B=B(: ,[ind(1),ind(14)]); %To look at 1 and 2 change ind to ind1:2
MCSP=B'*W;

Z1 = MCSP*M1;
Z2 = MCSP*M2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% P600 ERP Evaluation Set (1/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M3 = P600No(:,63:size(P600No,2));
M4 = P600Yes(:,59:size(P600Yes,2));
Z3 = MCSP*M3;
Z4 = MCSP*M4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% P600 ERP Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
title('P600 ERP Testing Set CSP Separation');
ylabel('Z1');
xlabel('Z2');
legend('P600 No Training Set', 'P600 Yes Training Set','Location','northeast');
hold off

figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
scatter(Z3(1,:),Z3(2,:),'om','Linewidth',1.5);
scatter(Z4(1,:),Z4(2,:),'xg','Linewidth',1.5);
title('P600 ERP Validation Set CSP Separation');
ylabel('Z1 & Z3')
xlabel('Z2 & Z4')
legend('P600 No Training Set', 'P600 Yes Training Set', 'P600 No Validation Set', 'P600 Yes
Validation Set', 'Location','northeast')
%% Four Feature Training Set (4/5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A = (size(alphano,2) - 15);
G = (size(alphayes,2) - 15);
Q = [N400No;P600No;thetano;alphano];
F = [N400Yes;P600Yes;thetayes;alphayes];
%%
C = 6;
E = 9;
H = C+1;
I = E+1;
NNN = 8;    %Divisibility number
MMM = 15-NNN;
%%
M1 = Q(:,H:size(Q,2)); %1+number subtracted for indices
M2 = F(:,I:size(F,2));
R1 = M1* M1';
R2 = M2 * M2';
%Calculate:

R1 = R1/trace(R1);
R2 = R2/trace(R2);
Rsum=R1+R2;
[EVecsum,EValsum]=eig(Rsum);
[EValsum,ind] = sort(diag(EValsum), 'descend');
EVecsum=EVecsum(:,ind);

W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix
S1=W*R1*W';
S2=W*R2*W';
[B,D]=eig(S1,S2);
[D,ind]=sort(diag(D));
%%
B=B(: ,[ind(1),ind(56)]);
% B=B(: ,[ind(1:14),ind(43:56)]); %To look at 1 and 2 change ind to ind1:2
% B=B(: ,[ind(1:56)]);
%
MCSP=B'*W;
Z1 = MCSP*M1;
```

```
Z2 = MCSP*M2;


M3 = Q(:,1:C); %Number subtracted for indices
M4 = F(:,1:E);

Z3 = MCSP*M3;
Z4 = MCSP*M4;

Ztrain = [Z1,Z2];
Ztest = [Z3,Z4];
TrainingData = [];
TestingData = [];
TrainingData = [TrainingData;Ztrain];
TestingData = [TestingData;Ztest];



%%

sq = size(Q,2);
sf = size(F,2);
sq = sq - C;
sf = sf - E;



sqp = sq+1;
sfp = sf+1;
sq = sq - NNN;
sf = sf - MMM;

for k = 1:9;

    M1 = [Q(:,1:sq), Q(:,sqp:size(Q,2))];
    M2 = [F(:,1:sf), F(:,sfp:size(F,2))];



    %Initialize:
    R1 = M1* M1';
    R2 = M2 * M2';
    %Calculate:

    R1 = R1/trace(R1);
    R2 = R2/trace(R2);
    Rsum=R1+R2;

    [EVecsum,EValsum]=eig(Rsum);
    [EValsum,ind] = sort(diag(EValsum), 'descend');
    EVecsum=EVecsum(:,ind);

    W=sqrt(pinv(diag(EValsum))) * EVecsum; %Transformation Matrix

    S1=W*R1*W';
    S2=W*R2*W';

    [B,D]=eig(S1,S2);
    [D,ind]=sort(diag(D));
    B=B(: ,[ind(1),ind(56)]);
%     B=B(: ,[ind(1:14),ind(43:56)]); %To look at 1 and 2 change ind to ind1:2
%     B=B(: ,[ind(1:56)]);
    MCSP=B'*W;

    Z1 = MCSP*M1;
    Z2 = MCSP*M2;


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %% Four Feature Evaluation Set (1/5)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    TT = sq +1;
    YY = sf +1;
    XX = sqp - 1;
    ZZ = sfp -1;
    M3 = Q(:,TT:XX);
    M4 = F(:,YY:ZZ);
    % M3 = Q(:,(A+1):size(Q,2));
    % M4 = F(:,(G+1):size(F,2));

    Z3 = MCSP*M3;
    Z4 = MCSP*M4;

    Ztrain = [Z1,Z2];
    Ztest = [Z3,Z4];

    TrainingData = [TrainingData;Ztrain];
    TestingData = [TestingData;Ztest];
    sqp = sq+1;
    sfp = sf+1;
    sq = sq - NNN;
    sf = sf - MMM;
end
Z3size = size(Z3,2);
Z4size = size(Z4,2);
%%
save('Subject 14 2 Feature Train & Test
Data','TrainingData','TestingData','Q','F','Z3size','Z4size');
%%%%%%%%%%%%%%%%%%%%%%8%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Four Feature Data Plotting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
hold on
grid on
scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
title('Subject 1 Four Feature Testing Set CSP Separation');
ylabel('Z1');
xlabel('Z2');
legend('No Training Set', 'Yes Training Set','Location','northeast');
hold off

figure
hold on
grid on
%scatter(Z1(1,:),Z1(2,:),'ob','Linewidth',1.5);
%scatter(Z2(1,:),Z2(2,:),'xr','Linewidth',1.5);
scatter(Z3(1,:),Z3(2,:),'om','Linewidth',1.5);
scatter(Z4(1,:),Z4(2,:),'xg','Linewidth',1.5);
plot([-1000,1000],[1000,-1000],'k--');
plot([-1000,1000],[-1000,1000],'k--');
title('Subject 1 Four Feature Validation Set CSP Separation');
ylabel('Z1 & Z3')
xlabel('Z2 & Z4')
legend('No Validation Set', 'Yes Validation Set', 'Location','northeast')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Biomedical Engineering Thesis Classification
% Written by: Michael Doyel
% May 26th, 2021
% Thesis Advisor: Dr. Chiu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all

% Load Dataset for Classification
load 'Subject 1 56 Feature Train & Test Data'


%%%%%%%%%%%%%%%%%%%%%%%
% Training Data
% Q Length is connected to no
% F length is connected to yes
% The training and test data are X by response # (size q size f) matrices
% where depending on the testing values there are either 20, 280, or 560
% columns.
% Increment both the M + at the end of the loop and the M* multiplier
% depending on the number of features being processed.
%
% The fit function can be changed based on the classifiaction model wanted
% for evalution.
%
% TestingData is the matrix that holds Each two channel 10 point validation
% every two rows ie rows 3 and 4 are the 2nd validation block and so on.
%
% TrainingData is the matrix for the data the model should be trained on.
% Q is the size of the No responses
% F is the size of the Yes responses
%
%
%%%%%%%%%%%%%%%%%%%%%%%
%% Model Training
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m = 1;
CorrectVec = zeros(10,1);
CT1 = 0;
for k = 1:10
    Ztrain = TrainingData(m:56*k,:);
    Ztest = TestingData(m:56*k,:);

    Ztrain = permute(Ztrain, [2 1]);
    Ztest = permute(Ztest, [2 1]);

    C = cell([135 1]);

    for j= 1:size(Ztrain);
        if j <= size(Q,2);
            C(j,1) = {'No'};
        else
            C(j,1) = {'Yes'};
        end
    end
    Mdl = fitcdiscr(Ztrain,C);
    label = predict(Mdl,Ztest);
    for y = 1:15;
        if y <= Z3size;
            if ~any(strcmp(label(y,:),'Yes'))
                CT1 = CT1+1;
            else
                CT1 = CT1 + 0;
            end
        else
            if ~any(strcmp(label(y,:),'No'))
                CT1 = CT1 + 1;
            else
                CT1 = CT1 + 0;
            end
        end
    end
    CorrectVec(k,:) = CT1
    m=m+56
    CT1 = 0
end
MEANCORRECT = mean(CorrectVec);
BEST = max(CorrectVec);
```