Brain Wave Data Analysis Through Time Series Simulations

Marlene Gonzalez
Department of Statistics, California State University of Fullerton, Fullerton, California.

Zachary Peskin
Department of Statistics, California State University of Fullerton, Fullerton, California.

Michael Piccolo
Department of Statistics, California State University of Fullerton, Fullerton, California.

Lucille Peterson
Department of Statistics, California State University of Fullerton, Fullerton, California.

California State University of Fullerton Department of Computer Engineering

Javier De La Cruz
Department of Computer Engineering, California State University of Fullerton, Fullerton, California.

Mohammad Khalil
Department of Computer Engineering, California State University of Fullerton, Fullerton, California.

Abel Desoto
Department of Computer Engineering, California State University of Fullerton, Fullerton, California.

Dr. Kiran George
Department of Computer Engineering, California State University of Fullerton, Fullerton, California.

Abstract: Studies involving brain computer interface (BCI) have continually evolved over recent decades, incorporating various techniques of statistical analysis and modeling. Such analysis has sought to accomplish classification of groups by means of BCI measures of electroencephalogram (EEG) signals, functional magnetic resonance imaging (fMIR), and functional near infra-red spectroscopy (fNIRS) signals as potential predictors of a particular class. EEG signals are commonly transformed from time to a frequency domain in order to extract and use the power spectral density (PSD) bands associated with different physical and mental activities. Studies have attempted to find various means of combining BCI and PSD data for statistical analysis where aggregation is applied prior to modeling. Utilizing data supplied by the Department of Computer Engineering of California State University of Fullerton (CSUF) and focussing primarily on EEG data, the research outlined in this paper aims to preserve BCI readings as a time series signal and utilize the autoregressive integrated moving average (ARIMA) model for statistical inference. Moreover, optimal ARIMA models are used for parametric simulation of data in order to increase sample size and improve the accuracy of modeling for classification. An algorithm is implemented where simulations of EEG signals from a single representative channel collected during experimental segments are used to classify between two different subjects. One-hundred EEG simulations are created for each subject and subsequently transformed into the PSD bands of alpha, beta, and theta prior to modeling. Models used for classification include k-nearest neighbors, logistic regression, random forest, and a single layer neural network and the performance of each model is based on accuracy, sensitivity, and specificity. As a result, the metrics of sensitivity and specificity are impacted more by two of the models whereas accuracy shows little variation between models, which means that the optimal model could depend on the needs of the research. However, in summary the model with the highest sensitivity, which in this case has the most accurate prediction rate of subject 1, is logistic regression and the model with the highest specificity, or the most accurate prediction rate of subject 2, is the single layer neural network.

Introduction

The Computer Engineering Department at California State University, Fullerton, has conducted extensive research on brain activity during mental stimuli. The experiments are supervised by Dr. Kiran George and are led by graduate students: Javier De la Cruz, Mohammad Khalil, and Abel Desoto.

How the brain reacts during cognitive tasks has been extensively researched through both invasive and non-invasive practices throughout history. The most common non-invasive form of data collection is through the brain-computer interface (BCI), which communicates brain activity to a computer by use of a headset [17]. The g.Nautilus Research fNIRS-8 headset used at CSUF is capable of collecting both electroencephalography (EEG) and functional near-infrared spectroscopy (fNIRS) signals simultaneously.

EEG signals measure the electrical changes on the scalp caused by neurons firing in the brain that can be transformed into spectral waves that are described in terms of frequency bands which are commonly separated into gamma (>30 Hz), beta (13-30 Hz), alpha (8-13 Hz), theta (4-8 Hz) and delta (0.5-4 Hz). Gamma waves are associated with concentration and problem-solving, beta waves are generated during most normal activities while awake and described as awareness, alpha waves are associated with relaxation or rest from a task, theta waves are generated during daydreaming or deep relaxation, and delta waves are generated during sleep [14]. Functional near-infrared spectroscopy (fNIRS) measures changes in the concentration of oxygenated hemoglobin (HbO) and deoxygenated hemoglobin (HbR) in the outer cerebral cortex of the brain and is complementary to EEG data. HbO concentration is expected to increase after neurons fire in the brain as a result of an increase in blood flow, and HbR is expected to decrease [15]. There are limitations for each BCI signal- EEG lacks the ability to tell us where in the brain the activity is coming from, and fNIRS lags in the time it takes for changes to appear (approximately 4-6 seconds). Where fNIRS data lacks in temporal resolution EEG makes up for it, and where EEG lacks in spatial resolution fNIRS makes up for it [16].

The goal is to analyze past data using machine learning methods to ultimately improve prediction accuracy. This goal will be accomplished by first replicating the results and then shifting the focus to improving results through different possibilities of both data aggregation and machine learning models.

The collection of EEG and fNIRS data is equivalent across all of the BCI experiments led by De la Cruz, Khalil, and Desoto, and only differ in cognitive tasks of interest.

Literature Review

The focus of this paper highlights data provided by De La Cruz [1], which is based on research that aims to classify stress through an experiment where subjects play *Reaction Training*, a brain-teaser game that is designed to test a player's reaction speed while also maintaining the accuracy and rules of the game onscreen. During the experiment, EEG and fNIRS signals were collected from the frontal lobe as well as heart rate variability (HRV) collected from the subject's index finger. The equipment used to collect and process data included the g.Nautilus Research fNIRS-8 headset used for gathering EEG and fNIR waves simultaneously and the g.tec MATLAB-Simulink platform for recording and processing the data. The study used eight of the fNIRS channels to collect the concentration of HbO and HbR and twelve EEG

channels. Additional equipment included the Wellue Smart Handheld Pulse Oximeter for collecting HRV data and MATLAB for estimating the power density spectrums of alpha, beta, and theta waves [1]. These estimates are characteristics of the frequency component from the time series EEG signal. The spectral estimation shows how signal power is distributed in terms of frequency, measured in cycles per second or hertz (Hz), and is important for this type of research as mental stimuli can result in either increases or decreases in the various spectral bands. The alpha band is expected to decrease as a result of increased levels of stress whereas beta and theta waves are expected to decrease.

Seven subjects were involved in the experiment with four segments per subject [1]. The first segment served as a control where each subject remains stationary and does not play the game. This is followed by segments two through four where three different games from the Reaction Time application are played in the following order: *Color Change*, *F1 Start Lights*, and *Numbers Order*. All four segments last for approximately 90 seconds, following a 15-second calibration period, during which data is recorded and subsequently used for analysis. The analysis is a comparison of the differences between each game segment and the control. As previously mentioned, alpha readings are anticipated to decrease as a result of increased stress whereas the beta and theta readings are expected to increase. Observed HbO and HbR readings are expected to decrease and increase, respectively, and HRV is expected to increase. Raw data is collected and processed through MATLAB for power spectral estimation of EEG signals and all readings are separately aggregated as an average over time.

The percent change between each game segment and the control of every aggregated quantity is found as a comparison measure. Out of the twenty-one total game segments, seventeen resulted in a decrease in band power for alpha waves, nineteen resulted in an increase of band power for beta waves, and fourteen resulted in an increase of band power for theta waves. The observed fNIRS showed little change in the HbO and HbR readings for each segment and five of the seven subjects showed an increase in average heart rate. The aggregated data was also used to classify “stressed” or “unstressed” using a KNN model through the MATLAB Classification Learner tool with a reported accuracy of 75%.

Similar research into stress detection from EEG and fNIRS data has been conducted by De La Cruz with slight variations. One experiment involved seven subjects that played the “quick decision-making” game *Not-Not* during the three-game segments [2]. Each of these segments included an increase in difficulty of “Normal,” “Hard,” and “Extreme” and the aim was to classify different levels of stress. In this experiment, the fNIRS data showed more significant changes as seventeen segments resulted in a decrease in HbO levels and eighteen showed increased HbR levels. KNN is also used for classification with a reported accuracy of 72%. A third study involved an experiment where six subjects played *Wordscapes* over the three-game segments [3]. In addition to EEG, fNIRS, and HRV data, galvanic skin response (GSR) data, which measures the “electrical conductivity of the subject’s skin,” is collected by attaching two sensors to the subject’s “nondominant hand.” It is reported here that each subject shows increases in skin conductivity in at least two of the game segments. Similar to [1], the classification of “stressed” and “unstressed” is tested using KNN in MATLAB as well as a support vector machine (SVM), with reported accuracies of 77.3% and 72.7%, respectively.

Past studies in this area include the work by Al-Shargie [4], which aimed to show an increase in prediction capabilities of stress with simultaneous measurements of EEG and fNIRS signals. The

research involved the detection of mental stress through an experiment where twelve subjects solve arithmetic problems under conditions of control and stress, where stress is induced through "time pressure and negative feedback of individual performance." The study shows an increase in the "mean detection rate" of a SVM with the "fusion" of EEG and fNIRS signals. The mean detection rates reported are 91%, 95%, and 98% for models using fNIRS, EEG, and a combination of the two signals. An additional outcome is that the research showed a reduction in HbO level readings from the prefrontal cortex that were associated with mental stress.

Bhutta [5] had the goal of combining fNIRS data with autonomic data collected from a polygraph test to "enhance the classification accuracy in single trial lie detection." The experiment involves 16 subjects who are left alone in a room and instructed to steal either a 5000 or a 10,000 Korean Republic Won (KRW) and are subsequently brought into a room for yes/no questioning. There are 10 question sessions per subject and each session is made up of five random questions of one truth, one lie, one neutral, and two control. Each subject is instructed to deny possession of the KRW note. Only truth and lie data is used in the analysis, which was collected during a 15-second wait period after the response was given, providing 20 data points per subject. Linear discriminant analysis (LDA) was used for classification and the results showed an 86.5% accuracy from combining fNIRS and polygraph features, an increase of over 10% from models when only fNIRS or polygraph features are used.

Other literature includes research into whether frontal asymmetry can contribute to individual emotional responses to either positive or negative stimuli [6]. Baseline EEG data was collected from subjects during two different sessions that were three weeks apart. Those Subjects with more relative left-side baseline frontal activation reported a greater positive response to positive film clips and subjects with more relative right-side baseline frontal activation reported a greater negative response to negative film clips. Desoto [7] researched the level at which hearing a story results in patterned brain activity. This study results in 65% of fNIRS data showing "evidence of neural alignment" between subjects who listen to the same audio file." Further research is to extend this general experiment to the classification of learning from EEG and fNIRS data collected while subjects watch educational videos. Lastly, Khalil [8] [9] researched the potential of using EEG and fNIRS data for lie detection. Unlike in the case of Bhutta [5], the goal here is to use only BCI data as an improvement over polygraph data. However, a comparison is never made. The various research conducted always included several different prediction models through the MATLAB Classification Learner tool including several different KNNs and SVMs in [8] and several different neural networks in [9]. What is consistent in terms of model performance is the true negative rates of the classifications are lower than the true positive rate.

Data Structure

When conducting experiments using the g.Nautilus Research fNIRS-8 headset, the g.tec MATLAB-Simulink platform records and processes the raw data. The MATLAB data set files consist of 30 rows and approximately between sixty to over one hundred thousand columns each, per subject, per segment. In total there are 27 MATLAB files of raw data, four for each of the subjects (one file containing data pertaining to the control segment was lost), where the first segment serves as the control and segments two through four are the treatments. For this particular stress experiment, the raw data set stored as a MATLAB file is transposed and exported as a CSV file. Of the transposed data set, the tens of

thousands of rows are readings of EEG and fNIRS data once every 0.002 seconds. It is also worth noting that much of the initial observations includes a calibration period, where a baseline is established and noise can be better filtered (certain movement types can make data from the headset unreliable), which is removed during analysis. The 27 exported CSV files now contain tens of thousands of rows and thirty columns, where the first column is time measured in intervals of 0.002 seconds. Of the remaining 29 columns, 2 through 17 are fNIRS data - 8 channels of information recorded, each being a pair of oxygenated (HbO) and deoxygenated (HbR) hemoglobin. The remaining columns pertain to EEG - the 18th column is a reference node that did not end up seeing much use in their data processing.

Data Wrangling

As mentioned in the prior section, an initial part of the wrangling process consists of transposing the raw data and exporting the twenty-seven MATLAB files to unique CSV files. Also, there is a calibration period of the g.Nautilus Research fNIRS-8 headset that is not retained for analysis, leaving only the last ninety seconds of observations since each segment is ninety seconds in length. Additionally, outliers potentially caused by motion artifacts are smoothed by applying the *tsclean* function, which applies *Friedman's SuperSmoother* function and is part of the *forecast* R package.

An important aspect of data preprocessing involves the transformation of the EEG signals from a time to the frequency domain. Because the EEG signals are initially collected in terms of time, this transformation must occur in order to observe the power spectral density (PSD) values associated with the spectral bands of alpha, beta, and theta. The time series EEG signals are made up of periodic waves in a wide spectrum of frequencies [10]. Frequency, measured in hertz (Hz), describes the number of cycles of an oscillating, periodic waveform per unit of time, which in this case is cycles per second. PSD shows how signal power is distributed in terms of frequency and can be estimated through a discrete Fourier transform (DFT) that is in terms of *k* discrete frequency bins

$$F(k) = \sum_{n=0}^{N-1} x[n] e^{-i2\pi kn/N}, \tag{1}$$

where *n* is a discrete measure of time. Both time and frequency are discretized since the observations are collected at discrete time intervals and the frequency domain is based on a finite sampling rate. It is also worth noting that with Euler's Formula, the exponential can be written as

$$e^{-i2\pi kn/N} = \cos\left(-\frac{2\pi kn}{N}\right) + i\sin\left(-\frac{2\pi kn}{N}\right) \tag{2}$$

showing that the resulting computation at the *k*th frequency bin is a complex number with a real and imaginary part from which the amplitude associated with the power can be computed. Traditional units of the PSD are squared microvolts $\left(\mu V^2\right)$ [10].

It is also important to note the inverse DFT

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} F[k] e^{i2\pi kn/N}, \tag{3}$$

which transforms the PSD back to the original signal in terms of time. To compute equation (1) efficiently a Fast Fourier Transform (FFT) algorithm is commonly used in computer software. PSD values for alpha, beta, and theta are estimated by De La Cruz. [1] in MATLAB using the *pwelch* function, followed by an aggregation of each of the three bands to a scalar by integrating over the frequency interval using the *bandpower* function. The visualizations and analysis PSD estimates in this paper were accomplished using the *eegkit* [11] package in the R programming language [12].

Exploratory Data Analysis (EDA)

For this research EDA will be limited to a single subject and will include various plots in terms of time and frequency domains for all segments. This includes EEG and fNIRS time series graphs as well as graphs of the different spectral bands in terms of their respective frequencies.

EEG Time Series

In the interest of reporting a reasonable number of plots, the EEG time series plots displayed as figures 1 and 2 are restricted to channels 2, 3, 6, 9, 10, and 12. When analyzing all 13 channels it was observed that certain channels consistently match in shape and relative amplitude across all four segments. This could be due to closeness in proximity of the electrodes, and for this reason channels 1, 4, 5, 7, 8, 11, and 13 are not displayed.

Figure 1 (a) shows the EEG time series signals from the control segment on the left compared to figure 1 (b), the same channels of segment 2 on the right. Note that all twelve line graphs represented by figure 1 are not displayed with a consistent scale of the y-axis. By observing the numerical values of microvolts, it appears that in some cases the amplitude of the readings displayed in figure 1 (b) show some level of increased value compared to the control. This is particularly true for channels 2 and 6 where the minimum and maximum amplitude ranges approximately between $\pm$ 40 and $\pm$ 60 microvolts respectively, which upon inspection appears to be significant when compared to the control. Figures 2 (a) and (b) display similar observations for channels 2 and 6 as well as channel 9, which has an approximate minimum and maximum ranging between $\pm$ 100 microvolts for segments 3 and 4.

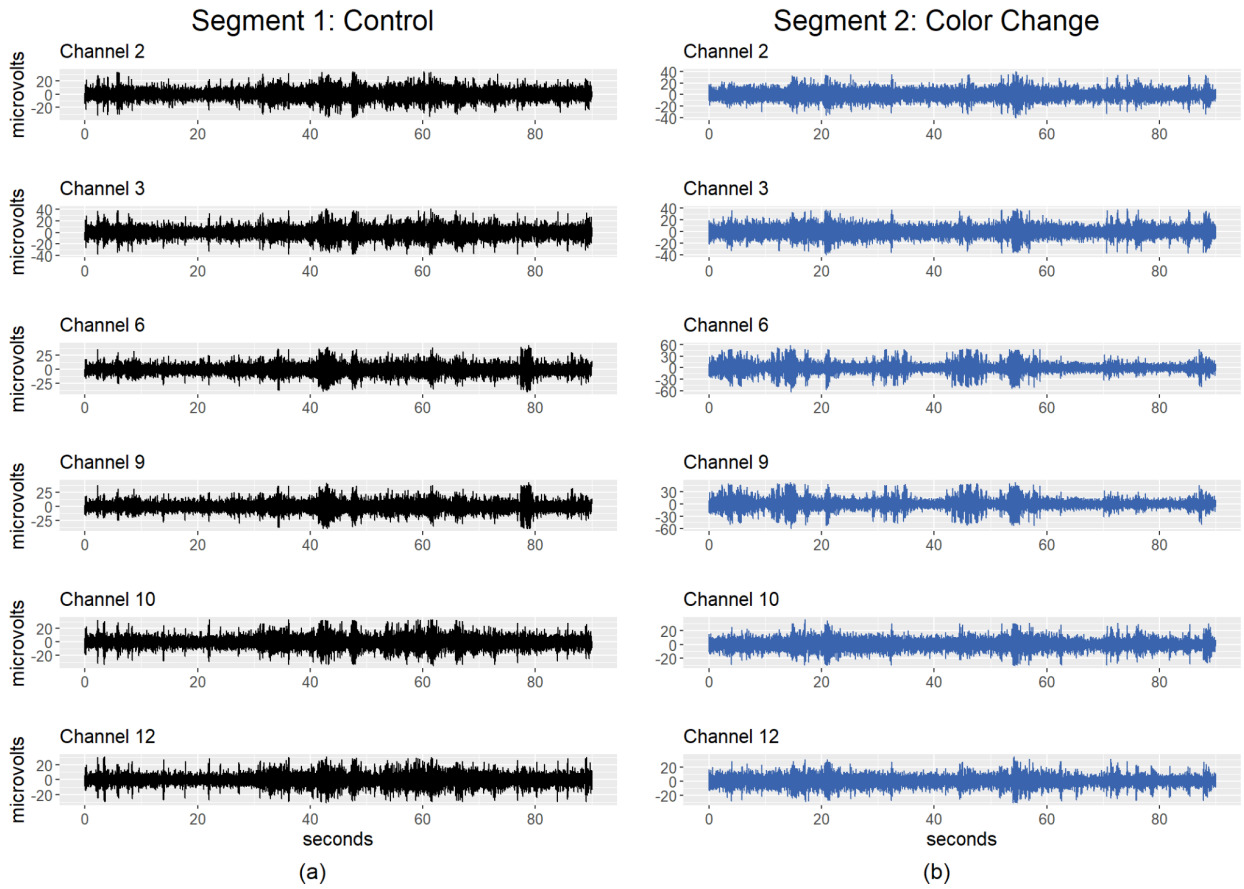

Figure 1. (a) Time series EEG readings from channels 2, 3, 6, 9, 10, and 12 of the control segment. (b) Time series EEG readings from channels 2, 3, 6, 9, 10, and 12 of the second segment where the subject plays the Color Change game for 90 seconds.

Recall that each channel of both EEG and fNIRS data were cleaned by applying the *tsclean* R function, which smooths outliers resulting from motion artifacts. There are more precise methods for identifying and reducing this type of noise such as performing independent component analysis (ICA), where the independent components that attribute to spikes in the data can be removed. This is a practical step that could be implemented in future work in order to ensure the integrity of the data used for analysis.

Since figures 1 and 2 do not capture every EEG channel for each segment, Table 1 can summarize the ranges between the minimum and maximum aptitudes for each reading in order to view potential increases in microvolts for every channel. Across segments 2 through 4 it appears the most consistent and drastic increase in the range of amplitude when compared to the control exists in channels 4, 5, 6, 9, 11, and 13. Every channel in segments 3 and 4 measures some increase in the range of amplitude greater than the control whereas segment 2 shows ranges less than the control in channels 1, 7, 8, and 10.

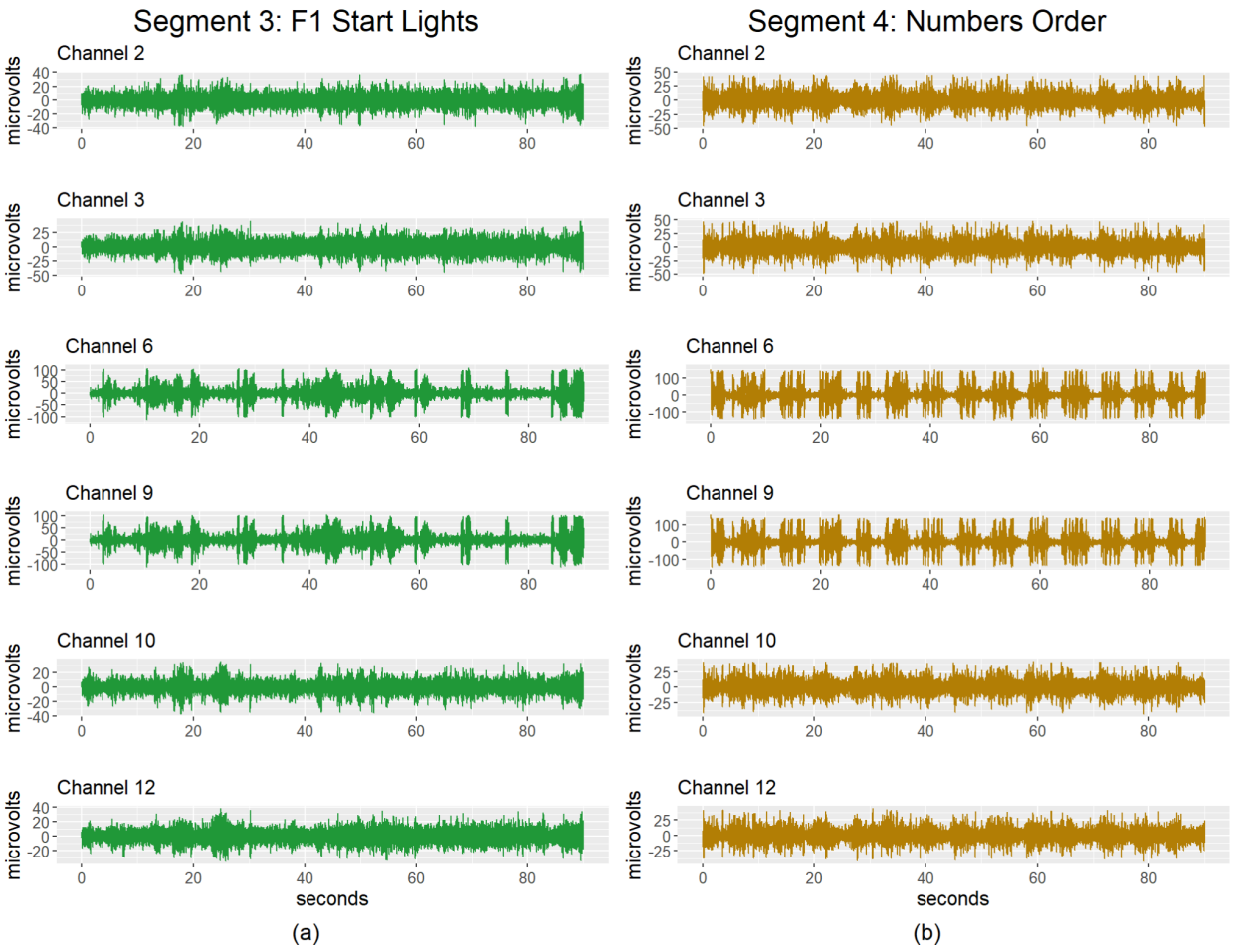

Figure 2. (a) Time series EEG readings from channels 2, 3, 6, 9, 10, and 12 of the second segment where the subject played the F1 Start Lights game for 90 seconds. (b) Time series EEG readings from channels 2, 3, 6, 9, 10, and 12 of the second segment where the subject played the Numbers Order game for 90 seconds.

Table 1: Range in Amplitude Per EEG Channel For Each Segment

CH	Seg	Range	Seg	Range	Seg	Range	Seg	Range
1	1	81.59	2	73.71	3	81.60	4	102.14
2	1	69.69	2	80.46	3	75.46	4	92.37
3	1	80.93	2	78.44	3	92.77	4	96.95
4	1	98.70	2	131.64	3	275.04	4	406.88
5	1	84.25	2	121.88	3	257.67	4	328.55
6	1	82.57	2	120.00	3	227.46	4	310.55
7	1	80.71	2	71.50	3	91.39	4	90.75
8	1	75.42	2	67.00	3	81.51	4	86.29
9	1	81.84	2	112.56	3	220.41	4	306.49
10	1	69.05	2	67.59	3	72.64	4	85.47
11	1	106.96	2	149.18	3	247.17	4	336.42
12	1	61.34	2	65.80	3	73.30	4	86.43
13	1	84.65	2	119.41	3	226.03	4	308.24

Table 1 shows a possible dichotomous separation of EEG channels in terms of an increase of amplitude range of segments 2 through 4 compared to the control. Comparing the same channels, some show a range much greater than the control and some channels show ranges that do not vary much from the control range. A dichotomous grouping also seems to exist with channel correlation as is seen in figure 3. The visualization from the heatmap appears to demonstrate a possible polarization between strong and weak positive correlation of EEG channels in segment 4. Although not displayed, similar observations are viewed from the other two non-control segments, however the binary grouping is less dispersed in the case of segment 2.

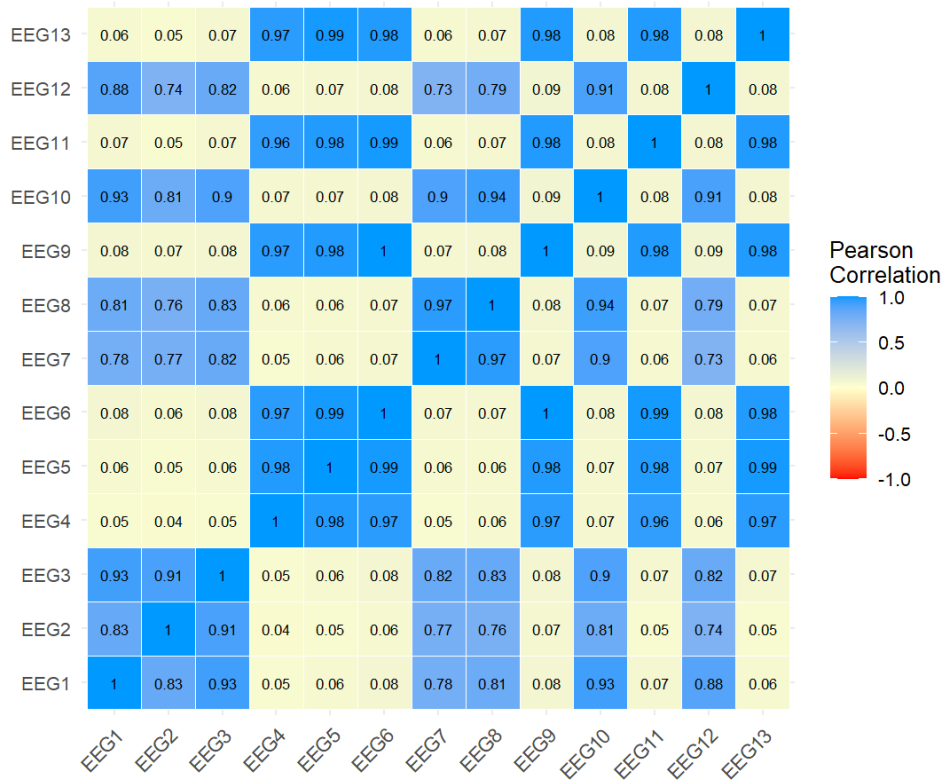

Figure 3. Correlation matrix of EEG channels from experimental segment 4 with heatmap overlay.

Table 2: Non-Control Representative Channels

Segment	EEG CH	EEG Corr.	HbO CH	HbO Corr.
2	EEG 9	0.6451	HbO 5	0.5836
3	EEG 6	0.5556	HbO 2	0.8026
4	EEG 10	0.5270	HbO 5	0.7426

From the preceding EDA it seems reasonable to find a single representative channel from each of the non-control segments that can be used for further analysis and modeling. Table 2 displays the EEG and HbO channels that are on average most correlated with all other channels. Plots for these representative EEG and HbO channels are displayed as figures 4 (a) and (b) respectively.

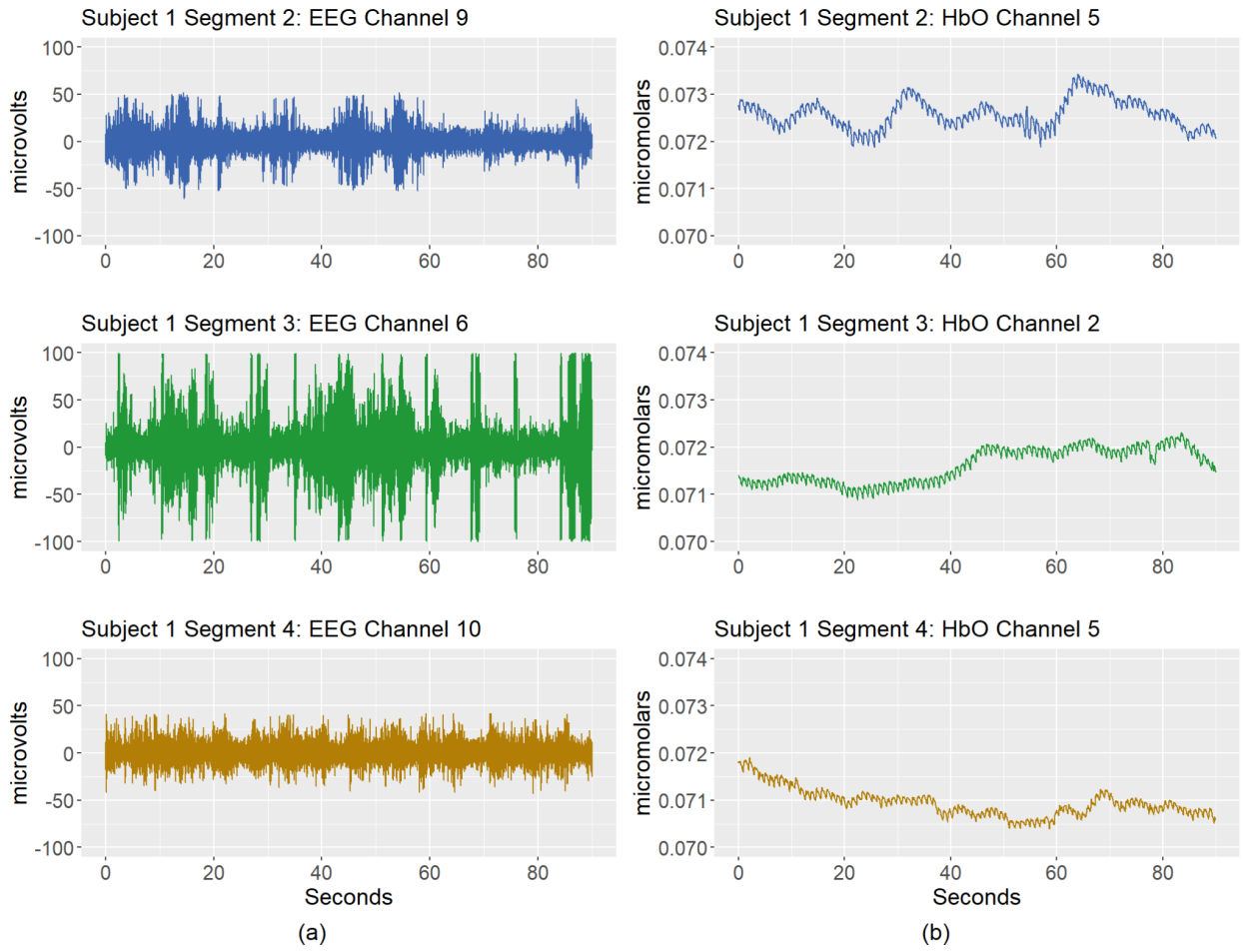

Figure 4. (a) Representative EEG channel from Segments 2 through 4 determined as on average most correlated with all other channels. (b) Representative HbO channel from Segments 2 through 4 based on the same criteria.

The representative EEG channels displayed in figure 4 (a) shows a maximum range in amplitude occurring in experimental segment 3. Channel 6 of segment 3 appears to vary approximately between $\pm$ 100 between the maximum and minimum amplitudes whereas the representative channels of segments 2 and 4 show a much lower range. This observation is also confirmed in table 1. In terms of fNIRS data, the focus of a representative channel has been placed on HbO readings as shown in figure 4 (b). Note that the views of the HbO line graphs are zoomed in to show some variation of signals in each segment. However, considering the values of the vertical axis, significant variation between segments might be questionable. The remainder of this research ultimately focusses on EEG data, but in the interest of potential inclusion of fNIRS data in future work, there will be some limited reporting of results pertaining to HbO data.

Power Spectral Density Bands

Applying the Fourier transformation allows for filtering the data by frequency to observe differences in segments for each channel by alpha, beta, and theta waves separately. The time element is not as present, though the raw EEG time series data can be used side by side: large spikes at a certain time may be attributable to a similar spike found in one or more wave-specific spectral band graphs, such that it may be discerned, for example, that the bulk of activity at a certain time in the segment was beta waves if the spike is predominantly present in the beta wave graph for that segment. As a demonstration, the three different waves of interest will be presented per segment for channel 10. The expectations are, under stress, decreases in alpha waves, increases in beta waves, and increases in theta waves, with more variation.

Figure 5 shows that the three non-control segments saw decreases in alpha wave activity, with the most significant decrease compared to the control being segment 2. Figure 6 shows that beta behavior compared to the control varies a fair amount, with expected increases in segment 4, but possible decreased values across segments 2 and 3. Stress was not exhibited for the first two games via beta waves in this particular channel. Theta wave behavior displayed in figure 7 is comparable to beta wave behavior as anticipated, with the fourth segment having increased activity, but the second and third being comparable or a bit decreased, all of this being noisier than the beta comparisons. Alpha and beta waves are better consistent points of reference because of this.

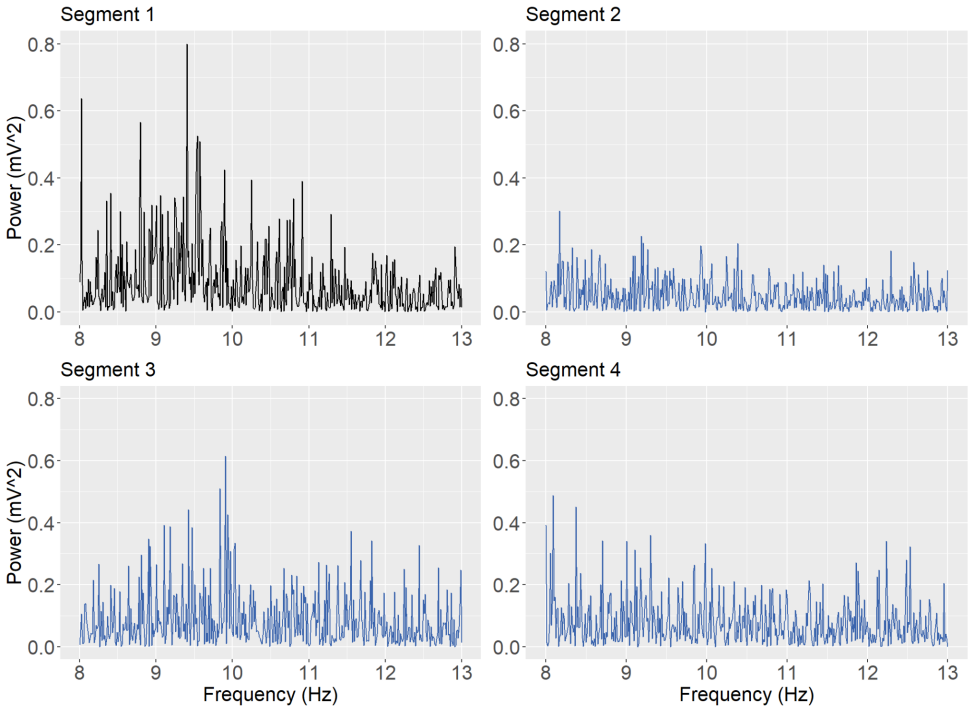

Figure 5: Channel 10 power spectral density estimates of alpha of all four segments.

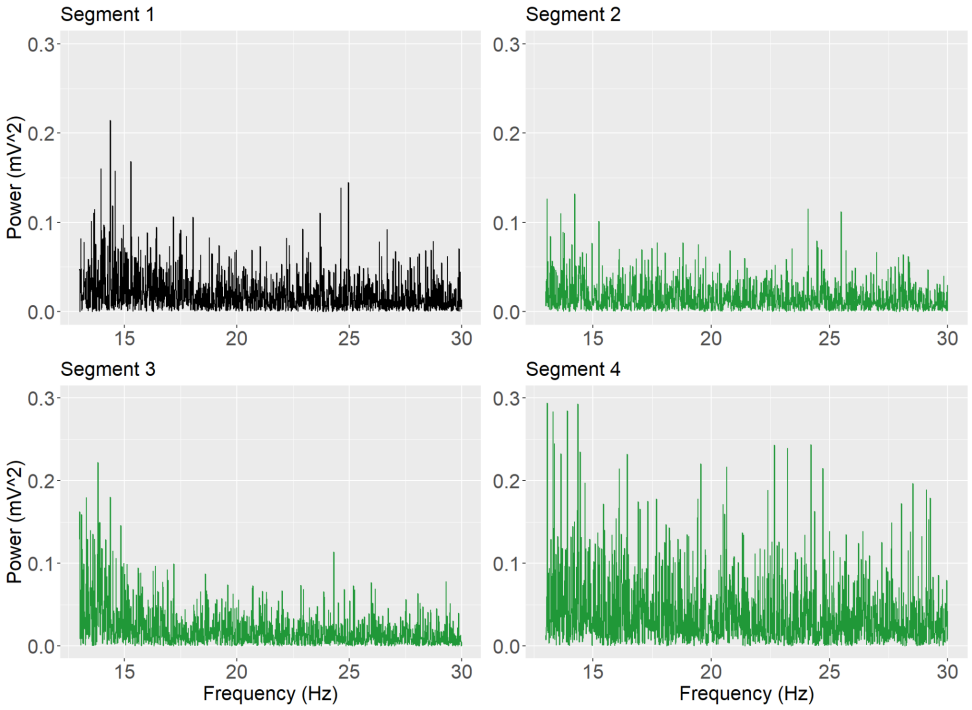

Figure 6: Channel 10 power spectral density estimates of beta of all four segments.

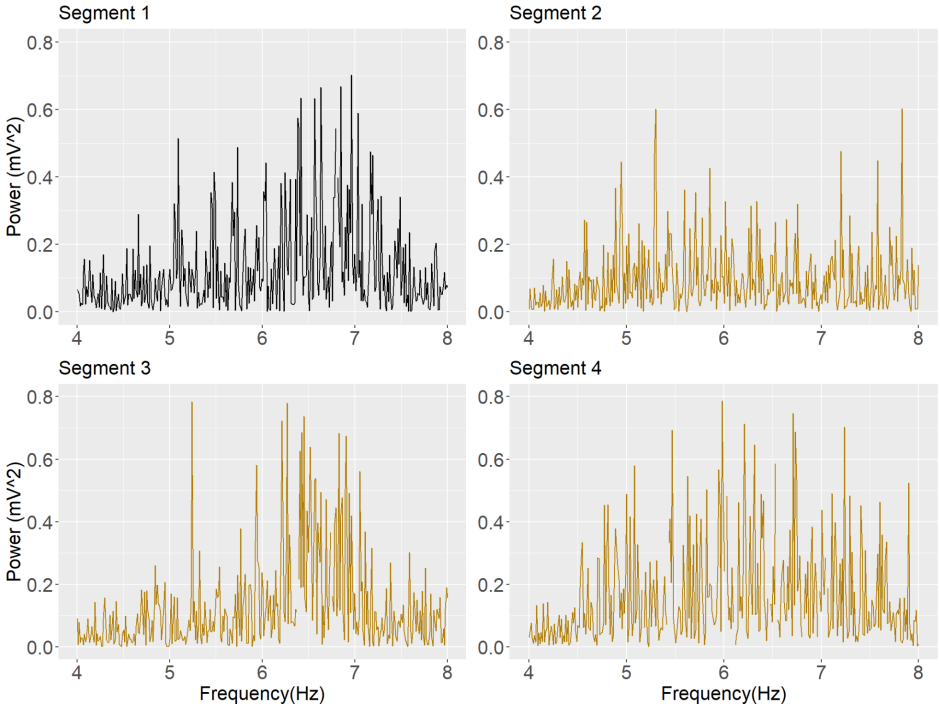

Figure 7: Channel 10 power spectral density estimates of theta of all four segments.

A comparison of two different segments through the differences of power related to the spectral bands can also be viewed, as is shown in figure 8. Here, the PSD estimates within each band for segment 2 are subtracted from those of segment 4. If these segments are exhibiting similar levels of stress then their average difference should be close to zero. To help visualize this idea, a kernel regression estimation was applied to create a smoothing curve from the data. This was done for channel 10 across the three spectral bands.

Visually it appears that average differences of alpha, beta, and theta between the two segments is close to, but slightly above, zero. In the case of theta, the smoothing is more curved than in the other two graphs where the regression line is increasing until about 7 Hz and then begins to decrease. The variation between the three spectral bands can be compared. Theta appears to consistently have a greater amount of variation in these two examples as it's maximum and minimum power ranges between approximately -0.5 and 1, whereas the ranges associated with alpha and beta appear to be similar to one another. The three graphs were not scaled with the same y-axis in order to view how close the smoothing curve is to zero while viewing the range of values in each graph.

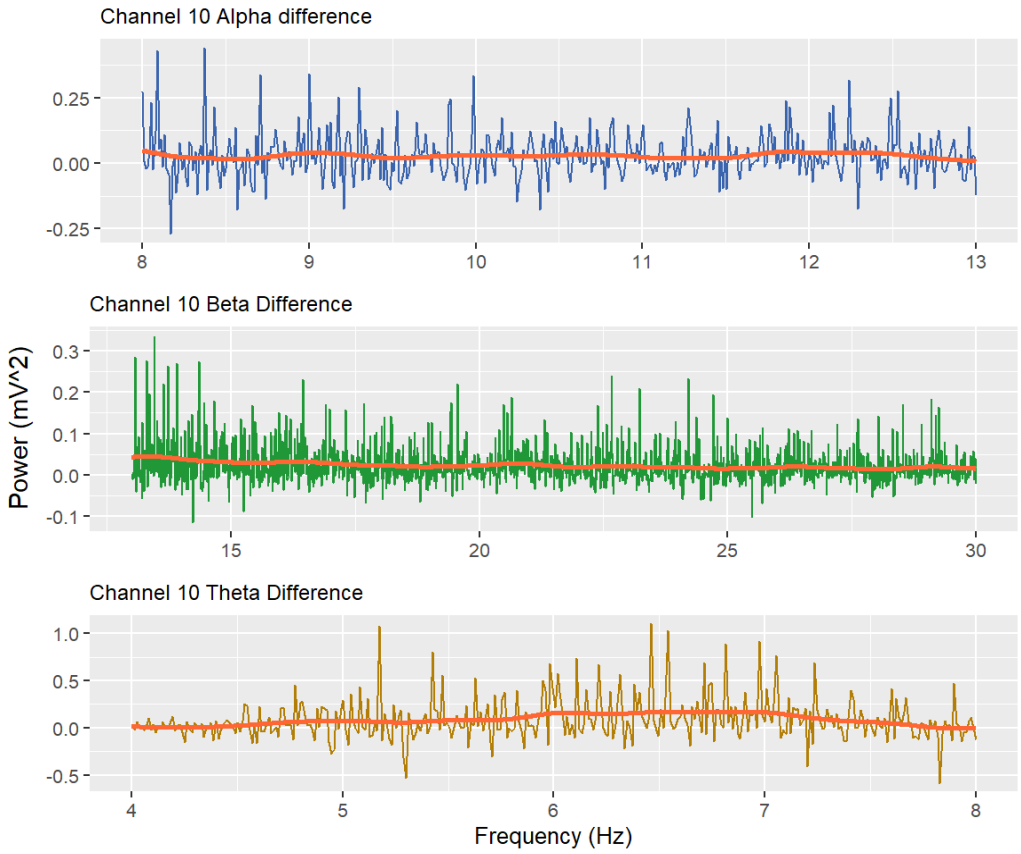

Figure 8: Channel 3 and channel 10 differences between segments 2 and 4 with a non-parametric kernel regression estimate for smoothing within each of the three spectral bands.

Modeling

When considering the raw EEG and fNIRS data as being time series, it makes sense to approach the task of classification initially with a time series solution. In doing so the focus will ultimately be placed on the EEG data in order to find the best way to classify through the power spectral density bands of alpha, beta, and theta. However, some details related to forecasting models fitted to fNIRS data is discussed so that it is available for potential future work.

Background in time series ARIMA modeling

A time series is a sequence of observations taken sequentially in time [18]. Time series forecasting, or predictions of future values, can be obtained through the *autoregressive* or the *moving average model*. From the autoregressive model of order p, or AR(p), a current observed value z_t is the weighted sum of p previous dependent observations at times $t-1,\ t-2,\ \ldots\ , t-p$ plus a random error at time t, which can be expressed by the equation

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \cdots + \phi_p \tilde{z}_{t-p} + \epsilon_t, \tag{4}$$

where $\tilde{z}_t = z_t - \mu$ is the current value's deviation from the mean and $\epsilon_t \sim N\left(0,\ \sigma_\epsilon^2\right)$.

In contrast, the moving average model of order q is based on the weighted sum of the observed errors at times $t-1,\ t-2,\ \ldots\ , t-q$ plus a random error at time t with equation

$$\tilde{z}_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \tag{5}$$

The *autoregressive moving average* model of order (p, q), or ARMA(p, q), is a combination of the two models, which can be seen in equation 6

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \cdots + \phi_p \tilde{z}_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} \tag{6}$$

As in AR(p) and MA(q), the error at time t is assumed to follow a normal distribution. There are $p + q + 2$ parameters that are estimated from the data including $\mu,\ \phi_1,\ \ldots\ , \phi_p, \theta_1,\ \ldots\ ,\ \theta_q,\ \sigma_\epsilon^2$.

In order to account for any potential non-stationarity of the data, the *autoregressive integrated moving average* (ARIMA) model is applied to both the EEG and fNIRS data. The distinction between ARMA (p, q) and ARIMA(p, d, q) is that the latter applies differencing indicated by the value of d. For this data set, only a differencing of $d = 1$ is considered since the visualizations of both the EEG and fNIRS data do not display any relations that would suggest otherwise.

The task of fitting an ARIMA model involves the determination of the values of p and q, which is to determine the number of previously observed values up to time $t - p$ and the number of previously observed errors up to time $t - q$ that are included in the fitted model. One method is using the autocorrelation function (ACF) and partial autocorrelation function (PACF) by viewing their plots to visually determine by which lag the autocorrelation is close to zero. The ACF is used to find the order of the moving average model whereas the PACF is used to find the order for the autoregressive model. The general distinction between ACF and PACF is that in PACF the correlated influence between past observations is removed, leaving only a direct autocorrelation between the current and past value. This visual technique can work in ideal cases, but can also be challenging as seen with this data.

ARIMA model selection of EEG data

Figure 12 (a) displays the plots of the autocorrelation function for the second channel of EEG data taken from the same subject of focus in the previous Data Visualization section. The top plot displays up to 1,000 lags, which is extreme, but shows an oscillating pattern between positive and negative autocorrelation that is present. Viewing up to 50 lags shows that the ACF begins to approach zero around lag 15, but continues to decrease into negative autocorrelation until reaching a local minimum around lag 35. This oscillating pattern, although gradually converging to zero, appears to continue. Comparable plots of thePACF shown in figure 12 (b), although appear to approach zero within much fewer lags, also exhibit some uncertainty in its interpretation as there are instances of spikes of autocorrelation.

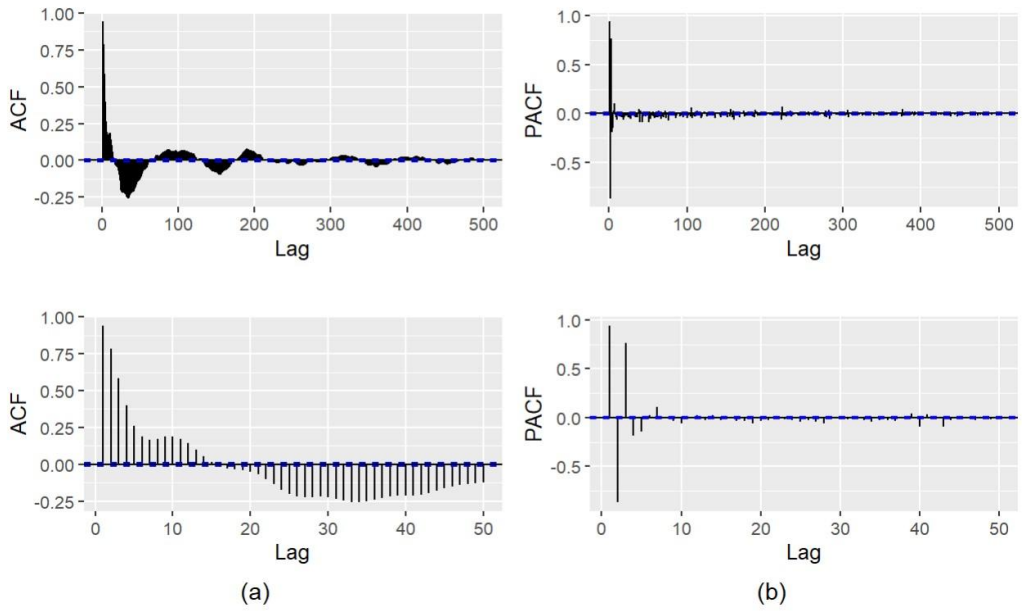

Figure 9: (a) *Autocorrelation function* plots for EEG channel 2 from experimental segment 2 with top plot showing up to 1,000 lags and bottom up to 50 lags. (b) *Partial autocorrelation plots* of the same data up to the same number of lags.

Figure 13 (a) compares the fourth channel between experimental segments two and three. Note not just the similarity between those segments but also their similarities to channel 2 shown in figure 12 (a). The

PACF does show some nuances in figure 13 (b) that might indicate a possible difference in autoregressive order between those experimental segments for EEG channel 4. However, just as in figure 12 (b), there are instances of spikes of autocorrelation beyond the initial drop towards zero. Although not displayed, nearly identical results occur for experimental segment 4 as well as for each channel within all segments. Because of this, ACF and PACF are difficult to interpret when determining the order of either autoregressive or moving average models, giving the need for an alternative approach.

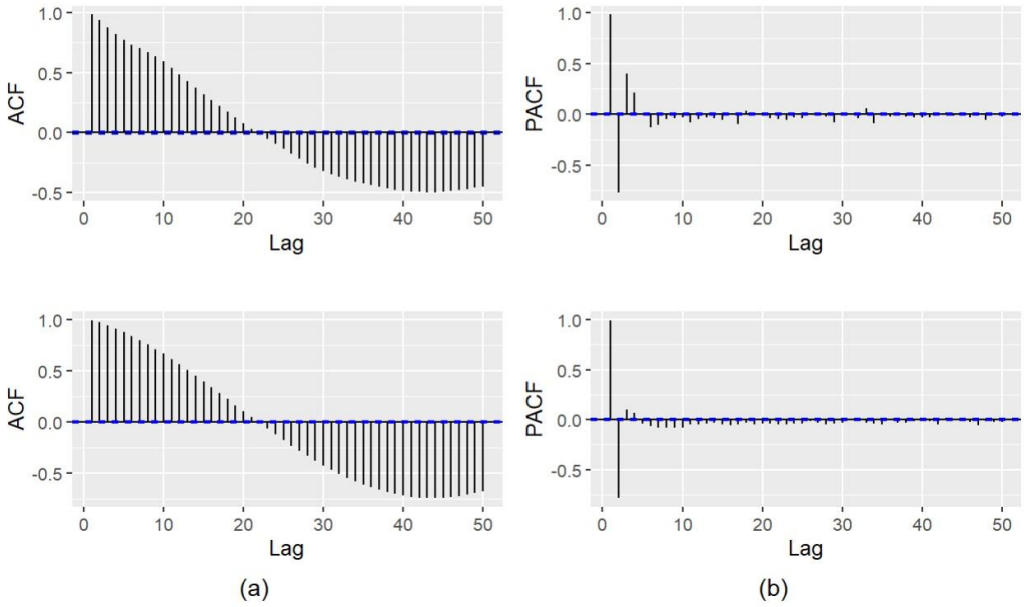

Figure 10: (a) *Autocorrelation function* plots for EEG channel 4 from experimental segment 2 (top) and experimental segment 3 (bottom) up to 50 lags. (b) *Partial autocorrelation function plots* of the same data up to the same number of lags.

Autoregressive Integrated Moving Average model selection of EEG data

An empirical approach to estimating the orders of p, d, and q of an ARIMA model is implemented since we cannot determine the order from the ACF and PACF plots. This approach also allows for control over the complexity of the model. The process begins by setting combinations of p =1, 2, 3 and q = 0 , 1 , 2 , 3 and calculating the Akaike's Information Criterion (AIC). The order of p and q that result in the lowest AIC is then chosen as the best model for each EEG channel. The process is replicated for both a difference of zero, d=0, and a differencing of one, d=1. The results for segment 2 with the lowest AIC value with and without differencing can be seen in Table 3. Segment 1 and segment 3 have been left out for brevity due to similar results of orders of mostly 3 for p and q. Table 4 is a summary of all best-fit models for each experimental segment of all channels. There are 3 unique best-fit models (3,0,3), (3,0,2), and (2,0,3) and most channels fall under ARIMA(3,0,3). This shows that the EEG data is stationary since all best-fit models set d=0 but it also shows that there may be some issues of overfitting since the order of the autoregressive and moving average terms are mostly of order 3. We can assume that if we extended the exhausted process further it would result in orders of the highest order.

Table 3: Segment 2 ARIMA model comparisons by EEG channel

CH	p	d	q	AIC	p	d	q	AIC
1	3	0	3	121582	3	1	3	123623
2	3	0	3	116129	3	1	3	119132
3	3	0	3	122007	3	1	3	124831
4	3	0	3	183776	3	1	3	184599
5	3	0	3	171278	3	1	3	171910
6	3	0	3	163134	3	1	3	163931
7	3	0	3	72545	2	1	3	75505
8	3	0	3	78557	3	1	3	82048
9	3	0	3	157920	2	1	3	158838
10	3	0	2	114250	3	1	3	116383
11	3	0	3	172174	3	1	2	173200
12	3	0	3	111295	3	1	3	114081
13	3	0	3	165679	3	1	3	166376

Table 4: ARIMA modles with lowest AIC

segment	channels	model
2	1-9, 11-13	ARIMA(3,0,3)
2	10	ARIMA(3,0,2)
3	1-3, 6-13	ARIMA(3,0,3)
3	4-5	ARIMA(2,0,3)
4	1-3, 6-13	ARIMA(3,0,3)
4	4-5	ARIMA(2,0,3)

Autocorrelation of Oxygenated Hemoglobin (HbO) values

Due to the inverse relationship between HbO and HbR in the fNIRS data, the ACF and PACF plots for HbO are provided due to the signals being slightly more reactive. The ACF plot for segment 2 channel 2 in figure 14 (a), shows very high correlation between one time to the next for the HbO time series. The PACF plot in figure 14 (b) displays significant lags throughout the plot with no visible tail off toward zero. For brevity all other segments and channels have been left out due to similar results of both high autocorrelation in the ACF plots and spikes in lags throughout the PACF plots. As in the EEG data, there is a need for an empirical approach to determine the order of the ARIMA model.

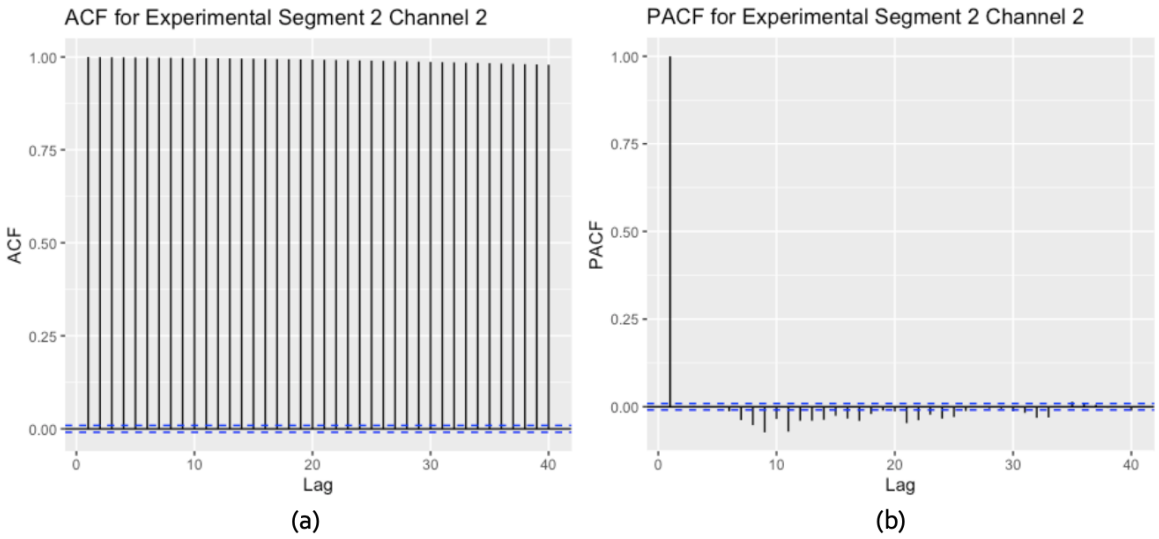

Figure 11. (a) *Autocorrelation function* plots for HbO channel 2 from experimental segment 2 with top plot showing up to 40 lags. (b) *Partial autocorrelation function plots* of the same data up to 40 lags.

Autoregressive Integrated Moving Average model selection of HbO data

The same empirical approach used previously for the EEG data is also used for each of the 8 HbO channels in segment 2. A summary of the results can be seen in Table 5. In channel 4 with a difference of zero, d=0, there is a clear indication of nonstationarity since the model is unable to provide a best model and requires a difference of one, d=1. Here we can note that in table 5 the best model chosen is mostly the simplest model, ARIMA(1,0,0). Table 6 for segment 3, has no issues meeting the stationarity assumption and results in even more best models of ARIMA(1,0,0). Table 7 for segment 4 displays 3 channels that do not meet the stationarity assumptions which again require a difference of one. A summary of the results for all of the best-fit models of all experimental segments and channels can be found in Table 8 where there is a good balance between model complexity and goodness-of-fit to the data.

Table 5: Segment 2 ARIMA model comparison by HbO data

CH	p	d	q	AIC	p	d	q	AIC
1	2	0	2	-927251.0	1	1	0	-927225.8
2	3	0	2	-951347.0	1	1	0	-951289.9
3	2	0	2	-903080.9	1	1	0	-903051.9
4	NA	0	NA	NA	1	1	0	-965475.2
5	2	0	2	-949139.6	1	1	0	-949123.0
6	2	0	2	-930209.9	1	1	0	-930194.7
7	3	0	3	-956730.2	1	1	0	-956675.8
8	3	0	3	-936145.0	1	1	0	-935839.8

Table 6: Segment 3 ARIMA model comparison by HbO data

CH	p	d	q	AIC	p	d	q	AIC
1	1	0	0	-947940.2	1	1	0	-947928.2
2	3	0	2	-960892.0	1	1	0	-960871.1
3	3	0	2	-924189.1	1	1	0	-924151.8
4	1	0	0	-968751.0	1	1	0	-968735.5
5	1	0	0	-964156.3	1	1	0	-964141.2
6	1	0	0	-937454.8	1	1	0	-937440.3
7	1	0	0	-978651.8	1	1	0	-978634.7
8	2	0	3	-976246.3	1	1	0	-976203.9

Table 7: Segment 4 ARIMA model comparison by HbO data

CH	p	d	q	AIC	p	d	q	AIC
1	NA	0	NA	NA	3	1	2	-959034.0
2	3	0	3	-966247.0	3	1	3	-966511.2
3	3	0	3	-924852.6	3	1	2	-926661.4
4	NA	0	NA	NA	1	1	1	-972210.1
5	2	0	2	-968705.5	3	1	3	-968979.2
6	2	0	3	-945264.6	3	1	2	-946682.5
7	NA	0	NA	NA	3	1	3	-978752.9
8	2	0	3	-986502.0	3	1	2	-988796.2

Table 8: ARIMA models with lowest AIC

segment	channels	model
2	1,3,5,6	ARIMA(2,0,2)
2	7,8	ARIMA(3,0,3)
2	2	ARIMA(3,0,2)
2	4	ARIMA(1,1,0)
3	1,4,5,6,7	ARIMA(1,0,0)
3	2,3	ARIMA(3,0,2)
3	8	ARIMA(2,0,3)
4	1,3,6,8	ARIMA(3,1,2)
4	2,5,7	ARIMA(3,1,3)
4	4	ARIMA(1,1,1)

Time Series Simulation and Modeling Algorithm

For the remainder of this study the focus will be isolated to EEG data with a future goal being to adapt what is discussed to eventually include fNIRS data as well. From the optimal ARIMA models displayed in table 4, the model associated with one of the representative channels discussed earlier is chosen. As a reminder, the representative channel is on average most correlated with all other channels within a segment. For segments 2, 3, and 4 the representative EEG channels were 9, 6, and 10 respectively. Using EEG data from channel 10 of segment 4, the algorithm outlined in figure 12 is implemented as a method that will ultimately result in classification.

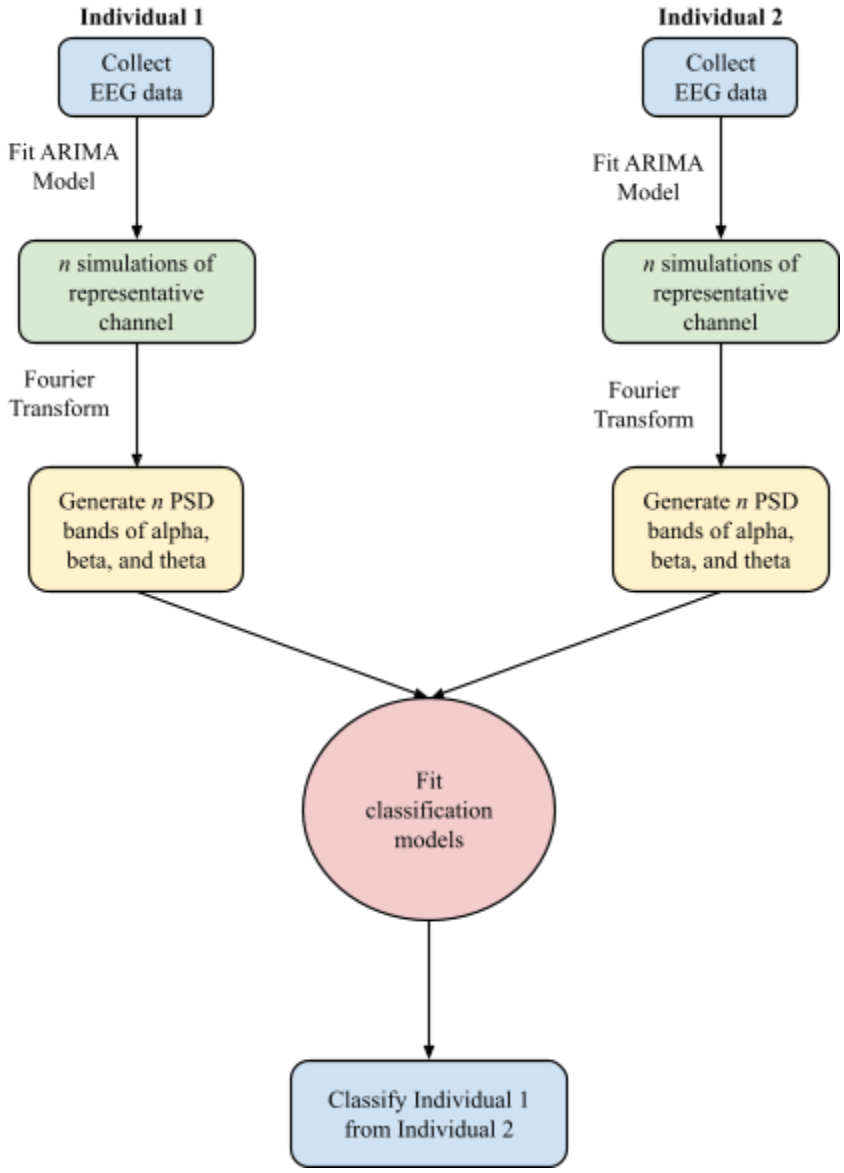

Figure 12. Flowchart outlining the algorithm used for classification of EEG time series data in the frequency domain.

Recall that the optimal model for segment 4 channel 10 is ARIMA(3, 0, 3). The purpose of the selected ARIMA model is for parametric simulation of data that can be used for classification. From the data of the representative EEG channel, $n = 100$ time series simulations are created using the fitted model to increase the sample size, as is outlined in figure 12. Simulation is used as a tool to overcome limitations created by a small number of test subjects, which commonly occurs given limited resources for data collection. Once simulations are generated, the Fouier transform is applied to each to obtain the unique PSD bands of alpha, beta, and theta. This process is repeated for a second individual using the data collected from the same channel during the same segment. After fitting the optimal ARIMA model from this data, simulation is used to the same effect. For this subject the optimal model is ARIMA(3, 0, 3).

The generated PSD data is stored in a data matrix along with assigned categorical labels necessary for fitting a classification model. The aim here will be to classify between individual 1 and individual 2, but this algorithm can easily be adapted for other classification purposes. Below is a short summary of the data matrix created using the algorithm outlined in figure 12.

- Total of 457,000 rows and 4 columns.
- **First column**:
    - Categorical variable labeling individuals 1 and 2.
- **Second column**:
    - Categorical variable to label alpha, beta, and theta as 1, 2, and 3 respectively.
- **Third column:**
    - Numeric variable of the frequency value of each observation.
- **Fourth column**:
    - Numeric variable of the strength (microvolts) of each observation.

Note that an increase of the number of simulations could improve the effectiveness of this process. The number of simulations set as $n = 100$ is chosen due to limited computational power given the resources of this study.

Classification Performance Metrics

The four different models considered for classification are k-nearest neighbors (KNN), logistic regression, random forest, and a single layer neural network. Training and testing data sets are of an 80-20 split of the simulated data set and each model is fit with a 5-fold cross validation on the training set. Model performance is based on the three metrics of accuracy, sensitivity, and specificity. Before discussing the detail of each model a brief overview of the three performance metrics is given.

Accuracy is a very common metric that is used as a measure of performance of classification. It measures the rate at which all predictions are correct and is found as the ratio of total correct predictions to the total number of predictions. Although a good baseline metric, accuracy does have some limitations that can lead to misleading interpretations. If there is an imbalance in the data, such as a binary response with 90% of observations belonging to label 1 and 10% belonging to label 2, then a model that classifies all predictions as label 1 will report 90% accuracy. In this case accuracy on its own is not a good indicator of model performance since the model misclassified all of the label 2 predictions.

Generally sensitivity is the true positive rate and is found as the ratio of correctly predicted positive labels to the actual total positives. In terms of this study, sensitivity is the rate at which each model correctly

classifies individual 1. On the other hand, specificity is generally considered as the true negative rate and is found as the ratio of correctly predicted negative labels to the actual negative labels and here is the rate at which each model correctly classifies individual 2. The clear benefit of viewing sensitivity and specificity is that it measures a model's performance in terms of classifying a single label, which can reveal a frequency of misclassifications that could be hidden if only viewing accuracy. Any metric comes with some limitations and it is always good practice to view others for a holistic measure of performance. Although not considered at this time, other metrics can include precision, F1-score, and kappa statistic.

K-Nearest Neighbors

KNN performs classification on testing data by finding the conditional probabilities of K training observations that are closest to each testing observation x_0 belonging to the class label j. Each observation is assigned to the jth class label with the highest probability. KNN computes the conditional probabilities of the K training observations by

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{I}(y_i = j), \tag{7}$$

where $\mathcal{N}_0$ designates the set of K training observations closest to the test observation x_0 that is measured by Euclidean distance [20]. As a simple description, KNN performs classification based on a majority rules procedure. For the simulated data, the optimal value of the tuning parameter was found to be $K = 19$ as determined through validation accuracy.

Table 9: KNN Confusion Matrix

	Actual S1	Actual S2
Pred S1	28136	9689
Pred S2	17566	36009

Once each model is fit to the training set it is ready to make predictions from the testing data and compared to the actual values of the response. The predicted and actual values are used to evaluate the performance metrics of accuracy, sensitivity, and specificity, which can be calculated from the confusion matrix shown in table 9. A confusion matrix will only be displayed and discussed for the KNN model whereas the performance of all models will be based on the metrics themselves, summarized by table 10 and figure 13.

Table 9 shows the number of times testing data was predicted to belong to subject 1 or 2 and, in each case, the number of those being correct and incorrect. This model classifies data belonging to subject 1 or subject 2 with an accuracy rate of approximately 70%. It would correctly identify subject 2 more often than subject 1, which is quantified with a specificity of approximately 79% and sensitivity of 62% respectively.

Logistic Regression

Logistic regression is the only parametric model used for classification in this study. As in the case of time series forecasting models, parameters are estimated from the data and there are assumptions that should be satisfied by the data, two of which involve a binary response and low correlation between predictors. By viewing diagnostic plots it appears that assumptions hold for logistic regression when modeling the simulated data. Based from the logistic function

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}, \tag{8}$$

classification of a binary response is obtained through the probability of belonging to the *j*th label [20]. A threshold can be established and is commonly set to 0.5, which is the case for the classification of individual 1 from individual 2.

Being a parametric model, there are advantages and disadvantages to logistic regression over the other models used in this research. Advantages include interpretability and the low computational power needed to fit the model. Also, there is no need for tuning hyperparameters, which can involve an additional time investment for finding the best model. A disadvantage is that one needs to be aware of and carefully consider model assumptions.

Random Forest

Random forest is a non-parametric ensemble learner that aims to reduce the variance caused by decision trees by averaging multiple models. Random forest does this by creating multiple trees of bootstrapped subsets of training data. Additionally there is a random selection of predictors at each tree node that are used to classify data. Similar to KNN, the *j*th label with the most consistent result by majority across the trees results as the classified label.

As tuning parameters, a fixed number of *m* predictors are chosen at random at each split as well as a fixed number of *n* trees. Through cross validation the optimal values of *m* predictors and *n* trees selected for modeling the simulated data is 2 and 100 respectively. Additional tuning parameters in R include minsplit, the minimum number of observations at a node for a split to occur, and CP value, a stopping parameter that can increase efficiency in fitting the model. This study uses the default values for these two parameters.

Single Layer Neural Network

A basic understanding of a single layer neural network's architecture is that an input layer of p predictors is fully connected to a hidden layer with K hidden units in a feedforward fashion where some processing is applied and sent to an output layer where a final processing allows for prediction. The number of input layers is equivalent to the number of p predictors and, in the case of classification, the number of output layers depends on the number of labels of the response. For a binary response the number of output layers is equal to one. At each hidden layer weights are assigned to the inputs and inside each hidden unit an activation function creates a nonlinear transformation from a linear combination of the p predictors. Weights are initiated at random but are iteratively estimated from the data through a process referred to as

backpropagation. Additional weights and activation is applied at the output layer [20].

The number of hidden layers and hidden units are predetermined as well as the activation function at each layer. When fitting a neural network the number of epochs, which is the number of backpropagation iterations, and batch size, the number of observations that are propagated, can be specified. Multiple neural networks were trained, tested, and compared against one another in order to find an optimal architecture. Through this process the selected model consists of one hidden layer with three hidden units, a ReLU activation at the hidden layer, and a sigmoid activation function at the output layer. The model was fit using 30 epochs and a batch size of 32.

Classification Modeling Results

Table 10: Summary of Model Performance

Model	Accuracy	Sensitivity	Specificity
K Nearest Neighbors	0.7018	0.6156	0.7879
Logistic Regression	0.7130	0.6613	0.7647
Random Forest	0.7160	0.6059	0.8261
Neural Network	0.7167	0.6064	0.8271

Table 10 displays a summary of the three metrics used to evaluate each of the models' performance. There does not seem to be much of a difference in terms of performance based on accuracy with the KNN model having the lowest value of 0.7018 and the neural network slightly outperforming the other models with an accuracy of 0.7167. Considering sensitivity and specificity does reveal more nuances in the performance. Although random forest and the neural network have the highest accuracy, they each have lowest sensitivity of approximately 0.606 for both cases. This means they both have the lowest rate of successfully predicting individual 1 whereas logistic regression has the highest sensitivity of approximately 0.6613. On the other hand, random forest and the neural network both have superior specificity of 0.8261 and 0.8271 respectively and logistic regression has the lowest at 0.7647. If for some reason one was interested in a model that favors correctly classifying individual 2 then the random forest or neural network might be preferred.

KNN seems to remain in the middle with a sensitivity and specificity of 0.6156 and 0.7879 respectively. Recall that this model was fit with tuning parameter $K = 19$. However, increased values of K fits a model that performs similarly to the random forest and neural network. This demonstrates the importance of viewing multiple metrics of performance. If one were to only consider accuracy, the fact that a trade off between a slightly higher accuracy at the cost of a much lower sensitivity is not observed. It also shows the effect hyperparameters can have on a model's performance and the need for tuning to avoid overfitting, which can occur with non-parametric modeling.

The bar graph of figure 13 provides a visual summary of model performance. Accuracy is fairly even through all models and in every case, the models are better at classifying individual 2 than individual 1, with wider gaps between sensitivity and specificity existing in the non-parametric models. An additional

task was carried out in order to explore alternative performance of classification through time series simulation. Here the response has been changed from the binary label of individual to a six labeled categorical variable of the possible PSD bands of alpha, beta, and theta. There are six labels in order to make the distinction between PSD bands belonging to individuals 1 from individual 2. Table 11 shows the results of a KNN classifier used for predicting wave type and almost all of the misclassified labels result from the model's difficulty in distinguishing between the two individuals, continuing previous trends. What this shows is that the proportions of misclassifications across wave types follows the proportion of data itself, which makes intuitive sense. There are, due to the significantly wider frequency range, significantly more beta wave observations than alpha and even more so theta, and this carries over into errors.

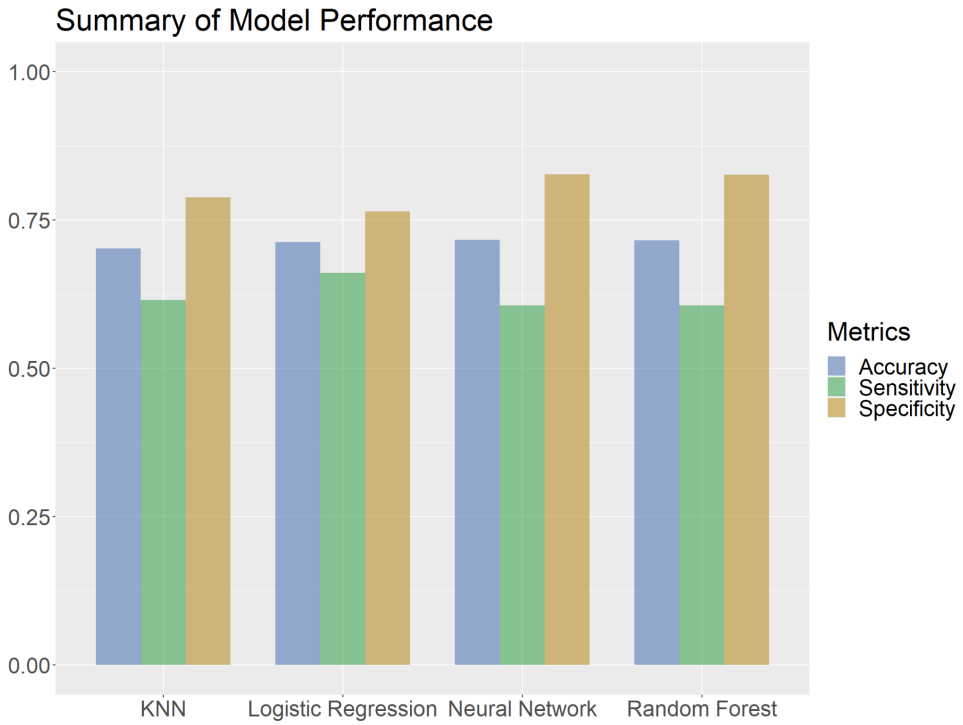

Figure 13. Bar graph to display and compare the performance of each model based on accuracy, sensitivity, and specificity.

Table 11: KNN Classification of PSD Estimated Band per Subject

	Actual Alpha1	Actual Beta1	Actual Theta1	Actual Alpha2	Actual Beta2	Actual Theta2
PredAlpha1	5417	2	0	3400	0	0
PredBeta1	0	18003	0	0	11846	0
PredTheta1	3	0	4467	0	0	2564
PredAlpha2	1824	0	0	6975	0	0
PredBeta2	0	6438	0	0	23327	0
PredTheta2	0	0	1294	0	0	5840

Conclusion

The classification of certain phenomena based on BCI data is an ongoing discipline that appears to have much potential for further exploration. There are many options on how to aggregate the collected data in a meaningful way and one must do so with a level of care in order to gain results that are both statistically significant and provide optimal results. The path taken in this study was to initially preserve the time series nature of the data in order to gain some level of inference. This ultimately results in the simulation of data from a parametric model fitted from a single representative EEG channel. Simulation, whether done parametrically or non-parametrically, is a tool that can overcome problems with limited sample sizes and allows for more accurate modeling. Two sets of 100 separate simulations were generated from data collected from two different individuals of the same segment, which was in turn transformed into the PSD bands of alpha, beta, and theta. A data matrix from this simulated data with appropriately assigned labels was used to fit the four different classification models from the non-aggregated frequency domain. The four models chosen are KNN, logistic regression, random forest, and a neural network with one hidden layer. The binary response being the two individual, model performance was based on the three metrics of accuracy, specificity, and sensitivity. The outcome shows that only viewing accuracy can be misleading as each model had similar accuracies but a variation in the levels of the other two metrics. Overall the neural network has the highest accuracy and specificity with values of 0.7167 and 0.8271 respectively and logistic regression results in the highest sensitivity of 0.6613. Consistently all four models are able to predict individual 2 than individual 1 from the data as shown with the higher values of specificity.

Future Work and Recommendations

Due to limitations in computational power there are opportunities to improve the outcomes of the learning models implemented. For example, a gridsearch for tuning hyperparameters may help to improve prediction accuracy since it finds the optimal set of hyperparameters out of all possible combinations. And although the focus of the research was on EEG data, the same approach to classification can be applied to fNIRS data. It is important to note that due to the complementary relationship between EEG and fNIRS, and the fact that fNIRS has good spatial resolution, it may be beneficial when finding the representative channels especially if the data collection is expanded beyond the frontal lobe. And these same techniques of finding the most representative channels can be applied to any experimental goal of interest.

For future research, the use of an expert in the experimental design portion of the experiment would be greatly beneficial to the quality of data being collected, such as expanding the pool of participants to extend to additional demographics. Additionally, repetition of unique tasks can add to the experiments by creating trials which can then be further analyzed by comparing them against each other and determining whether they are accurate and reliable. These trials could then implement blocking and randomization to limit any effects to the data, such as a learning effect of playing a game over and over again. Lastly, since EEG and fNIRS are time-series signals there can be great benefits to collecting additional behavioral data to be able to timestamp events in the data.

References

[1] De La Cruz, Javier, Janson Law, Nii-Kwame Oteng-Quarshie, and Kiran George. 2022. “EEG and fNIRS Analysis to Determine Acute Stress Resulting From Reaction Time Tests.” Manuscript submitted for publication. Computer Engineering Department, California State University, Fullerton.

[2] De La Cruz, Javier, Douglas Shimizu, and Kiran George. 2022. “EEG and fNIRS Analysis Using Machine Learning to Determine Stress Levels.” In *2022 IEEE World AI IoT Congress (AIIoT), 318–22. IEEE.*

[3] De La Cruz, Javier*, a*nd Kiran George. 2022. “Acute Stress Analysis Resulting from Word Construction Using EEG and fNIRS.” In *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 0362–66. https://doi.org/10.1109/UEMCON54665.2022.9965628.

[4] Al-Shargie, Fares, Tong Boon Tang, Nasreen Badruddin, Sarat C Dass, and Masashi Kiguchi. 2016. “Mental Stress Assessment Based on Feature Level Fusion of fNIRS and EEG Signals.” In *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS)*, 1–5. IEEE.

[5] Bhutta, M Raheel, Melissa J Hong, Yun-Hee Kim, and Keum-Shik Hong. 2015. “Single-Trial Lie Detection Using a Combined fNIRS-Polygraph System.” *Frontiers in Psychology* 6: 709.

[6] Wheeler, Robert E, Richard J Davidson, and Andrew J Tomarken. 1993. “Frontal Brain Asymmetry and Emotional Reactivity: A Biological Substrate of Affective Style.” *Psychophysiology* 30 (1): 82–89.

[7] Desoto, Abel, Joshua Dodd, Michael Babinec, and Kiran George. 2022. “Using EEG and fNIRS to Determine Neural Alignment Through Storytelling.” In *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 0352–54. IEEE.

[8] Khalil, Mohammad Affan, Maria Ramirez, Johnny Can, and Kiran George. 2022a. “Implementation of Machine Learning in BCI Based Lie Detection.” In *2022 IEEE World AI IoT Congress (AIIoT)*, 213–17. IEEE.

[9] Khalil, Mohammad Affan, and Kiran George. 2022. “Using Neural Network Models for BCI Based Lie Detection.” In *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 0505–9. IEEE.

[10] Zhang, Zhiguo. 2019. “Spectral and Time-Frequency Analysis.” *EEG Signal Processing and Feature Extraction*, 89–116.

[11] Helwig, Nathaniel E. 2018. *Eegkit: Toolkit for Electroencephalography Data*. https://CRAN.R-project.org/package=eegkit.

[12] R Core Team. 2022. R: *A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

[13] Nayak, Chetan S., and Arayamparambil C. Anilkumar. 2022. *EEG Normal Waveforms*. StatPearls Publishing, Treasure Island (FL). http://europepmc.org/books/NBK539805.

[14] Bigger, Alana, editor. “Healthline.” *What Is the Purpose of Theta Brain Waves?*, 1 July 2020.

[15] Luke, Robert, et al. “Oxygenated Hemoglobin Signal Provides Greater Predictive Performance of Experimental Condition than De-Oxygenated.” *BioRxiv*, 2021, https://doi.org/10.1101/2021.11.19.469225.

[16] Li, Rihui, et al. “Concurrent FNIRS and EEG for Brain Function Investigation: A Systematic, Methodology-Focused Review.” *Sensors*, vol. 22, no. 15, 2022, p. 5865., https://doi.org/10.3390/s22155865.

[17] Li, Rihui, et al. “Enhancing Performance of a Hybrid EEG-FNIRS System Using Channel Selection and Early Temporal Features.” *Frontiers in Human Neuroscience*, vol. 11, 2017, https://doi.org/10.3389/fnhum.2017.00462.

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

[19] Hyndman, Rob J., and George Athanasopoulos. 2018. *Forecasting: Principles and Practice.* 2nd ed. Otexts.com/ffp.

[20] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. An Introduction to StatisticalLearning: With Applications in r. Springer. https://faculty.marshall.usc.edu/gareth-james/ISL/.